

ฉลาดใช้สารพัดสูตร Excel อย่างมืออาชีพ

โดย

สมเกียรติ ฟุ่งเกียรติ

www.ExcelExpertTraining.com

Excel@ExcelExpertTraining.com

โทร 097-140-5555, 02 718 9331

สงวนลิขสิทธิ์ตามกฎหมาย

13 เมษายน 2564

สารบัญ

1. วิธีปรับ Excel ให้พร้อมต่อการใช้งานอย่างมืออาชีพ	6
2. การกำหนดตำแหน่งอ้างอิง	8
3. วิธีค้นหาสูตร	16
4. วิธีแกะสูตร	20
5. วิธีสร้างสูตรซ้อนสูตร	22
6. วิธีสร้างชื่อซ้อนชื่อ	24
7. เรื่องที่ไม่ค่อยรู้กันเกี่ยวกับสูตร	25
8. การแบ่งขอบเขตพื้นที่ตาราง	40
9. การกำหนดตำแหน่งอ้างอิงข้ามชีทหรือข้ามแฟ้ม	44
10. หลักการตั้งชื่อและใช้ชื่อใน Excel	54
11. การใช้สูตรปรับตำแหน่งอ้างอิง	67
12. การกำหนดตำแหน่งอ้างอิงแบบวงกลม	72
13. แป้นพิมพ์ลัดที่เกี่ยวข้องกับตำแหน่งอ้างอิง	79
14. หลักการแยกตัวแปร	80
15. วิธีใช้ Goal Seek	81

16. วิธีใช้ Data Table	86
17. หลากหลายแนวทางการใช้สูตร IF	91
18. การใช้สูตร IF ในการตัดสินใจ	96
19. หลบร้อนหนีสูตร IF ไปหาทางออกอื่น	105
20. หนีสูตร IF ไปใช้คำสั่ง Format	115
21. วิธีใช้ Data Validation	116
22. วิธีใช้ Conditional Formatting	117
23. สูตร Array	119
24. สูตร Array IF เพื่อใช้หายอดรวมแยกประเภท	131
25. ของจริง ของปลอม ของใช้ไม่ได้ใน Excel	144
26. หลักการใช้วันที่และเวลา	148
27. หลักการกำหนด Format Cells > Number	152
28. วิธีใช้ Add-In	156
29. Function VBA	157
30. General Functions	202
31. Day and Time Formulas	224

32. List & Database Formulas	247
33. Conditional Formatting & Data Validation Formulas	249
34. Array Formulas	254
35. การกำหนดขอบเขตตารางแบบยืดหยุ่น	282
36. การจัดการค่าซ้ำ	298
37. การคำนวณยอดรวมเพื่อจัดทำรายงาน	305
38. วิธีสร้างสูตรบนกราฟ	313
39. วิธีจัดการกับความอ้วนอ้วน...ของแฟ้ม	316
40. การออกแบบโครงสร้างตารางคำนวณ	326

คำนำ

ในระยะเวลาหลายสิบปีที่ผ่านมา แม้ Excel ถูกพัฒนาให้เป็นโปรแกรมที่เป็นประโยชน์มากขึ้น มีคำสั่งและมีเครื่องมือใหม่ๆเตรียมไว้ให้เลือกใช้มากขึ้นกว่าแต่ก่อนก็ตาม แต่ตัวสูตรที่ Excel มีอยู่กลับมีการเปลี่ยนแปลงน้อยมาก มีสูตรใหม่ที่เป็นต่อการใช้งานเกิดขึ้นใหม่เพียงเล็กน้อยเท่านั้น เช่น ใน Excel 365 รุ่นล่าสุดมีสูตรที่ช่วยลดขั้นตอนในการคำนวณได้ง่ายขึ้นโดยไม่จำเป็นต้องเสียเวลาคำนวณทีละขั้น แต่สูตรใหม่ไม่สามารถใช้กับ Excel รุ่นก่อน

หลักสูตรฉลาดใช้สารพัดสูตร Excel อย่างมืออาชีพก็ผ่านร้อนผ่านหนาวมาหลายปีเช่นกัน หลักสูตรนี้มุ่งให้ผู้ใช่ Excel รู้จักคิด ฉลาดเลือกใช้สูตร Excel ให้ถูกกับสถานการณ์ โดยเลือกเฉพาะสูตรที่มีความจำเป็นต้องนำมาใช้งานขึ้นมามากเพื่อทราบข้อดี ข้อเสีย และข้อควรระวัง แนะนำวิธีคิดออกแบบตารางให้เหมาะสมกับการคำนวณ และหาทางกระตุ้นให้ผู้เข้าอบรมเรียนรู้วิธีแก้ปัญหาได้ด้วยตัวเอง โดยมุ่งใช้สูตรที่ใช้ได้กับ Excel ทุกรุ่นเป็นหลัก

อย่างไรก็ตามเนื่องจากระยะเวลาเรียนในห้องอบรมมีเพียง 2-3 วัน ไม่สามารถแนะนำเรื่องราวและถ่ายทอดประสบการณ์การใช้สูตรได้ครบถ้วน จึงขอแนะนำให้อ่านคู่มือเล่มนี้ซึ่งมีเนื้อหาที่อาจไม่มีโอกาสหยิบยกขึ้นมามากมายอีกมาก

คู่มือประกอบการอบรมเล่มนี้ได้รวบรวมเนื้อหาสำคัญของหลายหลักสูตร ทั้งหลักสูตรสุดยอดเคล็ดลับและลัดของ Excel หลักสูตรหันมาใช้ Excel จัดการฐานข้อมูลแทน Access กันดีกว่า และหลักสูตรประยุกต์ใช้ Excel เพื่อวางแผนความต้องการวัสดุและจัดตารางการผลิต ตลอดจนการใช้สูตรใน VBA อีกทั้งได้นำตัวอย่างคำถามคำตอบและสูตรแบบแปลกๆจากเว็บหลายแห่งมารวมไว้ จึงหวังว่าผู้เข้าอบรมจะได้เปิดหูเปิดตาและได้แนวทางนำกลับไปสร้างงานของตน

ประเด็นสำคัญของการสร้างสูตร คือ ต้องรู้จักคิดและสร้างสูตรได้ด้วยตัวเอง มิใช่ลอกสูตรของคนอื่นมาใช้แล้วแกะหรือแก้ไขต่อให้ดีขึ้นกว่าเดิมไม่เป็น

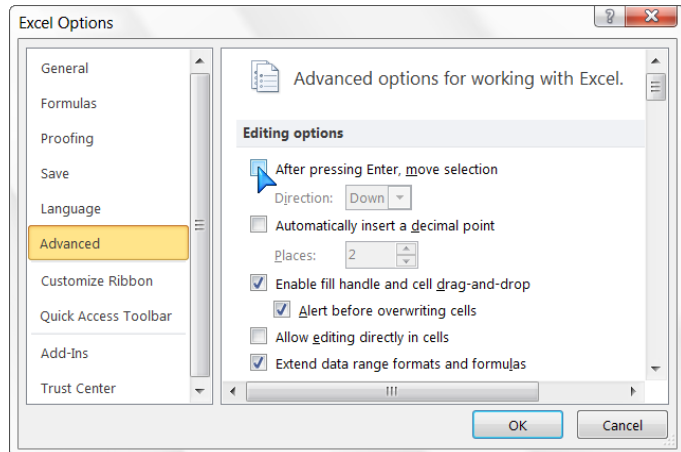
สมเกียรติ พุ่งเกียรติ

13 เมษายน 2564

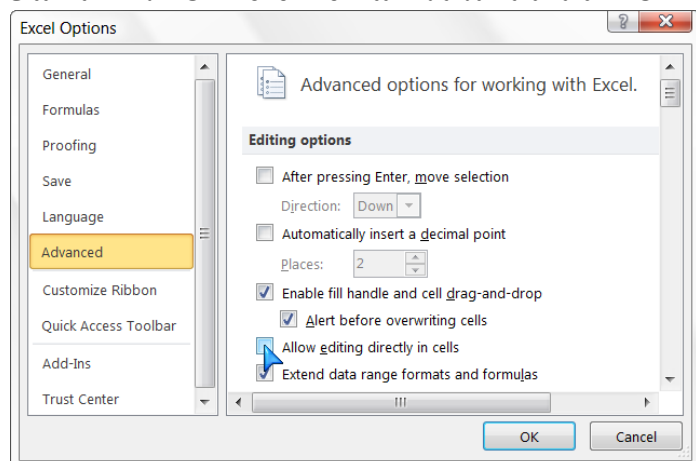
วิธีปรับ Excel ให้พร้อม ต่อการใช้งานอย่างมืออาชีพ

1. ต้องการปรับให้กดปุ่ม **Enter** เพื่อ
บันทึกค่าแล้วอยู่ที่เซลล์เดิม สั่ง
File > Options > Advanced > ตัด
กาช่อง After pressing Enter,
Move selection

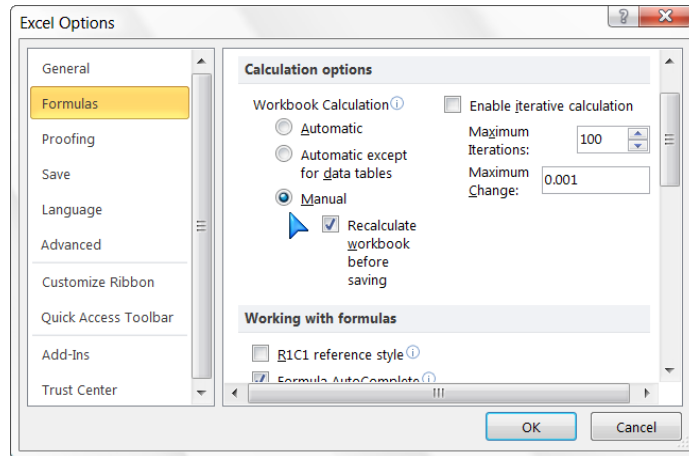
ทั้งนี้ใน Excel 2007 สั่ง Office >
Excel Options แทน File >
Options >



2. ต้องการปรับให้กดปุ่ม **F2** เพื่อ **Edit** แก้ไขบน **Formula Bar** แทนในเซลล์ สั่ง File >
Options > Advanced > ตัดกา
ช่อง Allow editing directly in
cells
คำสั่งนี้ยังมีผลทำให้เมื่อดับเบิล
คลิกที่เซลล์สูตรแล้ว Excel จะย้าย
ตำแหน่งไปยังตำแหน่งเซลล์ต้น
ทางที่อ้างอิงในสูตรนั้น และเมื่อกด
ปุ่ม F5 > Enter จะย้ายกลับมาที่
เซลล์เดิมที่เคยดับเบิลคลิก



3. ต้องการเปลี่ยนระบบการคำนวณเป็น **manual** สั่ง File > Options > Formulas > เลือก Manual จากนั้นต้องกดปุ่ม F9 เพื่อสั่งคำนวณทุกแฟ้มหรือกด Shift+F9 เพื่อสั่งคำนวณเฉพาะชีทที่เลือกอยู่เท่านั้น



หมายเหตุ ระบบการคำนวณจะเปลี่ยนตามระบบของแฟ้มแรกที่เปิดขึ้นก่อนเสมอ

การกำหนดตำแหน่งอ้างอิง

บทความนี้มีจุดประสงค์ให้ผู้ที่เคยใช้ Excel มาก่อนและมีพื้นฐานมาบ้าง ซึ่งพอทราบว่าตำแหน่งอ้างอิง (Cell Reference) ในตาราง Excel คืออะไรและนำไปใช้กันอย่างไร จะได้เปิดหูเปิดตาเรียนรู้เคล็ดลับที่คนทั่วไปอาจมองข้ามหรือไม่นึกฝันว่าจะเป็นไปได้ แล้วสามารถนำตำแหน่งอ้างอิงไปใช้กับสูตรและคำสั่งได้อย่างพลิกแพลงมากยิ่งขึ้น โดยเฉพาะการเลือกใส่เครื่องหมาย \$ ให้กับตำแหน่งอ้างอิงนั้น ไม่ว่าจะเป็น \$ ในเซลล์สูตร, \$ ใน Conditional Format, \$ ใน Data Validation, หรือ \$ ในในเครื่องมืออื่นๆของ Excel ก็ตาม ผมถือว่า \$ เป็นหัวใจของการใช้ Excel ที่เดียว

ตั้งหลักกันก่อนว่าจะใช้ Excel อย่างคอมพิวเตอร์

คุณลองนึกดูว่ามนุษย์เราได้รับประโยชน์จากการใช้เครื่องคอมพิวเตอร์ต่างจากสมัยที่ยังมีแต่เครื่องคิดเลขให้ใช้กันอย่างไรบ้าง หลายคนน่าจะตอบว่า เครื่องคอมพิวเตอร์ต้องช่วยให้คำตอบที่เร็วกว่าเครื่องคิดเลขใช่ไหม แล้วความเร็วที่วุ่นนี้ถูกวัดมาจากเวลาที่ใช่ไปของเครื่องคอมพิวเตอร์หรือเวลาของคุณกันละ

แน่นอนที่สุดว่าเครื่องคอมพิวเตอร์ย่อมทำงานเร็วกว่าเครื่องคิดเลข และเครื่องคอมพิวเตอร์รุ่นใหม่ย่อมทำงานได้เร็วกว่าเครื่องคอมพิวเตอร์รุ่นเก่า ยิ่งใช้ Excel ที่ติดตั้งบนเครื่องคอมพิวเตอร์รุ่นล่าสุด ก็ยิ่งได้คำตอบผลลัพธ์จากการคำนวณเร็วขึ้น ซึ่งถ้าใครหลงเข้าใจว่าตัวเองใช้ Excel แบบคอมพิวเตอร์เป็นอยู่แล้วเพียงเพราะดูจากความเร็วของเครื่องคอมพิวเตอร์ที่ใช้อยู่ ก็เป็นความเข้าใจที่ถูกต้องเพียงบางส่วน (และเป็นส่วนที่ไม่สำคัญเท่าใดเสียด้วย) เพราะถ้าคุณใช้ Excel อย่างคอมพิวเตอร์เป็น ต่อให้ยกเครื่องคอมพิวเตอร์รุ่นโบราณที่ทำงานช้าเป็นเต่า ก็จะไม่พบว่าคุณไม่ได้เสียเวลาต่างกันมากนัก

ลักษณะของการใช้ Excel อย่างคอมพิวเตอร์มีดังนี้

1. คุณจะต้องเสียเวลาสร้างงานนานหน่อย เนื่องจากต้องหาทางคิดสร้างงานให้เหมาะสมกับแต่ละสถานการณ์ ซึ่งคุณควรตั้งหลักไว้ว่า แฟ้มงานที่สร้างขึ้นมานั้น จะต้องสามารถนำกลับมาใช้ซ้ำแล้วซ้ำอีกได้โดยไม่ต้องย้อนกลับไปแก้ไขใหม่อีกแม้แต่บ่อย
2. หากจำเป็นต้องแก้ไข ต้องสามารถแก้ไขปรับปรุงแฟ้มได้ง่าย ใช้เวลาไม่มาก และไม่ส่งผลกระทบต่อการคำนวณเดิมที่ถูกต้องอยู่แล้ว ทั้งนี้ไม่ว่าจะกำหนดให้ผู้ที่ทำหน้าที่

แก้ไขนั้นคือคุณซึ่งเป็นผู้สร้างแฟมั้นเอง หรือแม้แต่จะเป็นผู้อื่นซึ่งไม่เคยใช้แฟมั้นมาก่อนก็ตาม

3. สามารถลดจำนวนคนและไม่ต้องเพิ่มอัตราค่าจ้างคนตามปริมาณงานที่เพิ่มขึ้น
4. ประหยัดเงินลงทุน เพราะไม่ต้องจ่ายเพื่อซื้อเครื่องหรือโปรแกรมคอมพิวเตอร์รุ่นใหม่
5. คุณต้องมีความสุขและมีเวลาว่างให้กับชีวิตมากขึ้น

สูตรฉลาด (Smart Formula)

Smart Formula แปลว่า สูตรฉลาด ซึ่งสูตรฉลาดที่ว่านี้จะฉลาดได้ก็ต่อเมื่อใช้ความฉลาดของคุณในการสร้างสูตรขึ้นมา เพื่อให้มีลักษณะการใช้งานได้ตามนี้

ในชีทหนึ่งๆหรือตารางหนึ่งๆ ต้องมีสูตรต้นฉบับเพียงเซลล์เดียว ซึ่งสามารถนำไปใช้ได้กับเซลล์อื่นๆได้ทั้งตาราง แล้วเมื่อใดที่จำเป็นต้องแก้ไขสูตรใหม่ ก็จัดการแก้ไขสูตรนี้เพียงเซลล์เดียวแล้ว Copy นำไป Paste เพื่อใช้ได้กับทุกเซลล์ในตาราง

ในโครงสร้างของตัวสูตรฉลาดอาจใช้วิธีกำหนดตำแหน่งอ้างอิงแบบเซลล์เดียวหรือแบบกำหนดขอบเขตตารางก็ได้ โดยไม่ว่าจะใช้วิธีใดก็หลักการกำหนดตำแหน่งอ้างอิงที่มีเครื่องหมาย \$ กำกับตำแหน่ง 3 แบบต่อไปนี้ไปไม่พ้น

- **ตำแหน่งอ้างอิงแบบ Absolute Reference** มีเครื่องหมาย \$ วางไว้ด้านหน้าของทั้งตำแหน่ง Row และ Column เช่น =\$A\$1 โดยให้ใช้ตำแหน่งอ้างอิงแบบนี้กับกรณีที่ต้องการนำค่าคงที่ไปใช้
- **ตำแหน่งอ้างอิงแบบ Relative Reference** เป็นตำแหน่งอ้างอิงซึ่งไม่มีเครื่องหมาย \$ เช่น =A1 ซึ่งสูตรที่อ้างอิงแบบนี้จะต้องใช้ที่เซลล์เดิมเซลล์เดียวไปตลอด ไม่สามารถ Copy นำไป Paste เพื่อทำให้ได้ค่าเดิมซ้ำที่เซลล์อื่นได้อีก เพราะตำแหน่งอ้างอิงจะเขยื้อนต่างจากตำแหน่งเดิมเสมอ ดังนั้นจึงเป็นการอ้างอิงแบบที่ใช้ในเซลล์สูตรกันน้อยที่สุด แต่กลับถูกนำมาใช้กันมากใน Conditional Formatting หรือ Data Validation
- **ตำแหน่งอ้างอิงแบบ Mixed Reference** เป็นตำแหน่งอ้างอิงที่นิยมใช้กันมากที่สุด โดยผู้สร้างสูตรต้องเลือกใส่เครื่องหมาย \$ ด้านหน้าตำแหน่ง Row หรือ Column เพียงอย่างใดอย่างหนึ่ง เช่น =A\$1 เพื่อคงตำแหน่ง Row 1 ไว้ตามเดิมตลอด หรือ =\$A1 เพื่อคงตำแหน่ง Column A ไว้ตามเดิมตลอด ไม่ว่าจะ Copy นำสูตรไป Paste ที่ตำแหน่งอื่นก็ตาม

วิธีสร้างสูตรให้มีตำแหน่งอ้างอิงที่มีเครื่องหมาย \$ นำหน้าตำแหน่งของ Row และหรือ Column นี้ ให้ใช้วิธี Pointing แล้วกดปุ่ม **F4** ซ้ำไปเรื่อยๆจนกว่าจะได้ตำแหน่ง \$ ตามต้องการ

1. ให้เริ่มจากพิมพ์เครื่องหมายเท่ากับ = ลงไปในเซลล์ที่ต้องการสร้างสูตร (เพื่อแสดงการยึดครองเซลล์นั้นไว้ก่อนว่า เป็นเซลล์ที่จะรับสูตรลงไปเมื่อเรากดปุ่ม Enter)
2. จากนั้นใช้วิธีชี้ลงไปในเซลล์ที่ต้องการ (Pointing แปลว่า ชี้) โดยใช้ Mouse คลิกเลือกเซลล์ที่ต้องการ ไม่ว่าจะอยู่ต่างชีทหรือต่างแฟ้มกันก็ตาม เช่น คลิกที่เซลล์ A1 จะได้สูตร =A1
3. จากนั้นให้กดปุ่ม F4 ไปเรื่อยๆ จะพบว่าเครื่องหมาย \$ แทรกลงในสูตรให้เองตามลำดับที่มีการกดปุ่ม F4 ดังนี้ A1 → \$A\$1 → A\$1 → \$A1 → A1 → แล้วจะมีเครื่องหมาย \$ วนซ้ำไปอีก
4. เมื่อได้ตำแหน่ง \$ ควบคุมตำแหน่ง Row และหรือ Column ตามต้องการแล้วให้กดปุ่ม Enter เพื่อรับสูตรลงไปในเซลล์ แล้วจะพบว่า Excel จะพากลับมาที่เซลล์ซึ่งพิมพ์เครื่องหมายเท่ากับนำหน้าสูตรเอาไว้ตั้งแต่แรกนั่นเอง

การใส่เครื่องหมาย \$ เพื่อเป็นการควบคุมตำแหน่งอ้างอิงไว้ไม่ให้เขยื้อนนี้ นอกเหนือจากการทำให้คุณสามารถ Copy สูตรไปใช้ในเซลล์อื่นได้ทันทีแล้ว ยังเป็นเรื่องน่าสังเกตว่าช่วยทำให้แฟ้มมีขนาดเล็กลงกว่าแฟ้มที่มีสูตรซึ่งไม่มี \$ กำหนดไว้เลย (ขอให้ทดลองลบเครื่องหมาย \$ ทั้งจากสูตรในตารางขนาดใหญ่เพื่อเปรียบเทียบดูขนาดแฟ้ม โดยใช้คำสั่ง Replace เครื่องหมาย \$ แต่ไม่ได้ Replace with ด้วยเครื่องหมายใด)

นอกจากตำแหน่งอ้างอิงที่ต้องมีเครื่องหมาย \$ เป็นองค์ประกอบสำคัญแล้ว ในโครงสร้างของสูตร Smart Formula ยังอาจจำเป็นต้องพึ่งสูตร IF, And, Or หรือสูตรอื่นๆซึ่งทำหน้าที่ตัดสินใจเพื่อช่วยให้สูตรมีเงื่อนไขในการคำนวณต่างไปจากเดิมได้เองทันที เมื่อมีการนำสูตรเดียวกันไปใช้ซ้ำที่ตำแหน่งเซลล์อื่น

ใน Excel 365 รุ่นล่าสุดมีวิธีการสร้างสูตรแบบ Dynamic Array เพิ่มขึ้นใหม่โดยจะกระจายสูตรที่ใช้ตำแหน่งอ้างอิงหลายเซลล์ไปให้เองตามขนาดของเซลล์ที่ใช้อ้างอิงโดยไม่ต้องใส่ \$ และไม่ต้องกดปุ่ม Ctrl+Shift+Enter เพื่อสร้าง {} ปิดหัวท้ายสูตรอีกต่อไป หากนำแฟ้มที่สร้างจากรุ่นก่อนมาเปิดใน Excel 365 จะพบเครื่องหมาย @ นำหน้าสูตรเพื่อดึงค่าเพียงค่าเดียวมาใช้และหยุดการกระจายสูตรออกไป

ตัวอย่าง Smart Formula แบบใช้ตำแหน่งอ้างอิงเซลล์เดียว

ขอให้สร้างตารางสูตรคูณ ซึ่งนำตัวเลขบนหัวตารางมาคูณกับตัวเลขด้านข้างซ้ายของตาราง

	C3 fx =C\$2*\$B3					
	A	B	C	D	E	F
1						
2			2	3	4	5
3		10	20	30	40	50
4		20	40	60	80	100
5		30	60	90	120	150
6						
7			C3 : =C\$2*\$B3			

ขั้นตอนการสร้าง

1. เริ่มจากเลือกเซลล์ C3 แล้วพิมพ์เครื่องหมายเท่ากับ = ลงไป
2. จากนั้นคลิกเซลล์ C2 จะได้สูตร =C2
3. กดปุ่ม F4 เพื่อเปลี่ยนสูตรให้เป็น =C\$2
4. พิมพ์เครื่องหมายคูณ *
5. จากนั้นคลิกเลือกเซลล์ B3 จะได้สูตร =C\$2*B3
6. กดปุ่ม F4 เพื่อเปลี่ยนสูตรเป็น =C\$2*\$B3
7. กดปุ่ม Enter เพื่อรับสูตรลงไปในเซลล์ C3
8. Copy ไปทับตาราง C3:F5

คำอธิบาย

- เซลล์ C3 เป็นเซลล์ที่มีสูตร Smart Formula ซึ่งโดยทั่วไปจะเลือกให้เซลล์หัวมุมด้านซ้ายบนของตารางเป็นเซลล์ที่มีสูตร Smart Formula
- คุณอาจเลือกสร้างสูตรคูณโดยใช้เลข 2 นำหน้าแล้วนำไปคูณกับเลข 10 เพื่อให้ได้สูตร =C\$2*\$B3 หรือจะเลือกใช้เลข 10 นำหน้าแล้วนำไปคูณกับเลข 2 ซึ่งจะทำให้เกิดสูตร =\$B3*C\$2 ก็ได้และทำให้เกิดผลลัพธ์ถูกต้องเหมือนกันทั้งสองแบบ แต่ขอแนะนำให้สร้างพฤติกรรมในการสร้างสูตรของตนว่า **“ทุกสูตรที่สร้างขึ้นจะนำตำแหน่งอ้างอิงตามแนวนอนขึ้นก่อนตำแหน่งอ้างอิงตามแนวตั้งเสมอ”** ซึ่งหลักนี้จะช่วยให้คุณแกะสูตรของตัวเองได้ง่ายขึ้น และหากสังเกตสูตรอื่นๆของ Excel เช่น Index หรือ Offset

จะใช้ตำแหน่งอ้างอิงจากแนวนอน Row ขึ้นก่อนตำแหน่งอ้างอิงจากแนวตั้ง Column เช่นกัน

- C\$2 ต้องมีเครื่องหมาย \$ วางไว้หน้าเลข 2 เนื่องจากตารางสูตรคุณนี้มี Row 2 เป็นตำแหน่งอ้างอิงที่ต้องควบคุมไว้ไม่ให้เปลี่ยน ส่วนตัว C ไม่ต้องมีเครื่องหมาย \$ นำหน้า เพื่อปล่อยให้ C เปลี่ยนตำแหน่งเป็น Column D E F หรือ Column อื่นๆตามแต่จะใช้สูตรนี้หรือไม่
- \$B3 ต้องมีเครื่องหมาย \$ วางไว้หน้า Column B เนื่องจากตารางสูตรคุณนี้มี Column B เป็นตำแหน่งอ้างอิงที่ต้องควบคุมไว้ไม่ให้เปลี่ยน ส่วนเลข 3 ไม่ต้องมีเครื่องหมาย \$ นำหน้า เพื่อปล่อยให้เลข 3 เปลี่ยนตำแหน่งเป็น Row 4 5 6 หรือ Row อื่นๆตามแต่จะใช้สูตรนี้หรือไม่
- หลักการใส่เครื่องหมาย \$ นี้ ให้จำไว้ว่า "ถ้าตำแหน่งอ้างอิงในสูตรมาจากค่าตามแนวนอน ให้ใส่ \$ ไว้หน้า Row (เพราะ แนวนอน = Row) แต่ถ้าตำแหน่งอ้างอิงในสูตรมาจากค่าตามแนวตั้ง ให้ใส่ \$ ไว้หน้า Column (เพราะ แนวตั้ง = Column)"

ตัวอย่าง Smart Formula แบบใช้ตำแหน่งอ้างอิงตามขอบเขตพื้นที่หัวตาราง

ขอให้สร้างตารางสูตรคุณ ซึ่งนำตัวเลขบนหัวตารางมาคูณกับตัวเลขด้านข้างซ้ายของตาราง

	C3		= \$C\$2:\$F\$2*\$B\$3:\$B\$5					
	A	B	C	D	E	F	G	H
1								
2			2	3	4	5		
3		10	20	30	40	50		
4		20	40	60	80	100		
5		30	60	90	120	150		
6								
7			C3 : = \$C\$2:\$F\$2*\$B\$3:\$B\$5					

วิธีการสร้างคล้ายกับวิธีแรก แทนที่จะคลิกเลือกเซลล์ C2 หรือเซลล์ B3 วิธีนี้ให้คลิกเลือกเซลล์ C2:F2 และ B3:B5 ซึ่งเป็นหัวตารางด้านบนและด้านซ้ายทั้งหมด แล้วกดปุ่ม F4 เพื่อทำให้เกิดสูตร = \$C\$2:\$F\$2*\$B\$3:\$B\$5

1. เริ่มจากเลือกเซลล์ C3 แล้วพิมพ์เครื่องหมายเท่ากับ = ลงไป
2. จากนั้นคลิกเลือกเซลล์บนหัวตารางด้านบนจากเซลล์เลข 2 - 5 จะได้สูตร =C2:F2

3. กดปุ่ม F4 เพื่อเปลี่ยนสูตรให้เป็น $=\$C\$2:\$F\2
4. พิมพ์เครื่องหมายคูณ *
5. จากนั้นคลิกเลือกเซลล์บนหัวตารางด้านซ้ายจากเซลล์เลข 10 - 30 จะได้สูตร $=\$C\$2:\$F\$2*B3:B5$
6. กดปุ่ม F4 เพื่อเปลี่ยนสูตรเป็น $=\$C\$2:\$F\$2*\$B\$3:\$B\5
7. กดปุ่ม Enter เพื่อรับสูตรลงไปเซลล์ C3 แล้ว Copy ไปหัวตาราง C3:F5 (Excel 365 จะกระจายสูตรลงไปเซลล์ C3:F5 ให้เอง)

คำอธิบาย

- สูตรแบบที่อ้างอิงกับตำแหน่งทั้งหมดบนหัวตารางหรือข้างตาราง แล้วใส่เครื่องหมาย \$ ให้กับตำแหน่งทั้ง Row และ Column นี้ ผมตั้งชื่อเรียกว่า **สูตรแบบล่อมรั้ว** โดยที่สูตร $=\$C\$2:\$F\$2*\$B\$3:\$B\5 จะทำหน้าที่นำตำแหน่งอ้างอิงบนรั้วตามแนวตั้งจากที่ตัดกันมาคูณกันทีละตัวให้เอง ทำให้ไม่ต้องคอยพะวงว่าจะต้องเลือกใส่เครื่องหมาย \$ เพื่อควบคุมตำแหน่งอ้างอิงตามแนวใดเป็นหลัก
- ถ้าเปรียบเทียบกับสูตรคุณวิธีแรกซึ่งใช้ตำแหน่งอ้างอิงแบบเซลล์เดียว จะพบว่าการใช้สูตรอ้างอิงแบบเซลล์เดียวมีความยืดหยุ่นมากกว่า เช่นเมื่อหัวตารางมีขนาดใหญ่มากขึ้นก็สามารถ Copy สูตรไปใช้ต่อได้ทันที แต่ถ้าสร้างสูตรแบบล่อมรั้ว จะต้องสร้างสูตรใหม่เพื่อกำหนดตำแหน่งอ้างอิงในสูตรให้มีขนาดขยายตาม (เว้นแต่ว่าคุณทราบวิธีขยายขอบเขตตำแหน่งอ้างอิง ซึ่งจะนำมาอธิบายต่อไปในภายหลัง)
- จุดอ่อนของสูตรแบบล่อมรั้วก็คือ สูตรแบบนี้ใส่เครื่องหมาย \$ ควบคุมตำแหน่งแบบ Absolute จึงคำนวณหาค่าตอบให้ได้เฉพาะตารางจาก C3:F5 ซึ่งอยู่ภายในขอบเขตของรั้วเป็นแนวของตำแหน่งคงที่ หาก Copy สูตรไปใช้นอกรั้วจะเกิด Error ขึ้นตามรูป

	C6		fx = \$C\$2:\$F\$2*\$B\$3:\$B\$5					
	A	B	C	D	E	F	G	H
1								
2			2	3	4	5	6	7
3		10	20	30	40	50	#VALUE!	#VALUE!
4		20	40	60	80	100	#VALUE!	#VALUE!
5		30	60	90	120	150	#VALUE!	#VALUE!
6		40	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!
7		50	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!
8								
9			C6 : = \$C\$2:\$F\$2*\$B\$3:\$B\$5					

ตัวอย่าง Smart Formula แบบใช้ตำแหน่งอ้างอิงแบบ Array ตามขอบเขตพื้นที่หัวตาราง

ขอให้สร้างตารางสูตรคูณ ซึ่งนำตัวเลขบนหัวตารางมาคูณกับตัวเลขด้านข้างซ้ายของตาราง

	C3 {=C2:F2*B3:B5}					
	A	B	C	D	E	F
1						
2			2	3	4	5
3		10	20	30	40	50
4		20	40	60	80	100
5		30	60	90	120	150
6						
7			C3 : {=C2:F2*B3:B5}			

ขั้นตอนการสร้าง

1. แทนที่จะเลือกเซลล์ C3 เพียงเซลล์เดียว คราวนี้ให้เลือกพื้นที่ตารางจากเซลล์ C3:F5 ทั้งหมดพร้อมกันไว้ก่อน
2. จากนั้นพิมพ์เครื่องหมายเท่ากับ =
3. คลิกเลือกเซลล์ C2:F2 บนหัวตาราง จะได้สูตร =C2:F2
4. พิมพ์เครื่องหมายคูณ *
5. คลิกเลือกเซลล์ B3:B5 จากข้างตาราง จะได้สูตร =C2:F2*B3:B5
6. จากนั้นแทนที่จะกดปุ่ม Enter ให้กดปุ่ม **Ctrl+Shift+Enter** พร้อมกันทีเดียว 3 ปุ่ม จะทำให้เกิดเครื่องหมายวงเล็บปีกกาปิดท้ายสูตรให้เองกลายเป็น {=C2:F2*B3:B5}

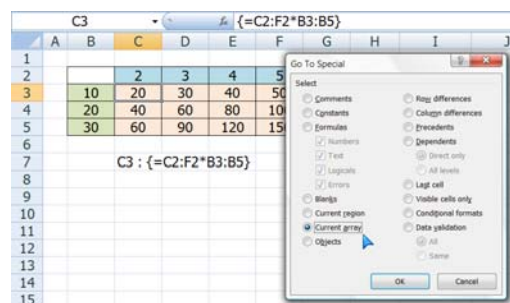
คำอธิบาย

- สาเหตุที่ต้องเลือกพื้นที่ตาราง C3:F5 ไว้ก่อนแทนการเลือกเซลล์ C3 เพียงเซลล์เดียว เพื่อให้ผลลัพธ์ที่ได้จากสูตร Array คำนวณกระจายค่าตอบลงไปในตารางให้ครบทุกตำแหน่ง (ใน Excel 365 หากสร้างลงไป C3 เซลล์เดียวจะกระจายสูตรให้เอง)
- ลองใช้ Mouse คลิกลงไปในช่วง Formula Bar เลือกเฉพาะส่วนของสูตร C2:F2 แล้วกดปุ่ม **F9** จะพบว่าตำแหน่งอ้างอิงส่วนนี้เปลี่ยนเป็น {2,3,4,5} จากนั้นให้กดปุ่ม Esc เพื่อคืนสภาพสูตรตามเดิม

- ลองใช้ Mouse คลิกลงไปในช่วง Formula Bar เลือกเฉพาะส่วนของสูตร B3:B5 แล้วกดปุ่ม **F9** จะพบว่าตำแหน่งอ้างอิงส่วนนี้เปลี่ยนเป็น {10;20;30} จากนั้นให้กดปุ่ม Esc เพื่อคืนสภาพสูตรตามเดิม
- หากกดปุ่ม **F2** ตามด้วย **F9** จะพบว่าสูตร Array เดิมบน Formula Bar {=C2:F2*B3:B5} จะเปลี่ยนเป็น {=20,30,40,50;40,60,80,100;60,90,120,150} แล้วขอให้สังเกตว่ามีเครื่องหมาย Comma และ Semi-Colon ขึ้นอยู่ระหว่างตัวเลข (ซึ่งเครื่องหมาย Comma , นี้แสดงถึงว่าค่าถัดไปยังคงอยู่ใน Row เดิม ส่วนเครื่องหมาย Semi-Colon ; แสดงว่าค่าถัดไปต้องขึ้น Row ใหม่)
- ดังนั้น {=20,30,40,50;40,60,80,100;60,90,120,150} จึงแสดงให้เห็นว่าสูตร Array นี้คืนค่าเป็นคำตอบซึ่งมีขนาดความกว้าง 4 Column และมีความสูง 3 Row ตามขนาดพื้นที่ตั้งแต่ C3:F5 ที่ต้องเลือกไว้ก่อน เพื่อให้กระจายค่าลงไปให้ครบนั่นเอง
- หากหัวตารางมีขนาดเปลี่ยนไป ต้องลบสูตร Array ที่สร้างไว้เดิมนี้ออกก่อนแล้วจึงสร้างสูตร Array ใหม่ลงไป ซึ่งในการลบสูตร Array ทั้งนี้ หากตอนที่สร้างสูตรเกิดจากการสร้างพร้อมกันทีเดียวหลายเซลล์ จะเลือกลบสูตรทั้งบางเซลล์ไม่ได้ ต้องเลือกเซลล์ C3:F5 ทั้งหมดแล้วจึงลบสูตรทั้งพร้อมกันทีเดียวจึงจะลบได้
- เนื่องจากสูตร Array ในตัวอย่างนี้เกิดจากการสร้างพร้อมกันทั้งตารางด้วยสูตรเดียว จึงทำให้ Excel คำนวณสูตรทั้งตารางสูตรคูณเพียงครั้งเดียว ถือเป็นเคล็ดลับและลัดที่น่าสนใจเป็นอย่างยิ่งเพราะถ้าสร้างได้จะทำให้เพิ่มคำนวณเร็วขึ้นอย่างผิดหูผิดตาทีเดียว (เร็วกว่าตัวอย่างสูตรคูณซึ่งสร้างสูตรลงไปทีละเซลล์แล้วต้อง Copy ไป Paste เพื่อให้คำนวณซ้ำอีกในเซลล์อื่น)
- หากต้องการค้นหาว่าสูตร Array ที่เห็นนั้นสร้างขึ้นพร้อมกันจากเซลล์ใดถึงเซลล์ใด ให้เริ่มจากคลิกเลือกเซลล์สูตร Array นั้นๆก่อนเพียงเซลล์เดียว แล้วกดปุ่ม **F5 >**

Special > ภายในของ Current array

จะพบว่า Excel เลือกพื้นที่ตารางส่วนที่เกิดจากการสร้างด้วยสูตร Array พร้อมกันนั้นให้ทันที จากนั้นหากต้องการลบหรือสร้างสูตร Array ใหม่ลงไปก็ให้กระทำต่อพื้นที่ตารางที่เลือกให้แล้วนั้นร่วมกันเสมอ



วิธีค้นหาสูตร

เมื่อเจอปัญหาในงาน ให้ค่อยๆวิเคราะห์ปัญหาทีละขั้นตอน แบ่งปัญหาแยกออกเป็นชั้นๆ ตามลำดับการคำนวณ ตามลำดับการตัดสินใจ ควรลองสร้างแนวทางแต่ละชั้นลงบนกระดาษ ก่อน เพื่อจัดลำดับการส่งค่าไปที่ละชั้นให้ถูกต้อง แล้วลองใช้มือ ใช้เครื่องคิดเลข คิดคำนวณ อย่างคร่าวๆก่อนว่า จะใช้สูตรคำนวณอย่างไร และได้ผลลัพธ์ออกมาเป็นเท่าใด

ปัญหาที่ว่ายาก อาจยากเพราะเราทำให้มันยากขึ้นเอง เช่น คำนวณว่า ถ้าซื้อสินค้าต้นทุน 127 บาทต่อหน่วย มาปรับกำไรขึ้นอีก 13.45% จะต้องกำหนดราคาขายเท่าใด

ปัญหานี้ทำให้ง่ายลง โดยใช้เลขตัวกลมแทนตัวเลขจริงไปก่อน ลองคำนวณต้นทุน 100 บาท แทน 127 บาท ลองกำหนดกำไรเป็น 10% แทน 13.45% จะคำนวณได้ผลลัพธ์ง่ายขึ้นจริงไหม

สาเหตุที่เราปรับปัญหาให้ง่ายลงนี้ เพื่อให้มองที่ลำดับขั้นการคำนวณแทนที่จะมัววุ่นอยู่กับตัวเลข เมื่อได้ขั้นการคำนวณที่ต้องการแล้ว จึงหันมาสร้างมาค้นหาสูตร Excel ที่จะใช้ ภายหลังจากสร้างสูตรเสร็จแล้ว เราจะเปลี่ยนตัวเลขให้ยากขึ้นเท่าใดก็ได้ทั้งนั้น Excel จะคำนวณให้เสร็จในพริบตา

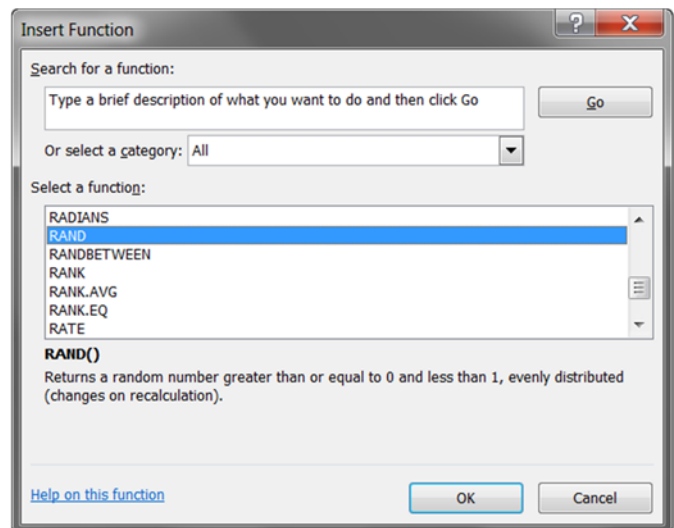
ขอให้ยึดหลักว่า Excel มีสูตรคำนวณสำหรับปัญหาของเราทุกอย่าง บางปัญหาต้องใช้สูตร ร่วมกับสูตรอื่นต่อกันไป บางปัญหาต้องใช้สูตรซ้อนสูตร แต่บางปัญหาใช้สูตรๆเดียวก็ได้ผลลัพธ์แล้ว ขั้นตอนการแก้ปัญหาก็เป็นภาระของมนุษย์ ต้องหาสูตรที่ต้องการใช้ให้เจอ

แทนที่จะเสียเงินซื้อหนังสือเรื่องสูตร ให้เปิดดูจาก Excel Help ซึ่งติดมากับโปรแกรม Excel อยู่แล้ว (หนังสือเรื่องสูตรที่ขายกัน ส่วนมากมักลอกคำอธิบายจาก Excel Help ไปใช้ มีน้อยเล่มมากที่เขียนอธิบายด้วยตัวเอง)

วิธีที่ 1 ค้นหาตามชื่อสูตร

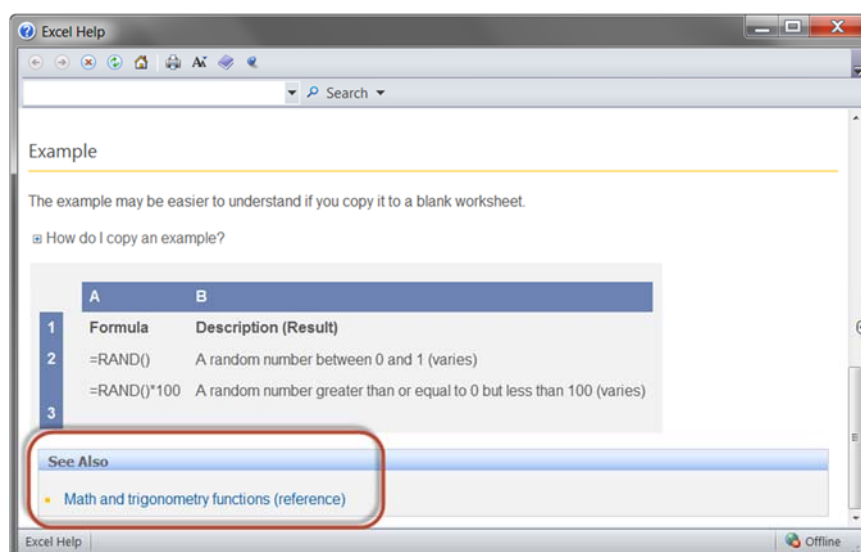
เลือกเซลล์ว่างๆที่ยังไม่มีสูตรใดๆ แล้วใช้คำสั่ง Formulas > Insert Function หรือคลิกที่ปุ่ม fx บน Formula Bar

ถ้าไม่รู้ว่สูตรนั้นๆอยู่ในกลุ่มไหน ให้คลิกเลือก All ในช่อง select a category แล้วคลิกลงไปในช่วง Select a function ตรงสูตรใดก็ได้ แล้วพิมพ์ตัวอักษรนำของสูตรนั้นๆลงไป Excel จะเลื่อนไปหาชื่อสูตรตามตัวอักษรที่ต้องการให้ทันที ส่วนด้านล่างได้จะแสดงรูปแบบของสูตร หรือคลิกที่คำว่า Help on this function ด้านล่างซ้ายสุดเพื่อเปิดดูรายละเอียดของสูตร



วิธีที่ 2 ค้นหาสูตรที่ใช้งานทดแทนกัน

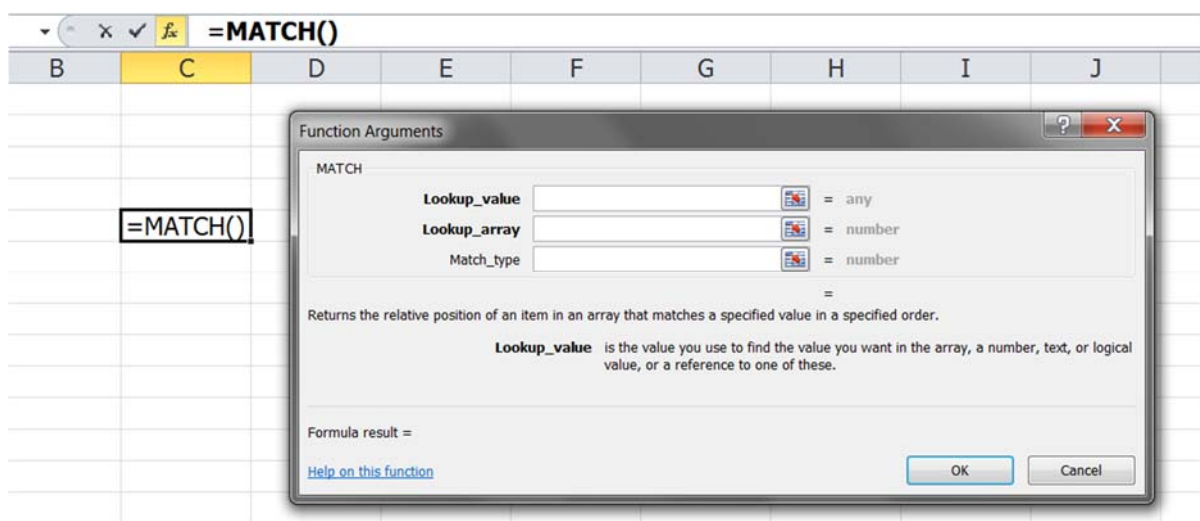
เมื่อเปิดสูตรที่ต้องการ แล้วเลื่อนจอลงไปด้านล่างสุด ให้คลิกคำว่า See Also เพื่อเปิดรายชื่อสูตรที่เกี่ยวข้องกัน



ควรรอบรู้สูตรซึ่งมีความเกี่ยวข้องกันทั้งหมดก่อน แล้วจึงตัดสินใจเลือกใช้สูตรซึ่งเหมาะสมกับปัญหาที่สุด

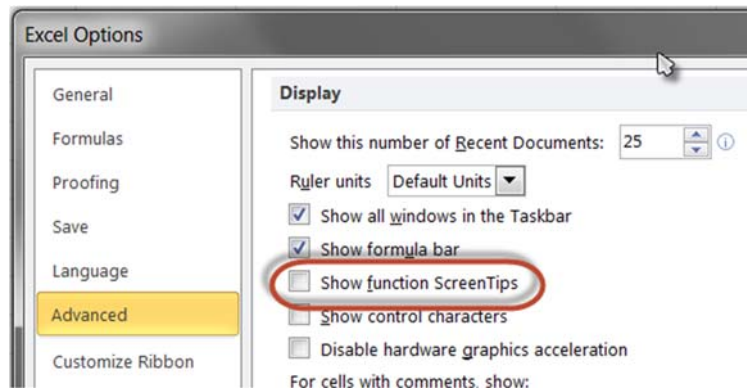
วิธีที่ 3 ค้นหาตัวแปรที่ใช้ในสูตร

ให้เริ่มพิมพ์ = ตามด้วยชื่อสูตรที่ต้องการใช้งานก่อน แล้วกดปุ่ม Ctrl+a เพื่อเปิด Help สำหรับช่วยเลือกตัวแปรในสูตรให้เอง



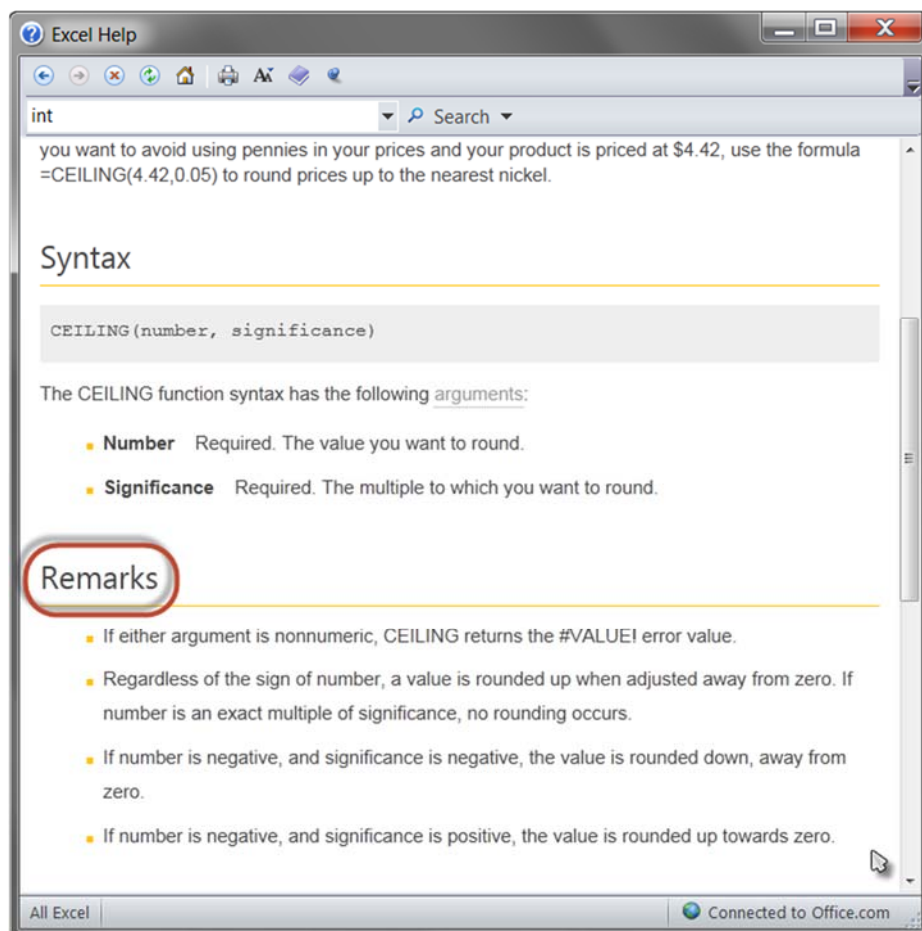
ไม่แนะนำให้ใช้ Function Screen Tips เพราะคำอธิบายที่ปรากฏขึ้นจะทับแนวของหัว column ทำให้มองไม่ว่าเป็น column ไต โดยตัดกาช่อง Show Function ScreenTips จากการสั่ง File > Options > Advanced (และควรสั่ง File > Options > Formulas > ตัดกาช่อง Formula AutoComplete ทิ้งไปด้วยเพราะ Excel จะพยายามเดาชื่อสูตรที่พิมพ์ลงไปให้)





วิธีที่ 4 ค้นหาข้อจำกัดของสูตร

เมื่อเปิดดูรายละเอียดของสูตรใน Excel Help แล้วให้ดูข้อจำกัดของสูตรได้จาก Remarks ซึ่งอธิบายข้อแม้หรือข้อจำกัดในการใช้งานของสูตรเอาไว้



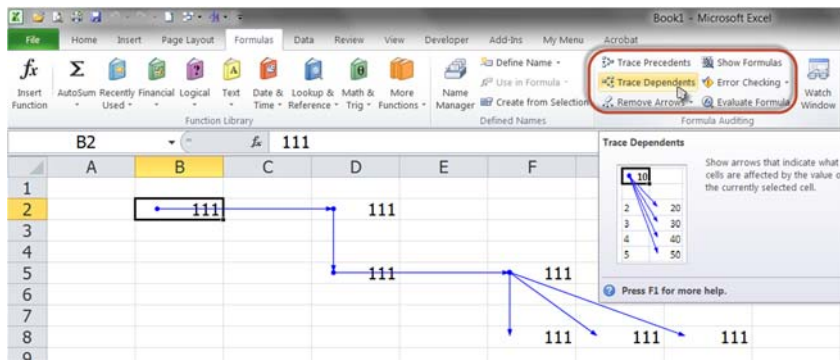
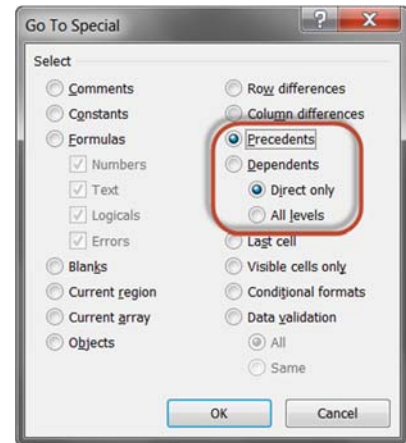
วิธีแกะสูตร

ลองพิจารณาสูตรคำนวณหาเวลาสิ้นสุดในงาน Material Requirements Planning ต่อไปนี้

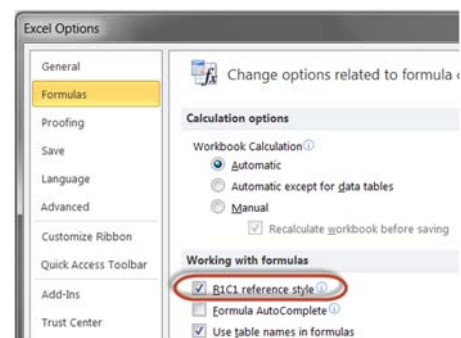
=INDEX(Begin,COUNTIF(Cum_Hour,"<"&(I10+((INDEX(Cum_Hour,MATCH(VLOOKUP(J10,Begin,1),Begin,1),1)-INDEX(Working_Hour,MATCH(VLOOKUP(J10,Begin,1),Begin,1),1))+((J10-VLOOKUP(J10,Begin,1))*24))))+1,1)+((I10+((INDEX(Cum_Hour,MATCH(VLOOKUP(J10,Begin,1),Begin,1),1)-INDEX(Working_Hour,MATCH(VLOOKUP(J10,Begin,1),Begin,1),1))+((J10 - VLOOKUP(J10, Begin, 1))*24)))- (INDEX(Cum_Hour,(COUNTIF(Cum_Hour,"<"&(I10+((INDEX (Cum_Hour,MATCH(VLOOKUP(J10,Begin,1),Begin,1),1)-INDEX(Working_Hour,MATCH(VLOOKUP(J10, Begin,1),Begin,1),1))+((J10-VLOOKUP(J10,Begin,1))*24))))+1,1)-INDEX(Working_Hour,(COUNTIF(Cum_Hour,"<"&(I10+((INDEX(Cum_Hour,MATCH(VLOOKUP(J10,Begin,1),Begin,1),1)-INDEX(Working_Hour,MATCH(VLOOKUP(J10,Begin,1),Begin,1),1)-INDEX(Working_Hour,MATCH(VLOOKUP(J10,Begin,1),Begin,1),1))+((J10-VLOOKUP(J10,Begin,1))*24))))+1,1)))/24)

- เริ่มจากสังเกตเครื่องหมายเท่ากับ เครื่องหมายวงเล็บ และเครื่องหมายลูกน้ำ ซึ่งเป็นองค์ประกอบแยกโครงสร้างของสูตร
- สังเกตตัวอักษรตัวใหญ่ เป็นชื่อสูตรและตำแหน่งอ้างอิง
- เมื่อกดปุ่ม F2 หรือคลิกลงไปในสูตรบน formula bar จะพบว่า Excel เปลี่ยนสีแต่ละส่วนของสูตร พร้อมกับมีกรอบสีตรงกันล้อมรอบเซลล์ เพื่อชี้ให้เห็นตำแหน่งของเซลล์ซึ่งถูกนำไปอ้างอิงในสูตร
- ให้คลิกลงบนสูตร แล้วใช้แป้นพิมพ์เลื่อนลูกศรผ่านเครื่องหมายวงเล็บ จะพบว่าเครื่องหมายวงเล็บคู่กันทั้งด้านหน้าและด้านหลัง จะเปลี่ยนเป็นสีดำเข้มประมาณ 2 วินาที เพื่อบอกว่าเป็นวงเล็บคู่เดียวกัน
- เมื่อพบวงสูตรที่เป็นวงเล็บชุดเดียวกันแล้ว ให้คลิกเลือกสูตรนั้นในช่อง Formula Bar แล้วกดปุ่ม F9 จะเห็นผลลัพธ์เฉพาะส่วนที่เลือกไว้ แล้วให้กดปุ่ม Esc เพื่อย้อนกลับไปแสดงสูตรตามเดิม
- ให้ double คลิกลงไปในเซลล์สูตร จะพบว่า Excel ย้ายตำแหน่งตัวชี้ไปเลือกเซลล์ต้นทางที่ใช้ในสูตร ทั้งนี้ต้องตัดกาช่อง Allow editing directly in Cells ในคำสั่ง File > Options > Advanced ไว้ก่อน และเมื่อไปยังเซลล์ต้นทางแล้วหากต้องการย้ายกลับมายังเซลล์สูตรเดิม ให้กดปุ่ม F5 แล้วกดปุ่ม Enter
- ควรใส่วรรคระหว่างเครื่องหมายที่ใช้ในสูตร เพื่อแยกให้ห่างออกจากกัน จะได้สะดวกในการแกะสูตร หากต้องการแยกสูตรขึ้นบรรทัดใหม่ในสูตรเดิม ให้กด Alt+Enter
- หากอยู่ระหว่างการแก้ไขสูตร แต่ยังไม่เสร็จ ให้พิมพ์เครื่องหมายคำพูดใส่ด้านหน้าเครื่องหมายเท่ากับ เพื่อให้สูตรกลายเป็นตัวอักษรไปก่อน หากไม่ใช้วิธีนี้ ต้องคอยระวังอย่ากด Enter เพราะ Excel จะรับสูตรลงไปและเตือนข้อผิดพลาดหรือแก้ไขสูตรต่างไปจากที่ต้องการ

- ควรพิมพ์ Comment กำกับเซลล์ โดยคลิกขวาเข้าไปในเซลล์ที่ต้องการ แล้วเลือกเมนูย่อย Insert Comment เพื่ออธิบายสูตรที่ใช้งานว่ามีความเป็นมาอย่างไร
- สูตรซึ่งอยู่ในวงเล็บในสูตร จะส่งค่าต่อมาให้สูตรส่วนนอก ดังนั้นจึงควรแกะสูตรจากนอกเข้าไป เพื่อเข้าใจกับความหมายของสูตรทั้งหมดก่อน แล้วแกะสูตรจากในออกมาอีกครั้งเพื่อไล่ลำดับการคำนวณส่งค่าในสูตรเซลล์หนึ่งๆ
- สูตรบางสูตรไม่ได้อยู่ในเซลล์ แต่เป็น Formula Name ให้ดูได้จากคำสั่ง Formulas > Name Manager เพื่อเข้าไปดูและแก้ไขสูตร
- ถ้าต้องการค้นหาเซลล์ต้นทางหรือปลายทางที่เชื่อมโยงกันด้วยสูตร ให้เลือกเซลล์สูตรแล้วกดปุ่ม F5 > Special > ทางช่อง Row Differences หรือ Column Differences หรือเลือก Precedents / Dependents และเลือก Direct only เพื่อหาเซลล์ถัดไป แล้วกดปุ่ม OK จากนั้นให้กดปุ่ม F4 ต่อไปเรื่อยๆเพื่อสั่งให้ Excel ทวนคำสั่งค้นหาเข้าไปเรื่อยๆจนกว่าจะหมดการเชื่อมโยง
- ถ้าต้องการลากเส้นลูกศรแสดงความเชื่อมโยงส่งค่าระหว่างสูตร ให้ใช้เมนู Formulas > Formula Auditing > Trace Precedents/Dependents



- แสดงสูตรในเซลล์ให้เห็นแทนการแสดงผลลัพธ์ ให้ใช้เมนู Formulas > Formula Auditing > Show Formulas และใช้คู่กับคำสั่ง File > Options > Formulas > ทางช่อง R1C1 reference style เพื่อแสดงรูปแบบตำแหน่งอ้างอิงว่าเป็นแบบเดียวกันหรือไม่ และแก้คืนสู่สภาพเดิมเมื่อตรวจสอบสูตรเรียบร้อยแล้ว



วิธีสร้างสูตรซ้อนสูตร

จุดประสงค์ของการนำสูตรมาซ้อนสูตร เพื่อให้สูตรๆเดียวสามารถคำนวณให้ผลลัพธ์ได้เลยทันที โดยไม่ต้องพึ่งพาสูตรในเซลล์อื่นๆ ทำให้เกิดประโยชน์ดังนี้

- ช่วยจำกัดเซลล์ที่ต้องใช้เพื่อสร้างสูตร
- ช่วยให้ค้นหาเซลล์สูตรได้ง่ายเนื่องจากสูตรถูกนำมารวมกันไว้เพียงที่เดียว
- หากมีจำนวนเซลล์ที่มีสูตรยาวๆจำนวนไม่มากนัก จะทำให้ file มีขนาดเล็กลง เนื่องจากไม่ต้องเปลืองเซลล์หรือ sheet เพื่อแยกเก็บสูตร และช่วยให้คำนวณรวดเร็วขึ้น

ส่วนข้อเสียของการสร้างสูตรซ้อนสูตรก็คือ สร้างยาก และทำให้แก้ไขสูตรในภายหลังยากขึ้น จึงต้องเปรียบเทียบข้อดีข้อเสียดังกล่าวนี้เสมอ หากสร้างเองใช้เองแก้ไขเองคงไม่ต้องห่วงนัก แต่ถ้าสร้างสูตรแล้วทิ้งไว้ให้คนอื่นแก้ไข คงต้องลืมนึกไปเลยว่า จะเริ่มต้นแก้ไขกันยังงัยดี ส่วนมากมักสร้างสูตรขึ้นใหม่เลยดีกว่า

หลักการสร้างสูตรซ้อนสูตร

1. ควรรวมกลุ่มของสูตรซึ่งคำนวณเป็นลำดับต่อเนื่องกันไว้เป็นสูตรเดียว และแยกขั้นตอนซึ่งไม่ต่อเนื่องออกเป็นอีกสูตรหนึ่ง
2. ควรแยกสูตรออกให้ทำงานค้นหาซึ่งมีความหมายสมบูรณ์ในขั้นนั้นๆ และง่ายต่อการตรวจสอบว่า ลำดับการคำนวณขั้นใดผิดพลาด จะได้สะดวกต่อการแก้ไขให้ถูกต้อง
3. ย่อรวมทุกขั้นตอนของการคำนวณไว้ในสูตรๆเดียวจนยาวเกินไป

วิธีสร้างสูตรซ้อนสูตร

ให้สร้างสูตรซึ่งควรเป็นสูตรวงเล็บในสุดก่อนสูตรอื่น สูตรนี้มักใช้คำนวณหาค่าพื้นฐานแรกสุด ซึ่งจะถูกนำไปใช้เป็นตัวแปรในสูตรอื่นถัดไป เช่น สร้างสูตรคำนวณหาเลขที่บรรทัด ด้วยสูตร
`=MATCH(MinimumCost, TotalCost, 0)`

จากนั้นจึงสร้างสูตรอื่นคร่อมข้างนอกเพื่อนำผลลัพธ์จากสูตร Match ไปใช้ เช่น

`=INDEX(LotSize, MATCH(MinimumCost, TotalCost, 0), 1)`

ถ้าเป็นสูตรช่วยการตัดสินใจ เช่น สูตร IF ให้พิจารณาหาสูตรซึ่งใช้ตัดสินใจอื่นๆ มาซ้อนเข้าไปในสูตรเดิมเมื่อถึงตัวแปรซึ่งต้องการใช้สูตรอื่นคำนวณค่าให้ เช่น ต้องการตัดสินใจคะแนน 0 - 50 ให้เกรด C ถ้าคะแนน 51 - 80 ให้เกรด B แต่ถ้าเกินกว่า 80 ให้เกรด A ให้สร้างสูตรหาเกรด A ให้ได้ก่อน

=IF(คะแนน>80,"A","BC")

แล้วค่อยพิจารณาต่อว่า จะตัดสินเกรด B C ต่อไปด้วยสูตร IF อย่างไร

=IF(คะแนน>80,"A",IF(คะแนน>50,"B","C"))

ถ้าไม่ถนัดสร้างสูตรต่อกันซ้อนกันไปเลยทีเดียว ให้สร้างสูตรแยกแต่ละขั้นตอนออกไปลงในเซลล์แยกจากกันก่อน แล้วจึง copy ตัวสูตร ไป paste ทับในตำแหน่งอ้างอิง เช่น

A1: =NOW()

A2: =DAY(A1)

ให้ copy สูตร NOW() จากเซลล์ A1 ไปทับคำว่า A1 ในเซลล์ A2 จะได้สูตรซ้อนกันเป็น =DAY(NOW())

วิธี copy สูตร ให้คลิกเลือกส่วนของสูตรที่ต้องการจากสูตรบน Formula Bar เช่น เลือกคำว่า NOW() แล้วคลิกขวาเลือก copy (หรือกด Ctrl+c) แล้วกด Esc หนึ่งครั้งเพื่อออกจากการเลือก แล้วจึงไปคลิกเลือกคำว่า A1 บน Formula Bar ของสูตรในเซลล์ A2 แล้วคลิกขวาสั่ง paste (หรือกด Ctrl+v)

วิธีสร้างชื่อซ้อนชื่อ

ใช้คำสั่ง Formulas > Define Name เพื่อดังชื่อให้กับสูตร หลังจากนั้นจึงนำชื่อนั้นไปซ้อนในชื่ออื่นต่อไปอีก เช่น สูตรในงานวางแผนการผลิต ซึ่งใช้เวลาเริ่มงาน และเวลาที่ใช้ทำงาน คำนวณหาเวลาสิ้นสุดงาน ต่อไปนี้

Calc1 =VLOOKUP(RelativeStart, Begin, 1)

Calc2 =INDEX(Working_Hour, MATCH(Calc1, Begin, 1), 1)

Calc3 =INDEX(Cum_Hour, MATCH(Calc1, Begin, 1), 1)

Calc4 =Calc3 - Calc2

Calc5 =RelativeStart - Calc1

Calc6 =Calc4 + (Calc5*24)

Calc7 =RelativeHour + Calc6

Calc8 =COUNTIF(Cum_Hour, "<" & Calc7) + 1

Calc9 =INDEX(Working_Hour, Calc8, 1)

Calc10 =INDEX(Cum_Hour, Calc8, 1)

Calc11 =Calc10 - Calc9

Calc12 =(Calc7 - Calc11) / 24

Calc13 =INDEX(Begin, Calc8, 1)

Calc14 =Calc13 + Calc12

วิธีนำชื่อที่ตั้งไว้มาใส่ในสูตร ให้กดปุ่ม F3 เพื่อเลือกชื่อที่ตั้งไว้ เพื่อสร้างสูตรลงไปในเซลล์ก่อน แล้วจึง Copy สูตรไปใส่ลงในช่องสูตรของ Define Name

เวลาใช้งานก็นำสูตรชื่อ Calc14 ไปใช้เท่านั้น ก็จะคำนวณสูตรทั้งหมดตั้งแต่ Calc1 – Calc13 ให้อัตโนมัติ

เรื่องที่ไม่ค่อยรู้กันเกี่ยวกับสูตร

โปรแกรม Microsoft Excel รุ่นใหม่ มีได้มีเสน่ห์อยู่ที่รูปร่างหน้าตาบนหน้าจอที่ประกอบไปด้วยแถบribbonแบบใหม่(ที่ยังไม่ค่อยคุ้นเคย) และมีพื้นที่ตารางซึ่งมีขนาดใหญ่มากขึ้นกว่าเดิมหลายเท่าเพียงเท่านั้น แต่สาเหตุที่ทำให้ Excel มีเสน่ห์ กลายเป็นโปรแกรมยอดนิยมมานานแสนนาน ก็เพราะมีองค์ประกอบสำคัญ นั่นคือ Excel ตั้งแต่รุ่นแรกๆได้จัดเตรียมสูตรสำเร็จรูปไว้มากมายให้นำมาใช้กัน หากสูตรที่มีอยู่ยังไม่สามารถใช้คำนวณหาคำตอบที่ต้องการได้ เราก็สามารถนำสูตรที่มีอยู่นั้นมาใช้ร่วมกัน ซ้อนกัน หรือผูกสูตรเข้าด้วยกัน จนกล่าวได้ว่าเราสามารถนำ Excel หาคำตอบได้กับโจทย์การคำนวณทุกเรื่อง ปัญหาใดที่คิดว่าไม่สามารถใช้ Excel หาคำตอบ นั้นเป็นเพราะคนเราต่างหากที่ยังคิดไม่ออก ไม่ใช่ Excel หาคำตอบไม่ได้ บางปัญหาอาจต้องใช้เวลาคิดนับสิบปีกว่าจะทราบว่าต้องเอาสูตรนั้นมาซ้อนกับสูตรนี้แล้วก็จะได้คำตอบที่ต้องการออกมานั่นเอง

เคล็ดลับที่ช่วยให้สามารถใช้สูตรจนเข้าใจของ มิใช่ว่าต้องรู้จักสูตรทุกสูตรที่ Excel มีอยู่หรอก เพราะสูตรที่ใช้กันเป็นประจำนั้นมีไม่มากนัก แคร่รู้จักการใช้สูตร Sum, If, And, Or, Choose, VLookup, Match, Index, CountIF, SumIF, Offset, Indirect, และ Array Formula ให้เป็น และสามารถนำสูตรแค่นี้แหละมาใช้ร่วมกันให้เป็นก็เพียงพอแล้ว แต่กว่าจะใช้สูตรให้เป็นยังมีหลายเรื่องที่ต้องทำความเข้าใจกัน หากคิดเพียงว่าขอใช้สูตรเหล่านี้เป็นก็พอแล้ว คุณก็จะถึงทางตันในไม่ช้า โดยบทความนี้จะขอนำเรื่องที่ไม่ค่อยรู้กันเกี่ยวกับสูตรมาอธิบาย หลายเรื่องเป็นเรื่องสำคัญที่คิดกันว่าไม่สำคัญ บางเรื่องก็ง่ายแต่คิดไม่ถึงว่าเรื่องง่ายนั้นแหละยังมีอะไรหลายอย่างซ่อนอยู่อีก

ทำอย่างไรให้เก่งสูตร Excel

1. ไม่จำเป็นต้องซื้อหนังสือเกี่ยวกับสูตร เพียงแค่เปิดโปรแกรม Excel แล้วกดปุ่ม F1 ก็จะได้เปิด Excel Help ขึ้นมาค้นหาสูตรที่ Excel มีอยู่ได้แล้ว หากคุณต้องการซื้อหนังสือเกี่ยวกับสูตร ขอให้หาซื้อหนังสือที่อธิบายเรื่องสูตรได้ละเอียดกว่าที่มีใน Help (ซึ่งหาได้ยากมาก เพราะคนแต่งหนังสือมักเขียนอธิบายต่างจาก Help ไม่มาก)
2. ถ้าซื้อหนังสือ Excel มาแล้ว ขอให้เปิดอ่านตั้งแต่หน้าแรกจนถึงหน้าสุดท้ายโดยไม่ต้องเปิดคอมพิวเตอร์ เพราะถ้ามันเปิดอ่านไปทำตัวอย่างไปบนหน้าจอพร้อมกัน กว่าที่จะอ่านหมดเล่มก็จะใช้เวลาเกินปีครึ่ง (ซึ่ง Excel จะมีรุ่นใหม่ให้พวกเราได้ใช้งานกันประมาณ

ทุกปีครั้ง แล้วคุณจะต้องซื้อหนังสือ Excel รุ่นใหม่มาตั้งต้นอ่านกันตั้งแต่หน้าแรกกันใหม่อีก สุดท้ายก็ไม่สามารถรู้จัก Excel ครบทั้งหมดสักที)

3. ถ้ายังไม่ได้ซื้อหนังสือ ขอให้ค้นหาจาก Google แต่ต้องเปรียบเทียบคำอธิบายจากหลายแหล่งให้ดี เพราะยุคนี้มีคำแนะนำผิดๆหรือแนะนำสิ่งที่ไม่ควรใช้เยอะมาก
4. ขอให้พยายามรอบรู้ก่อนรู้สึก คุณควรรอบรู้ไว้ก่อนว่า Excel มีสูตรอะไรบ้าง จากนั้นเมื่อถึงเวลาทำงานจึงค่อยใช้เวลารู้สึกเรียนรู้สูตรแต่ละสูตรให้คล่องในภายหลัง
5. ตั้งหลักไว้เลยว่า Excel มีสูตรที่คุณสามารถนำมาใช้กับงานทุกอย่างของคุณได้อยู่แล้ว ถ้าไม่มีสูตรสำเร็จรูปที่คำนวณลัดหาคำตอบได้โดยตรง ให้หาทางนำสูตรหลายสูตรมาใช้งานร่วมกัน
6. เริ่มแรกที่ทดลองใช้สูตร ควรใช้กับโจทย์ง่ายๆที่ทราบคำตอบที่ถูกต้องอยู่แล้ว จะได้เข้าใจหลักการคำนวณที่ Excel ใช้ว่ามีขั้นตอนการคิดคำนวณอย่างไร และเมื่อได้คำตอบตรงกับที่ทราบ จะช่วยทำให้คุณเกิดความมั่นใจ และเกิดความกล้าที่จะนำสูตรไปใช้ในปัญหาอื่นต่อไป ลองคิดดูว่า $=2*3$ กับ $=123*4.56$ สูตรใดที่ง่ายกว่ากัน
7. ถ้าสูตรคำนวณได้คำตอบอื่นต่างจากที่ต้องการ อย่าคิดว่าตัวเองสร้างสูตรผิด แต่ควรพยายามคิดค้นหาให้ได้ว่าคำตอบที่ได้นั้นหมายถึงการคำนวณของอะไร หลายๆครั้งที่เดียวที่คุณจะได้สูตรลัดและลับแปลงๆที่คนอื่นไม่รู้จักกัน
8. อย่ารีรอที่จะสร้างสูตรให้เห็นผลลัพธ์กับตัวว่าถูกหรือผิด ถ้าคุณเอาแต่กลัวๆกลัวๆไม่ยอมสร้างสูตรสักทีก็จะมีวันเก่งสูตรได้หรอก ขอให้สร้างสูตรบ่อยๆ สูตรที่ผิดหรือถูกเหล่านั้นจะเป็นบทเรียนสอนคุณให้เก่งกว่าคนอื่นที่ไม่เคยยอมทำผิด
9. ถ้าปัญหาการคำนวณซับซ้อนยากมาก ควรแยกเซลล์ให้ใช้เซลล์หลายๆเซลล์แยกคำนวณสูตรแต่ละขั้น จากนั้นจึงค่อยหาวิธีนำสูตรมาต่อกันเป็นสูตรยาวๆสูตรเดียวในเซลล์เดียว
10. พยายามหาทางแก้ปัญหาสร้างสูตรด้วยตัวเองให้ได้ ถ้าทำได้เองแล้วคุณก็จะเกิดความรู้สึกภูมิใจ เกิดความกล้าและอยากลอง ทำให้เก่งขึ้นไปได้อีก อย่าลอกสูตรคนอื่นมาใช้ ถ้าพยายามอย่างไรแล้วคิดสูตรไม่ออก ขอให้ค้นหาคำตอบจาก Google ก่อนที่จะเอ่ยปากถามใคร

หวังว่าคำแนะนำ 10 ข้อนี้จะเป็นแนวทางช่วยให้คุณกลายเป็นคนเก่งสูตร Excel และขอฝากคนที่ชอบใช้สูตรรุ่นใหม่ที่เพิ่งเกิดขึ้นว่า อย่าเพิ่งรีบร้อนนำสูตรใหม่ๆเหล่านั้นมาใช้ เพราะตราบดีที่เพื่อนของคุณยังคงใช้ Excel รุ่นเก่ากว่าอยู่ เขาจะไม่สามารถนำแฟ้มของคุณที่สร้างไว้โดยใช้สูตรรุ่นใหม่มาคำนวณได้เลย คุณนั่นแหละที่จะต้องรับภาระย้อนกลับไปแก้ไขสูตรรุ่นใหม่ให้เป็นรุ่นเก่าเพื่อเปิดแฟ้มใน Excel รุ่นเก่าที่คนอื่นเขายังไม่เลิกใช้กัน

Formula vs Function

ทั้งสองคำนี้ถ้าแปลเป็นไทยก็แปลว่า “สูตร” เหมือนกัน โดยคำว่า Formula เป็นคำที่มีความหมายกว้างกว่า Function กล่าวคือ

- **Formula** หมายถึง สูตรทุกอย่างที่มีเครื่องหมายเท่ากับ = นำหน้า เช่น =A1 หรือ =1+2 หรือ =Now()
- **Function** หมายถึง สูตรสำเร็จรูปที่ติดมากับโปรแกรม Excel เช่น สูตร Sum, Max, Min, If, VLookup หรือ Now เป็นต้น

ดังนั้นเพื่อทำความเข้าใจให้ตรงกัน ในบทความนี้จะใช้คำว่า สูตร ซึ่งหมายถึงทั้ง Formula และ Function หรืออีกนัยหนึ่ง สูตรก็คืออะไรก็ตามที่มีเครื่องหมาย = นำหน้า ส่วนตัวผมเองจะเขียนอธิบายเรื่องสูตรในกระดาษโดยใช้ตัวใหญ่ปนตัวเล็ก เช่น Sum หรือ VLookup เพื่อช่วยให้อ่านแต่ละคำของสูตรได้ง่าย แต่เมื่อถึงคราวนำไปใช้ใน Excel โปรดพิมพ์สูตรด้วยตัวเล็กทั้งหมดเสมอ

Text vs Number

ค่าที่พิมพ์บันทึกลงไปในเซลล์หรือผลจากการคำนวณมีอยู่เพียง 3 อย่างคือ ค่าที่ถือเป็น Text ค่าที่ถือเป็น Number และค่าที่ error ไม่สามารถนำไปใช้คำนวณต่อได้

- **Text** คือ ค่าที่พิมพ์ลงไปในเซลล์หรือเกิดจากสูตรคำนวณแล้วจะขีดซ้ายของเซลล์ทันที (โดยไม่ต้องกำหนด Format) เช่น พิมพ์คำว่า abc หรือ กขค หรือแม้แต่ตัวเลขที่พิมพ์ต่อท้ายเครื่องหมายฝนทอง เช่น `123 หรือผ่านการกำหนด Format Number ให้ถือเป็นแบบ Text ซึ่งโดยทั่วไปหากเซลล์ด้านขวายังว่างอยู่ จะพบว่าถ้าเซลล์มี Text ที่ยาวกว่าความกว้างของเซลล์ จะยอมให้ Text นั้นแสดงต่อไปในเซลล์ด้านขวา
- **Number** คือ ค่าที่พิมพ์ลงไปในเซลล์หรือเกิดจากสูตรคำนวณแล้วจะขีดขวาของเซลล์ทันที (โดยไม่ต้องกำหนด Format) เช่น พิมพ์ตัวเลข 123 หรือพิมพ์วันที่ 14/2/2009 ซึ่งโดยทั่วไปถ้าเซลล์มีความกว้างไม่พอที่จะแสดงตัวเลขทั้งหมด Excel จะแสดงด้วยเครื่องหมาย ##### ข้างจนเต็มเซลล์ (แก้ไขได้โดยคลิกขวาแล้วไปเลือก **Format Cells > Alignment > กางช่อง Shrink to fit** เพื่อสั่งให้ Excel ย่อขนาดของ Font ให้มีขนาดพอดีที่จะแสดงได้เต็มความกว้างของเซลล์)
- **Error** เป็นผลที่เกิดจากการคำนวณที่ไม่สามารถนำไปคำนวณต่อได้ เช่น

- **#N/A** เมื่อสูตรหาข้อมูลที่ต้องการไม่พบ (Not Available)
- **#NAME?** เมื่อสูตรหาชื่อที่ต้องการไม่พบ เช่น =answer โดยที่ในแฟ้มนั้นไม่ได้ตั้งชื่อ Range Name ว่า answer ไว้ก่อน
- **#NULL!** เมื่อสูตรหาตำแหน่งที่ตัดกันไม่ได้ เช่น =A:A B:B
- **#NUM!** เมื่อสูตรใช้ตัวเลขที่เป็นไปไม่ได้ในการคำนวณ เช่น =Sqrt(-1)
- **#REF!** เมื่อตำแหน่งเซลล์ที่เคยอ้างไว้ในสูตรถูก delete ทิ้งไป
- **#VALUE!** เมื่อสูตรไม่สามารถคำนวณได้ เช่น นำตัวเลขคูณตัวอักษร =123*a
- **#SPILL!** เมื่อสูตรที่จะกระจายตัวถูกเซลล์ที่ใช้งานอื่นมาขวางไว้

หลักการใช้ Excel ที่ดี คือ ค่าใดๆที่เกี่ยวข้องกับตัวเลข ก็ขอให้เป็นตัวเลขที่ชัดเจนของเซลล์เสมอ ไม่ควรใช้ตัวเลขที่ขีดซ้ายของเซลล์

ขนาดแฟ้ม vs ประเภทข้อมูล

ขนาดของแฟ้มขึ้นกับประเภทของข้อมูล โดยเรียงลำดับขนาดแฟ้มจากเล็กไปใหญ่ได้ตามนี้

1. แฟ้มที่เก็บข้อมูลเป็น Number เช่น 1, 111, 123
2. แฟ้มที่เก็บข้อมูลเป็น Text เช่น a, aaa, abc
3. แฟ้มที่เก็บข้อมูลเป็นสูตรที่คืนค่าเป็น Number เช่น =1, =111, =123
4. แฟ้มที่เก็บข้อมูลเป็นสูตรที่คืนค่าเป็น Text เช่น ="a", ="aaa", ="abc"

ที่แปลกก็คือ ขนาดของแฟ้ม ไม่ขึ้นกับความยาวของค่าในเซลล์ เช่น ถ้าเซลล์มีค่าที่บันทึกไว้เป็น 1 หรือเพิ่มความยาวตัวเลขเป็น 111 หรือ 123 ก็จะมีขนาดแฟ้มเท่ากันกับ 1 ตัวเดียว หรือถ้าสร้างเป็นสูตร =1 หรือเพิ่มความยาวตัวเลขเป็น =111 หรือ =123 จะมีขนาดแฟ้มเท่ากันกับ =1

ดังนั้นถ้าต้องการลดขนาดแฟ้มให้เล็กที่สุด แทนที่จะพิมพ์เลข 1 แล้วสร้างสูตรบวกเลขต่อกันไปทีละเซลล์ให้กลายเป็นลำดับตัวเลข 1, 2, 3, 4, 5 ก็ควรพิมพ์เลขลงไปในเซลล์โดยตรงแทนการสร้างสูตร และสูตรใดๆที่คำนวณเสร็จเรียบร้อยแล้วและไม่มีการเปลี่ยนแปลงค่าตอบต่างจากเดิมอีกแล้ว เราก็ควร Copy แล้ว Paste Special ให้กลายเป็น Value จะทำให้แฟ้มเล็กลงไปมากที่สุด

ส่วนเซลล์สูตรที่เกิด Error ขึ้นก็ควรใช้สูตร IF ปรึบ Error ให้เปลี่ยนเป็นค่าอื่นแทน ดังนี้

- ในกรณีที่ A1 คำนวณเป็น Number ให้ใช้ `=IF(IsError(A1), 0, A1)`
- ในกรณีที่ A1 คำนวณเป็น Text ให้ใช้ `=IF(IsError(A1), "ค่าเตือนใดๆก็ได้", A1)`

ซึ่งตัวผมเอง นิยมใช้สูตร `=IF(IsError(A1), 0, A1)` ทั้งในกรณีที่เดิมคําคำนวณเป็น Number หรือ Text ก็ตาม เพราะสามารถใช้ Format ปรับการแสดงผลเลข 0 ให้แสดงเป็นค่าอื่นหรือแม้แต่ทำให้ดูเหมือนกลายเป็นเซลล์ว่างได้ต่อไป

สูตร `=IFERROR(A1,0)` แม้สั้นกว่าแต่ไม่เปิดให้กรณีที่ไม error ให้เปลี่ยนเป็นค่าอื่น

ตำแหน่งเซลล์ที่เลือกสำหรับสร้างสูตร

เห็นหัวข้อนี้แล้วคงสงสัยกันว่าตำแหน่งเซลล์ที่เลือกสร้างสูตรลงไปนั้นเป็นประเด็นที่ต้องใส่ใจกันด้วยหรือ ถ้าคิดกันง่ายๆว่า “ในเมื่อฉันต้องการจะสร้างสูตรลงไปตรงนี้ก็ตรงตามใจฉัน ฉันจะสร้างสูตรลงไปตรงนี้เสียอย่าง ใครจะทำไม” ก็แสดงว่าคุณมองข้ามอะไรที่เป็นเรื่องสำคัญอย่างหนึ่งไปแล้ว

ตำแหน่งเซลล์ที่คุณสร้างสูตรหาคำตอบลงไป สามารถใช้เป็นชีตตายให้กับงานของคุณทีเดียว เพราะถ้าอยู่มาวันหนึ่ง เมื่อย้ายเซลล์สูตรไปที่ตำแหน่งอื่น สูตรที่เคยคำนวณหาคำตอบได้ตลอดมา อาจคำนวณคำตอบที่ผิดหรือแสดงผลลัพธ์แตกต่างไปจากเดิมก็ได้

วิธีทดสอบสูตรที่คุณหรือใครสร้างขึ้นว่าเป็นสูตรที่ใช้ได้หรือไม่ ให้ทดลองย้ายเซลล์สูตรไปที่อื่น ย้ายไปเรื่อยๆหลายๆตำแหน่ง พร้อมกันนั้นทดลองย้ายตารางข้อมูลที่ใช้อ้างอิงไปที่อื่นด้วย ลอง Insert Row หรือ Columnแทรกตารางข้อมูลเข้าไปอีก ถ้าสูตรนั้นยังคงหาคำตอบได้เหมือนเดิม แสดงว่าสูตรนั้นใช้งานได้

ทราบไหมว่าถ้าตารางข้อมูลที่เก็บค่าอยู่ในตารางช่วง B2:D5 คุณควรสร้างสูตรคำนวณซึ่งเกี่ยวข้องกับการใช้ข้อมูลในตารางนี้ไว้ที่เซลล์ใด ซึ่งจะรับประกันว่าสูตรที่สร้างขึ้นจะสามารถนำไปใช้งานได้ทุกที่

คำตอบก็คือ ห้ามสร้างสูตรในแถวเดียวกับตารางข้อมูล นั่นคือ อย่าสร้างสูตรในช่วง Column B ถึง D และช่วง Row 2 ถึง 5

สาเหตุที่ห้าม เพราะการสร้างสูตรใดๆที่มีตำแหน่งอ้างอิงไปยังค่าในแถวเดียวกับตารางข้อมูล อาจคำนวณคำตอบเป็นค่าในตารางตามแนวเดียวกันก็ได้

หากสร้างสูตรในเซลล์ซึ่งไม่ได้อยู่ในแนวเดียวกับช่วง Column B ถึง D และช่วง Row 2 ถึง 5 แล้วค้นหาทางปรับสูตรนั้นให้คำนวณหาคำตอบที่ต้องการ ยอมรับประกันได้ว่า สูตรนั้นเป็นสูตรที่ใช้งานได้จริงและจะยังคงคืนค่าเป็นคำตอบเดิมไปตลอด ไม่ว่าจะย้ายเซลล์สูตรไปที่ตำแหน่งอื่นก็ตาม

พอพบว่าสูตรทำงานได้จริงแล้ว จากนั้นจะย้ายเซลล์สูตรไปวางไว้ในแนวคู่ขนานกับตารางข้อมูลที่ใช้ในการคำนวณก็ไม่ต่างกัน

สูตรก็มีจังหวะหายใจด้วยนะ

ผู้ที่ใช้ Excel เป็นประจำต้องรู้จักสูตรต่อไปนี้เป็นอย่างดี $=IF(C2=0, 0, B2/C2)$

สูตรข้างต้นนี้คือสูตรที่ใช้สำหรับปรับสูตรที่นำตัวเลขมาหารกันไม่ให้แสดงค่าเตือน Error ว่า #DIV/0! ขึ้นมา

เมื่อเซลล์ C2 ซึ่งเป็นเลขตัวหารมีค่าเป็น 0 ให้คืนค่า 0 เป็นตัวเลขคำตอบแทน แต่ถ้าเซลล์ C2 ไม่ได้เป็นเลข 0 ก็ย่อนำ C2 มาหาร B2 ได้ตามปกติ ผู้ที่สร้างสูตรหารกันจะหลีกเลี่ยงสูตรดังกล่าวนี้ไม่ได้

ขอถามว่า เวลาที่คุณสร้างสูตร คุณมีพฤติกรรมในการสร้างสูตรกันอย่างไร คุณจะพิมพ์ลงไปในเซลล์ว่า $=if(c2=0,0,b2/c2)$ รวดเดียวให้ครบทั้งหมดแล้วจึงกดปุ่ม Enter หรือไม่ หรือจะค่อยๆ พิมพ์ $=if$ แล้วหยุดหายใจพักหนึ่งแล้วจึงพิมพ์ $c2=0$ ต่อ แล้วก็หยุดพักหายใจอีกเฮือกหนึ่งจึงค่อยๆ พิมพ์ส่วนที่เหลือต่อ

ข้อผิดพลาดที่พบบ่อยในการสร้างสูตรของคนที่ไม่ค่อยรู้การใช้ Excel ก็คือ มักหลงลืมเครื่องหมายวงเล็บ พิมพ์เครื่องหมาย Comma ขาดไป บางครั้งพิมพ์สลับที่ บางครั้งพิมพ์ขาด บางครั้งพิมพ์เกิน ทำให้เมื่อกดปุ่ม Enter ลงไปแล้ว Excel ต้องเตือนขึ้นมาบอกให้คุณทราบว่าสูตรที่สร้างนั้นผิด

ผู้สร้างสูตรต้องมีสมาธิพอตัว สมาธิที่ว่่านี้อาจเกิดจากการรู้จักว่าเมื่อใดจะหยุดเมื่อใดจะสร้างต่อ ถ้ารู้หลักจังหวะหายใจจะช่วยให้คุณสร้างสูตรได้ง่ายขึ้น ไม่ว่าสูตรสั้นๆ หรือสูตรยาวๆ ก็ใช้หลักจังหวะหายใจนี้กันทั้งนั้น

จังหวะหายใจที่ว๊านี้ ไม่ใช่จังหวะหายใจของ Excel หรือกะ Excel มันไม่ได้มีการหายใจ เหมือนกับมนุษย์ แต่เป็นจังหวะหายใจของคุณนั่นแหละ ต้องรู้ว่าเมื่อใดที่ควรจะหยุดแล้วหายใจ ลึกเอือกหนึ่งแล้วจึงสร้างสูตรต่อ ซึ่งเรื่องการหายใจนี้ไม่ต้องลอกจังหวะการหายใจให้เหมือนกับ ผมก็ได้ ตัวใครตัวมัน ขอให้เลือกจังหวะหายใจที่ตัวเองคิดว่าถนัดก็แล้วกัน

ขอยกสูตรทั้งหมดมาดูกันอีกครั้ง

=IF(C2=0, 0, B2/C2)

มาดูกันว่าถ้าเป็นวิธีของผม จะสร้างสูตรนี้โดยหยุดพักเพื่อหายใจกันตรงไหนบ้าง เอาหายใจเข้า แล้วยะ จากนั้นเริ่มสร้างสูตรโดยใช้วิธีตามลำดับต่อไปนี้

1. พิมพ์เครื่องหมายเท่ากับ
2. พิมพ์คำว่า if(
3. หยุดเพื่อหายใจ จากนั้นสร้างสูตรต่อด้วยขั้นต่อไป
4. หยิบเมาส์คลิกเซลล์ C2
5. แล้วพิมพ์ =0, ต่อ
6. หยุดเพื่อหายใจ
7. ในใจอ่านสูตรที่ได้ตามนี้ =if(C2=0, ว่าถ้าเซลล์ตัวหามีค่าเท่ากับ 0 ละก็
8. พิมพ์ 0, ต่อ
9. หยุดเพื่อหายใจ
10. ในใจอ่านสูตรที่พิมพ์ 0, ว่า ให้คำตอบที่ได้เป็น 0 แทนนะ จากนั้น
11. หยิบเมาส์คลิกเซลล์ B2
12. พิมพ์เครื่องหมายหาร /
13. แล้วหยิบเมาส์คลิกเซลล์ C2
14. หยุดเพื่อหายใจ
15. ในใจอ่านสูตร B2/C2 ที่ได้มาว่า แต่ถ้าตัวหารไม่ใช่เลข 0 ให้คำตอบที่ได้เป็นสูตรหาร กันตามปกติ
16. จากนั้นจึงกดปุ่ม Enter รับสูตรลงไปในเซลล์ โดยไม่ต้องพิมพ์เครื่องหมายวงเล็บปิด เพราะถ้ามีวงเล็บเปิดตัวเดียว Excel จะพิมพ์เครื่องหมายวงเล็บปิดให้เอง

จับหลักได้ไหมเอ่ย ขอให้สังเกตว่าให้หยุดเพื่อหายใจได้หลังจากพิมพ์จบตรงเครื่องหมายนั้นๆ ใจ ถ้าใช้เครื่องหมายจุดจุดแทนช่วงหยุดเพื่อหายใจ จะได้สูตรตามแบบนี้

=if(.....C2=0,.....0,..... B2/C2

ฝึกหัดเพื่อพักหายใจให้ได้จังหวะ วิธีนี้จะช่วยลดข้อผิดพลาดหลงลืมการใส่เครื่องหมายวงเล็บหรือเครื่องหมาย Comma ในการสร้างสูตรไปได้เยอะทีเดียว

ธรรมชาติของสูตร

ถ้าเข้าใจธรรมชาติต่อไปนี้ของสูตร จะทำให้เราใช้สูตร Excel ได้ง่ายโดยไม่ต้องท่องจำและลดข้อผิดพลาดเมื่อนำสูตรมาใช้

1. ชื่อสูตรเป็นตัวอักษรภาษาอังกฤษตัวใหญ่ เช่น IF, MAX, MIN ดังนั้นเวลาพิมพ์สูตรให้พิมพ์ตัวเล็ก พอกดปุ่ม Enter แล้วจะพบว่า ถ้าเราสะกดชื่อสูตรถูกต้อง Excel จะเปลี่ยนตัวอักษรจากตัวเล็กไปเป็นตัวใหญ่ทั้งหมดให้ทันที
2. เราสามารถพิมพ์สูตรโดยถ้าในสูตรมีเครื่องหมายวงเล็บเปิดเพียงวงเล็บเดียว พอกดปุ่ม Enter จะพบว่า Excel พิมพ์เครื่องหมายวงเล็บปิดให้เอง แต่ถ้ามีวงเล็บเปิดหลายตัว เราต้องใส่เครื่องหมายปิดวงเล็บให้ครบทุกคู่ เพราะถ้ากดปุ่ม Enter ลงไป Excel จะพยายามคิดใส่เครื่องหมายวงเล็บปิดให้ครบแต่อาจใส่วงเล็บปิดผิดคู่ให้ก็ได้
3. วงเล็บในสูตรต้องใส่ให้ครบคู่และถูกตำแหน่ง มีวงเล็บเกินดีกว่าขาดวงเล็บ
4. ถ้าไม่ใส่วงเล็บ Excel จะไล่คำนวณตามลำดับดังนี้ ยกกำลังก่อนแล้วตามด้วยคูณหรือหาร แล้วตามด้วยบวกหรือลบ (คูณหาร หรือบวกลบ มีศักดิ์ศรีเท่ากัน จะไล่คำนวณจากซ้ายไปขวาแล้วแต่ว่าเครื่องหมายใดมีก่อน) ดูตัวอย่างท้ายข้อ 7
5. โครงสร้างสูตร Excel ถูกสร้างขึ้นมาเลียนแบบลำดับความคิดของมนุษย์ ดังนั้นถ้าเราเข้าใจตัวเองว่าเรามองเราคิดที่ละขั้นอย่างไร เราก็จะสร้างสูตรได้ง่ายโดยไม่ต้องท่อง เช่น เวลาเราคิดเรื่องเงื่อนไขว่า ถ้าทำดี ต้องได้ดี แต่ถ้าไม่ทำดี ต้องได้เรื่องอื่นๆ ตัวสูตร IF ของ Excel ก็จะมีลำดับในโครงสร้างสูตรตามแบบที่เราคิดออกมาเป็น =IF(ทำดีใช่ไหม, ได้ดี, ได้อื่นๆ) ขอให้อ่านรายละเอียดประเด็นนี้จากบทความเรื่อง เมื่อสูตร Excel มีชีวิตจิตใจ หรือจากลิงค์ www.excelexperttraining.com/blogs/archives/z300-ExcelCore000706.php
6. ในโครงสร้างสูตร ถ้ามีการอ้างอิงกับตำแหน่งเซลล์ตามแนวนอนและแนวตั้ง จะต้องใส่ตำแหน่งอ้างอิงตามแนวนอนไว้ก่อนแนวตั้งเสมอ ดูได้จากสูตร Index หรือสูตร Offset (ดังนั้นเราควรสร้างสูตรคำนวณบวกลบคูณหารอื่นๆ ให้ใช้ตำแหน่งอ้างอิงตามแนวนอนก่อนแนวตั้งด้วยเช่นกัน เพื่อเป็นแบบแผนช่วยให้เข้าใจที่ไปที่มาและแกะสูตรได้ง่าย)

7. ถ้าในโครงสร้างสูตรประเภทฐานข้อมูล มีตัวแปร (Option) ที่อาจใส่หรือไม่ใส่ก็ได้ เช่นสูตร VLookup หรือสูตร Match ขอให้ยึดหลักว่า

7.1. ถ้าไม่ใส่ Option หรือละไว้ หรือใส่ตัวเลข 1 นั้น สูตรนี้จะใช้กับตารางข้อมูล ที่เรียงค่าจากน้อยไปมาก (จำไว้ว่าเรียงตามธรรมชาติต้องเรียงจากน้อยไปมากเสมอ)

7.2. ถ้าใส่ Option เป็น False หรือใส่ตัวเลข 0 สูตรนี้จะใช้กับตารางข้อมูลที่ไม่เรียงลำดับ (จำไว้ว่า 0 แปลว่าไม่ หมายถึงไม่ต้องเรียงลำดับ)

7.3. ถ้าใส่ Option เป็นตัวเลข -1 นั้น สูตรนี้จะใช้กับตารางข้อมูล ที่เรียงค่าจากมากไปน้อย (จำไว้ว่าเรียงผิดธรรมชาติต้องเรียงจากมากไปน้อย)

ตัวอย่างลำดับการคำนวณตามเครื่องหมายบวกลบคูณหาร

ถ้าสร้างสูตร $=1+2-3*4/5^6$ ลงไปในเซลล์ Excel จะคำนวณตามหลักการดังนี้

1. คำนวณเครื่องหมายยกกำลัง $^$ ก่อน นั่นคือจะคำนวณ 5^6
2. ตามด้วยการคำนวณคูณหรือหาร แล้วแต่เครื่องหมายใดมาก่อนจากซ้ายไปขวา นั่นคือจะคำนวณ $3*4$ แล้วจึงนำผลคูณที่ได้ไปหารด้วยค่าตอบที่ได้จาก 5^6
3. ตามด้วยการคำนวณบวกหรือลบ แล้วแต่เครื่องหมายใดมาก่อนจากซ้ายไปขวา นั่นคือจะคำนวณ $1+2$ แล้วจึงนำผลบวกที่ได้ไปลบกับผลลัพธ์ที่ได้จาก $3*4/5^6$

จะสร้างสูตรให้สั้นลงไปได้แค่ไหน

สูตรสำเร็จรูปของ Excel 2003 ที่เราใช้กันจนชิน (จนไม่อยากจะเปลี่ยนมาใช้ Excel 2007 หรือ Excel รุ่นที่ใหม่กว่า) มีข้อกำหนดในตัวสูตรว่า ในวงเล็บของสูตรสำเร็จรูปหนึ่งๆนั้น สามารถซ้อนสูตรเข้าไปได้อีก 7 สูตร และในวงเล็บของบางสูตรจะถูกแบ่ง Argument หรือตัวแปรที่คั่นด้วยเครื่องหมาย comma แบ่งได้สูงสุด 30 Arguments (หรืออีกนัยหนึ่งใส่ Comma ได้สูงสุด 29 ตัวเพื่อแบ่งส่วนในวงเล็บออกเป็น 30 ส่วน) และสูตรที่ยาวที่สุดที่สามารถพิมพ์ลงไปในเซลล์หนึ่งๆได้นั้น ยาวได้สูงสุด 1,024 ตัวอักษร

ข้อกำหนดเหล่านี้เน้นว่าเพียงพอกับการคำนวณทั่วไป นานๆทีจึงมีโจทย์ที่ต้องใช้จนเกินกว่าขีดจำกัด ซึ่งแก้ไขได้ง่ายๆโดยแยกการคำนวณเป็นส่วนๆ แล้วแบ่งแต่ละเซลล์ให้แยกคำนวณแต่ละส่วนของสูตร จากนั้นจึงนำผลการคำนวณที่ได้มาคำนวณรวมกันต่อเป็นขั้นๆ หรือไม่ก็ต้องฉลาดเลือกใช้สูตรอื่นที่คำนวณได้คำตอบแบบเดียวกันแทน

สำหรับสูตรสำเร็จรูปใน Excel รุ่น 2007 เป็นต้นมา ปรับข้อกำหนดดังกล่าวเป็นดังนี้

1. ในวงเล็บของสูตรสำเร็จรูป สามารถซ้อนสูตรเข้าไปได้อีก 64 สูตร (Nested levels of functions)
2. ในวงเล็บของสูตรสำเร็จรูป สามารถแบ่งออกเป็น 255 arguments (Arguments in Functions)
3. ในเซลล์หนึ่งๆรับสูตรได้ยาวที่สุด 8,192 characters (Length of formula contents)

ข้อกำหนดเหล่านี้เปิดกว้างให้เราสร้างสูตรได้สบายขึ้น ซึ่งขอแนะนำว่าอย่าใช้ความสบายจนเกินไป เพราะตัวคุณนั้นแหละที่จะถูกลงโทษหากสร้างสูตรแบบสบายเกินไป เพราะหากต้องย้อนกลับมาแกะสูตรเพื่อแก้ไขอีกในภายหลัง มันไม่ใช่งานที่ง่ายเลยที่จะแกะสูตรซ้อนกันหลายสิบชั้น

พึงระลึกไว้เสมอว่า สูตรที่คุณสร้างตั้งแต่แรกนั้น ต้องเป็นสูตรที่ใช้งานได้ตลอดไปโดยไม่ต้องย้อนกลับมาแก้ไขสูตรอีก แต่ถ้าจำเป็นต้องแก้ไขสูตร ก็ต้องแก้ไขได้ง่าย ไม่ว่าคุณหรือเพื่อนของคุณก็ต้องแกะสูตรได้ง่าย

ดังนั้นการที่ Microsoft ปรับข้อจำกัดของสูตรให้ยืดหยุ่นมากขึ้นนี้ ถ้าสูตรใดที่คุณคิดว่ามีเงื่อนไขของการคำนวณไม่ยากนัก"ต่อการจำของคุณ"และง่ายที่จะย้อนกลับมาแก้ไข ก็เชิญซ้อนสูตรเข้าไปหลายๆชั้นได้ตามใจ แต่ถ้ามีทางเลือกอื่นที่ดีกว่า เช่น มีสูตรอื่น หรือต้องปรับโครงสร้างตารางเสียใหม่เพื่อให้เข้ากับสูตรนั้น ก็ขอแนะนำให้ตัดสินใจเลือกให้ดี

สูตรที่หาค่าได้ ก็ใช้บอกตำแหน่งได้

เชื่อไหมว่าเรื่องที่จะอธิบายนี้แทบทุกคนที่ใช้ Excel ทราบดีอยู่แล้ว ใช้กันอยู่ทุกวัน แต่นึกไม่ถึงว่าสูตรที่หาค่าได้ จะใช้บอกตำแหน่งที่แปลกแตกต่างจากที่เข้าใจกันอย่างไร

พอสร้างสูตรลงไปในเซลล์ใดๆก็ตามว่า =A1 สูตรนี้ก็จะดึงค่าจากเซลล์ A1 มาแสดงให้เห็น และพร้อมกันนั้นเราก็จะทราบด้วยว่า ค่าที่ได้นี้มาจากเซลล์ A1 เพราะดูจากตัวสูตรที่ใช้ตำแหน่งอ้างอิง =A1 นั่นเอง

ที่พิเศษไปกว่านี้ หากคุณตัดก๊อปปี้ในช่อง Allow edit directly in cells ทิ้งไป จะพบว่าเมื่อดับเบิลคลิกที่เซลล์สูตร =A1 จะย้ายตำแหน่ง cursor ไปที่เซลล์ต้นทาง A1 ให้ทันที

สิ่งที่อธิบายข้างต้นนี้เคยอธิบายมาหลายต่อหลายครั้งในการอบรมและในเว็บ ExcelExpertTraining.com ซึ่งพฤติกรรมแบบนี้แหละที่นำไปสู่เรื่องแปลกที่คาดไม่ถึง จากเดิมที่เขียนว่า "สูตรที่หาค่าได้ ก็ใช้บอกตำแหน่งได้" ขอเขียนใหม่ให้ละเอียดชัดเจนกว่าเดิมว่า

สูตรใดๆที่สามารถหาค่าที่เก็บไว้จากตารางได้โดยตรง สูตรนั้นย่อมเป็นสูตรที่บอกตำแหน่งได้ โดยอย่าสร้างสูตรนั้นลงไปในเซลล์ แต่ต้องนำสูตรนั้นไปซ่อนในสูตรอื่นก็จะกลายเป็นสูตรที่บอกตำแหน่ง

ยกตัวอย่างเช่น สูตร =INDEX(A1:B10,5,2) เป็นสูตรหาค่าจากตารางข้อมูลช่วง A1:B10 ในตำแหน่ง Row 5 ตัดกับ Column 2 จะได้ค่าจากเซลล์ B5

ถ้าอยากจะทำตามไปดูว่าเซลล์ B5 อยู่ที่ไหนโดยไม่ต้องนั่งแกะสูตรว่า Row 5 ตัดกับ Column 2 คือเซลล์ใด ให้พิมพ์เมาส์คลิกลากทับสูตร =INDEX(A1:B10,5,2) ที่แสดงบน Formula Bar จากนั้นกดปุ่ม F5 แล้วกดปุ่ม Enter จะพบว่าสูตร =INDEX(A1:B10,5,2) เปลี่ยนไปเป็น =B5 ให้ทันที พอเห็นดังนี้แล้วให้กดปุ่ม Esc เพื่อให้สูตรคืนสู่สภาพเดิม

ยิ่งกว่านั้น ถ้า Copy สูตร =INDEX(\$A\$1:\$B\$10,5,2) ไว้ก่อนแล้วกดปุ่ม F5 แล้วกดปุ่ม Ctrl+v เพื่อ Paste สูตรนี้ลงไปในช่วง Reference แล้วกดปุ่ม Enter จะพบว่า Excel พาคุณย้ายตำแหน่งไปที่เซลล์ B5 ให้เลย (สังเกตว่าวิธีนี้ต้องกำหนดตำแหน่งอ้างอิงเป็น Absolute คือ \$A\$1:\$B\$10 ก่อนด้วย)

ยิ่งกว่านี้ของยิ่งกว่านั้นอีก ถ้านำสูตร INDEX(A1:B10,5,2) ไปซ่อนในสูตรอื่น เช่น สูตร Offset จะทำให้สูตร INDEX(A1:B10,5,2) ทำหน้าที่ชี้ตำแหน่งต่อให้กับ Offset

ขอให้สร้างสูตร =OFFSET(INDEX(A1:B10,5,2),2,3) แล้วลองใช้วิธีข้างต้นค้นหาว่าหมายถึงเซลล์ใด จะพบว่าเป็นเซลล์ E7 ใหม เพราะเซลล์ E7 เป็นเซลล์ที่อยู่ถัดลงมาจากเซลล์ B5 จำนวน 2 row และถัดไปด้านขวาจำนวน 3 column เทียบเท่ากับการใช้สูตร =OFFSET(B5,2,3) นั่นเอง

แต่ถ้าสร้างสูตร =INDEX(A1:B10,5,2) ไว้ในเซลล์ D12 แล้วใช้สูตร Offset อ้างอิง D12 ในแบบ =OFFSET(D12,2,3) จะพบว่า D12 ไม่ได้ช่วยบอกตำแหน่งต่อให้กับสูตร Offset เพราะสูตร Index ในเซลล์ D12 ทำหน้าที่หาค่าจากเซลล์ B5 มาแสดงเสร็จก็จบหน้าที่ไปแล้ว

สูตรที่สามารถหาค่าที่เก็บไว้จากตารางได้โดยตรง ซึ่งทำงานได้แบบที่อธิบายนี้ ได้แก่ If, Choose, Index, Offset, Indirect (แต่ไม่รวมสูตร VLookup เพราะมันไม่ได้หาค่าได้โดยตรง) ซึ่งไม่จำเป็นต้องคืนค่าเพียงค่าเดียวเป็นตำแหน่งเซลล์เดียว โดยสามารถคืนค่าเป็นตารางเพื่อใช้บอกตำแหน่งตารางทั้งตารางก็ยังได้

จำนวนเซลล์สูตรเดียว

ผู้สร้างสูตร Excel ทั่วไป ค้นเคยกับการสร้างสูตรหนึ่งที่คืนค่าเพียงหนึ่งค่า หรืออีกนัยหนึ่งเรียกว่า สูตรเดียวคืนค่าเดียว ซึ่งสร้างลงไปในเซลล์เดียว แต่น้อยคนนักที่จะทราบว่า สูตร Excel สูตรหนึ่งๆสามารถคืนค่าเป็นคำตอบได้หลายค่าก็เป็นไปได้เหมือนกัน สูตรพวกหลังนี้เรียกว่า สูตร Array

การที่เราเลือกเซลล์เดียว แล้วพิมพ์สูตรลงไปในเซลล์เดียวนั้น มีสาเหตุเนื่องจากสูตรนั้นคืนค่าได้คำตอบเพียงค่าเดียว

ถ้าสูตรหนึ่งๆคืนค่าเป็นคำตอบได้หลายค่า หากสร้างสูตรนั้นลงไปในเซลล์เดียว ย่อมไม่ถูกต้อง แม้จะเห็นคำตอบแสดงขึ้นมาก็ตาม เพราะคำตอบที่แสดงขึ้นมาให้เห็นนั้นเป็นเพียงคำตอบแรกคำตอบเดียว เราหมดโอกาสทราบคำตอบค่าอื่นๆ (ถ้าใช้ Excel 365 รุ่นล่าสุด Excel จะกระจายคำตอบออกไปให้เองซึ่งเป็นวิธีการใหม่เรียกว่า Dynamic Array)

พออ่านถึงตรงนี้ คนที่ไม่รู้จักกับสูตร Array คงเริ่มสงสัยว่า สูตรแบบไหนกันที่มีคำตอบได้หลายค่าในสูตรเดียว มาทดลองโดยเริ่มจากพิมพ์ค่าใดๆก็ได้ลงไปเซลล์ A1:A3 สมมติว่าพิมพ์ 11, 22, 33 ตามลำดับ

	D4		fx {=A1:A3}	
	A	B	C	D
1	11	11		
2	22			
3	33			
4				11
5				22
6				33
7				

แล้วสร้างสูตร =A1:A3 ลงไปในเซลล์ B1 จะเห็นคำตอบเป็นเลข 11 ค่าเดียวใช่ไหม (ถ้าใช้ Excel 365 รุ่นล่าสุด จะพบว่าสูตรกระจายตัวลงไปเซลล์ B1:B3 ให้เอง หากต้องการให้แสดงแบบ Excel รุ่นก่อนให้ใช้ =@A1:A3 จะบังคับให้ Excel 365 ทำงานแบบ Excel รุ่นก่อน)

จากนั้นให้แกะดูค่าที่แท้จริงในสูตร โดยกดปุ่ม F2 ตามด้วยปุ่ม F9 จะพบว่าสูตร =A1:A3 เปลี่ยนเป็น ={11;22;33}

ขอให้สังเกตว่าเครื่องหมายวงเล็บ { ที่เกิดขึ้นนั้นอยู่หลังจากเครื่องหมายเท่ากับ ซึ่งแสดงค่าแบบ Array และระหว่างตัวเลขจะมีเครื่องหมาย semi-colon ; คั่น ซึ่งแสดงว่าค่าถัดไปอยู่ใน Row ถัดไป (หรือขึ้น Row ใหม่)

พอเห็นค่าแล้วว่าสูตรนี้มี 3 ค่าและแต่ละค่าอยู่ตามแนวตั้ง ก็ให้กดปุ่ม Esc เพื่อย้อนกลับไปที่สูตร =A1:A3 ตามเดิม

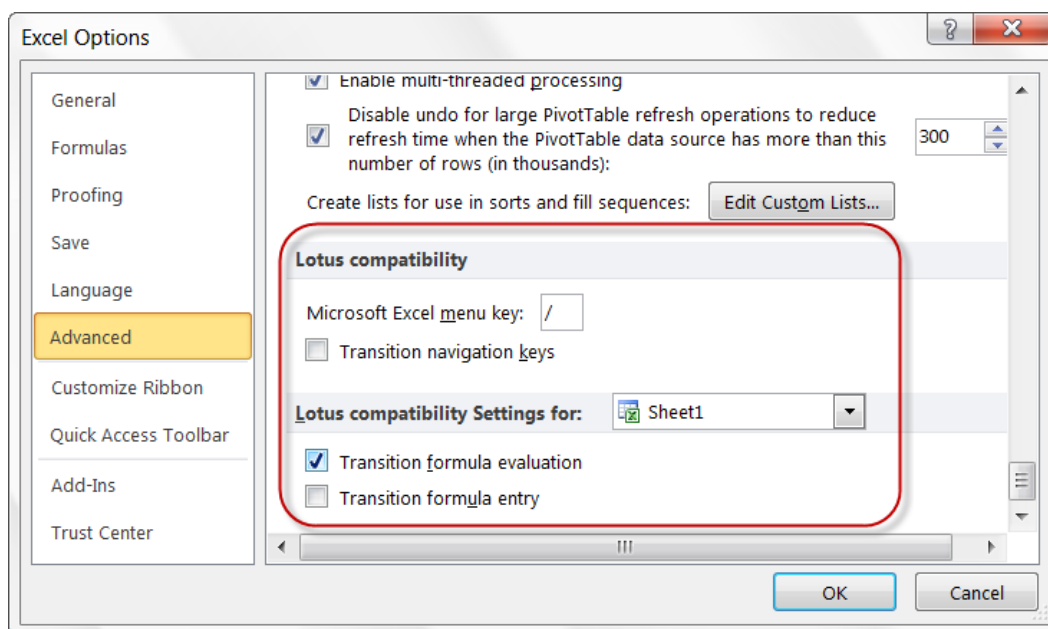
คราวนี้ถึงขั้นตอนสำคัญ หากต้องการกระจายค่าแต่ละค่าในสูตร Array ลงไปในเซลล์ ให้เริ่มจากเลือกเซลล์ 3 เซลล์ตามแนวตั้ง (เพราะเราทราบจากการแกะสูตรมาก่อนแล้วว่าค่า 3 ค่าตามแนวตั้ง) โดยจะเลือกที่ตำแหน่งใดก็ได้ในชีท จากนั้นสร้างสูตร =A1:A3 ลงไปแล้วกดปุ่ม Ctrl+Shift+Enter พร้อมกัน จะพบว่า Excel กระจายค่าในสูตรลงไปในเซลล์แต่ละเซลล์ให้เห็นจนครบ และตัวสูตรเปลี่ยนเป็น {=A1:A3} โดยสังเกตว่าคราวนี้เครื่องหมาย { ตัวแรกวางไว้ก่อนเครื่องหมายเท่ากับ แสดงว่าเป็นวงเล็บปีกกาที่เกิดจากการกดปุ่ม Ctrl+Shift+Enter (ไม่ใช่วงเล็บปีกกาที่ใช้กำกับลำดับของค่าที่ต้องอยู่หลังเครื่องหมายเท่ากับ)

ประโยชน์ของการเลือกจำนวนเซลล์ให้ครบเหมาะกับจำนวนและแนวของค่าที่ต้องการนี้ จะช่วยให้เราเห็นค่าทุกค่ากระจายตัวลงไปเซลล์แต่ละเซลล์ ทำให้ประหยัดเวลาไม่ต้องคอยกดปุ่ม F2 ตามด้วย F9 เพื่อแกะสูตรเป็นครั้งๆไป อีกทั้งสูตรที่สร้างขึ้นพร้อมกันทีเดียวหลายเซลล์ จะทำให้ Excel เสียเวลาในการคำนวณเพียงครั้งเดียว

วิธีแก้ error #VALUE! ด้วย Lotus 1-2-3

เมื่อนำเซลล์ที่มีค่าเป็นตัวเลขมาบวกด้วยเซลล์ที่มีค่าเป็นตัวอักษรโดยการสร้างสูตรบวกที่นำเซลล์มาบวกกัน เช่น =A1+A2 จะพบว่าได้คำตอบเป็น #VALUE! นั้น เราสามารถสั่งให้ Excel คำนวณหาคำตอบเป็นยอดรวมเฉพาะตัวเลขได้โดยใช้สั่งให้ Excel คำนวณแบบ Lotus 1-2-3

ใน Excel 2010 เริ่มจากสั่ง File > Options > Advanced > แล้วเลื่อนจอไปด้านล่างสุดจะพบ Lotus compatibility Settings for: ชื่อชีทที่ต้องการให้คำนวณแบบ Lotus 1-2-3 โดยให้กาช่อง Transition formula evaluation (Excel 2007 ให้คลิกปุ่ม Office แทน File)



คำสั่ง Transition formula evaluation นี้เกิดขึ้นในสมัยที่คนทั้งโลกยังนิยมใช้ Lotus 1-2-3 กัน ดังนั้นเพื่อช่วยทำให้คนเปลี่ยนใจมาใช้ Excel ได้ง่ายขึ้น ทางไมโครซอฟท์จึงสร้างคำสั่งนี้ขึ้น เพื่อให้พวกเราสามารถนำแฟ้มที่สร้างจาก Lotus 1-2-3 มาเปิดใช้ใน Excel ต่อได้ทันทีโดย Excel จะเปลี่ยนระบบการคำนวณในชีทที่เลือกไว้ให้คำนวณตามแบบที่ Lotus 1-2-3 ใช้นั้นคือ ใน Lotus 1-2-3 ถือว่าตัวอักษรมีค่าเท่ากับ 0 (ส่วน Excel ถือว่าตัวอักษรมีค่ามากกว่าเลขทั้งปวง)

นอกจากนั้นหากกาช่อง Transition formula entry จะช่วยทำให้สูตรที่สร้างจาก Lotus 1-2-3 release 2.2 ถูกเปลี่ยนเป็นสูตรของ Excel ให้ทันทีเมื่อเปิดแฟ้มที่ใช้สูตรนั้นด้วย Excel

ส่วนช่อง Transition navigation keys จะเปลี่ยนพฤติกรรมของปุ่มลูกศร ปุ่ม Tab หรือปุ่มใดๆที่ทำหน้าที่เคลื่อนย้ายตำแหน่งเซลล์ที่เลือก ให้ทำตามที่ Lotus 1-2-3 ใช้ ซึ่งช่วงแรกกว่าผมจะหันมาใช้ Excel อย่างเต็มตัว ต้องอาศัยกาช่องนี้อยู่นานเพราะมือยังไม่คุ้นกับปุ่มที่เคยเลื่อนไปเลื่อนมาตามแบบที่ตัวเองเคยชิน

ตัวอักษรไม่ได้มีค่าเท่ากับ 0

ก่อนจะจบบทความนี้ ทราบไหมว่า ตัวอักษร a หรือตัวอักษรใดๆก็ตามที่บันทึกลงไปในเซลล์ มิได้ถือว่ามีค่าเท่ากับ 0 หรอกนะ ถ้าไม่เชื่อ สมมติว่าเซลล์ A1 มีค่าเป็น a ขอให้ลองสร้างสูตร $=A1=0$ ดูก็จะได้พบว่าได้คำตอบเป็น FALSE ส่วนสูตร $=A1>999999999$ จะได้คำตอบเป็น TRUE

Excel ถือว่าตัวอักษรใดๆมีค่ามากกว่าเลขทั้งปวง อีกนัยหนึ่งมีค่าเป็น Infinity หากสร้างสูตร $=\text{Sum}(A1,123)$ แล้วได้คำตอบเท่ากับ 123 เป็นเพราะ Excel จะละเลยไม่นำตัวอักษรมาคำนวณในสูตร Sum นี่แหละเป็นเรื่องที่หลายคนชอบเข้าใจผิดกันว่า ตัวอักษรมีค่าเป็น 0

ถ้าสร้างสูตร $=A1+123$ จะได้คำตอบเป็น #VALUE! เพราะ Excel จะไม่ยอมรับการนำตัวอักษรมาบวกกับตัวเลข แต่เราสามารถเปลี่ยนพฤติกรรมของ Excel ให้คิดแบบ Lotus 1-2-3 ได้ตามคำอธิบายข้างต้น

ถ้าถามว่าระหว่างสูตรบวกกับสูตร Sum เราจะเลือกใช้การบวกเลขวิธีใดกับงานสำคัญ

คำตอบก็คือ ต้องเลือกใช้สูตรบวก เพราะหากมีการบันทึกตัวเลขผิด กลายเป็นบันทึกตัวอักษรลงไปแทน จะพบว่า Excel ไม่ยอมคำนวณให้ โดยได้คำตอบเป็น #VALUE! ซึ่งเท่ากับช่วยเตือนเราให้ทราบว่าเกิดข้อผิดพลาดขึ้นมาในระหว่างการบันทึกข้อมูลนั่นเอง

การแบ่งขอบเขตพื้นที่ตาราง

ในซีทของ Excel 2007 เป็นต้นมามีพื้นที่ตารางใหญ่กว่ารุ่นก่อนอย่างมาก จากเดิมมีเพียง 65,536 row และ 256 column ได้ปรับเพิ่มเป็น 1,048,576 row และ 16,384 column ซึ่งถ้าคุณพิมพ์ค่าลงไปแล้วพยายาม paste ให้ครบทุกเซลล์ในซีท จะพบว่า Excel ต้องใช้เวลาอันแสนนานทีเดียวกว่าจะได้ข้อมูลเต็มทั้งซีท และเมื่อสั่ง save จะไม่มี hard disk เครื่องไหนที่สามารถเก็บแฟ้มข้อมูลนั้นได้ เพราะแฟ้มจะมีขนาดใหญ่เกินกว่าจะรับได้ (ขออย่าพยายามทดลองทำตามนี้เด็ดขาด เพราะคุณจะรอไม่ไหวแล้วต้องหันไปปิดเครื่องคอมพิวเตอร์ไปก่อนที่จะ save เสีย)

สาเหตุที่ Excel ขยายขนาดตารางให้ใหญ่ขึ้นจนดูเหมือนใหญ่กว่าความเป็นจริง มิได้มุ่งให้คุณจำเป็นต้องใช้ให้ครบทุกเซลล์ เพียงแต่จะช่วยให้สามารถรับข้อมูลจากโปรแกรมอื่นเข้ามาเก็บไว้ในซีทได้สะดวกขึ้น และเมื่อคุณหันมาใช้ Excel 2007 แล้วก็ได้หมายความว่าจากนี้ไปคุณไม่จำเป็นต้องใช้โปรแกรมฐานข้อมูลอื่นอีกต่อไป เพราะในที่สุดแล้วเมื่อข้อมูลมีปริมาณมากขึ้นและมีความสัมพันธ์ในฐานข้อมูลซับซ้อนตามขึ้นไปแล้ว โปรแกรมที่สร้างขึ้นมาเพื่อจัดการฐานข้อมูลโดยเฉพาะย่อมเหมาะกว่าอยู่ดี

อย่าว่าแต่ขนาดของตารางที่มีขนาดใหญ่สามารถรับข้อมูลได้มากขึ้นนี้เลย ต่อให้ใช้ Excel 2003 ที่มีขนาดตารางเล็กกว่า หากผู้ใช้ Excel ไม่เป็น พอนำข้อมูลมาเก็บไว้ในซีทอย่างไม่มีหลักการ แล้วเมื่อถึงเวลาจะต้องแยกข้อมูลออกจากกันเป็นส่วนๆก็จะทำได้ยาก จึงขอเสนอให้ใช้หลัก **ICO** แบ่งขอบเขตพื้นที่ตาราง กล่าวคือ

1. ให้แบ่งแยกตารางข้อมูลออกเป็น 3 ประเภท คือ ตาราง Input, ตาราง Calculate, และ ตาราง Output (หรือใช้ตัวย่อว่า ICO)
2. **ตาราง Input** คือ ตารางที่ใช้เก็บข้อมูลที่พิมพ์ลงไปหรือเป็นข้อมูลที่รับมาจากโปรแกรมอื่น หรืออีกนัยหนึ่งเป็นตารางฐานข้อมูลนั่นเอง ตาราง Input นี้ต้องออกแบบให้โครงสร้างของตารางมีลักษณะดังนี้
 - 2.1. หัวตารางด้านบนและแต่ละรายการ ต้องเป็น row เดียว ซึ่งอาจแต่งข้อความในเซลล์ให้แสดงเป็นหลายบรรทัดได้ โดยกดปุ่ม **Alt+Enter** เพื่อขึ้นข้อความในบรรทัดใหม่ในเซลล์เดิม

- 2.2. ต้องบันทึกข้อมูลใน column ใด column หนึ่งติดต่อกันไปตลอด ห้ามเว้นช่องว่าง ซึ่งโดยทั่วไป column ซ้ายสุดมักจะมีข้อมูลติดต่อกันไปอยู่แล้วเนื่องจากเป็นข้อมูลของเลขที่รายการ หรือข้อมูลรหัส หรือข้อมูลวันที่ซึ่งเรียงไปตามลำดับรายการที่เกิดขึ้น
- 2.3. ต้องเว้นขอบตารางไม่ว่าจะเป็นขอบด้านบนล่างซ้ายขวาอย่างน้อยข้างละ 1 เซลล์ ห้ามนำตาราง Input ไปติดกับเซลล์ข้อมูลอื่นซึ่งไม่ใช่ข้อมูลเรื่องเดียวกับตารางข้อมูลนั้น
3. ตาราง **Calculate** คือ ตารางที่ใช้สร้างสูตรคำนวณ ไม่จำเป็นต้องมีโครงสร้างตามแบบของตาราง Input แต่ในตาราง Calculate นี้คุณต้องหาทางสร้างสูตรที่เซลล์หัวมุมเป็นสูตรฉลาดที่สามารถปรับตำแหน่งอ้างอิงได้เอง และในขณะที่สร้างสูตรต้องใช้ปุ่ม F4 เพื่อใส่เครื่องหมาย \$ เพื่อควบคุมตำแหน่งอ้างอิงในสูตรไปยังเซลล์ที่บันทึกค่าตัวแปรที่ใช้คำนวณไว้ ซึ่งโดยทั่วไปเซลล์ค่าตัวแปรจะวางไว้ตามแนวคู่ขนานกับแนวของตาราง Calculate เพื่อช่วยให้เครื่องหมาย \$ ที่ใส่ไว้ในสูตร ทำหน้าที่ควบคุมตำแหน่งของแนวของเซลล์ตัวแปรไว้เพื่อนำค่ามาใช้คำนวณนั่นเอง
4. ตาราง **Output** คือ ตารางแสดงผลลัพธ์ที่ต้องการ โดยตารางนี้จะออกแบบให้มีหน้าตาอย่างไรก็ได้ตามแต่จะนำไปแสดงผลบนจอภาพหรือในหน้ากระดาษที่สั่งพิมพ์

สาเหตุที่ต้องใช้หลัก ICO แยกข้อมูลแต่ละประเภทออกจากกัน เพื่อช่วยทำให้เราสามารถใช้ Excel รองรับกับข้อมูลที่มีปริมาณมากขึ้นไปเรื่อยๆ ได้ตลอด ซึ่งในช่วงแรกที่ยังมีข้อมูลไม่มากนัก อาจใช้ชีทเดียวเก็บข้อมูลทั้ง I C และ O ไว้ด้วยกัน จากนั้นเมื่อมีปริมาณข้อมูลมากขึ้นหรือข้อมูลมีความสลับซับซ้อนมากขึ้นจึงค่อยย้ายตารางแยกไปเก็บในชีทอื่นหรือแฟ้มอื่นได้ทันทีโดยไม่กระทบกับสูตรคำนวณที่สร้างไว้

ข้อผิดพลาดที่พบเห็นบ่อยครั้งเกิดจากการออกแบบตารางตามแบบรายงานที่ต้องพิมพ์ เช่น หากมีรายงานยาว 100 หน้า ก็มักออกแบบตารางใน Excel ให้ยาวเหยียดต่อกันลงมาทั้ง 100 หน้าแล้วจึงสั่งพิมพ์ออกไปทีละหน้า ซึ่งวิธีนี้ทำให้แฟ้มมีขนาดใหญ่และยากต่อการแก้ไขเพราะจำเป็นต้องย้อนกลับไปไล่แก้ไขในตารางแต่ละหน้า

แต่หากออกแบบตารางแบบ ICO เป็น ให้เริ่มจากแยกตารางข้อมูลที่เป็น Input ออกไป แล้วใช้สูตรดึงค่าจากตาราง Input มาคำนวณในตาราง Calculate จากนั้นจึงดึงผลลัพธ์ที่คำนวณเสร็จแล้วไปสร้างตาราง Output ซึ่งจุดสำคัญอยู่ที่ตาราง Output นี้แหละ โดยเราต้องหาทางสร้างตาราง Output ไว้เพียงหน้าเดียว จากนั้นต้องสร้างสูตรดึงข้อมูลแต่ละหน้ามาแสดงในตาราง Output หน้าเดียวนี้ให้ได้ ช่วยให้แฟ้มมีขนาดเล็กและแก้ไขได้ง่ายกว่ามาก

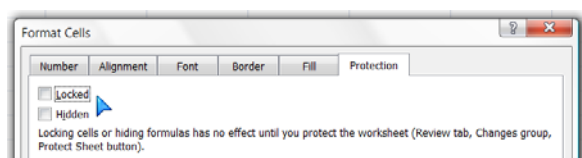
สมมติว่าหน้ากระดาษที่พิมพ์มี 30 บรรทัดต่อหน้า ดังนั้นพอพิมพ์เลขหน้าที่ 1 ลงไป ต้องสร้างสูตรดึงผลลัพธ์บรรทัดที่ 1 - 30 มาแสดงในหน้านี้ จากนั้นพอเปลี่ยนเลขหน้าเป็นเลข 2 ก็จะได้บรรทัดที่ 31 - 60 มาแสดง พอเป็นหน้า 3 ก็ได้บรรทัดที่ 61 - 90 มาแสดง ซึ่งสูตรที่ดึงข้อมูลเช่นนี้ได้ก็คือสูตร VLookup, Match, Index เป็นต้น

การจัดเตรียมพื้นที่ ICO เพื่อรับข้อมูลและป้องกันการแก้ไข

แฟ้มที่สร้างเสร็จพร้อมใช้งาน ต้องมีระบบแยกแยะตำแหน่งตารางที่เปิดให้รับค่าใหม่ได้พร้อมทั้งหาทางป้องกันตารางที่ไม่ต้องการให้ใครเข้าไปแก้ไข โดยต้องพิจารณาจากพื้นที่ตารางแต่ละส่วน กล่าวคือ

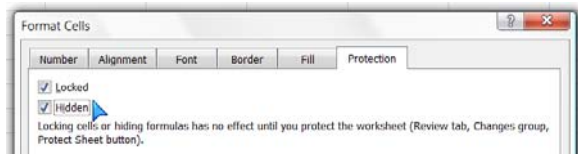
1. ตาราง **Input** เป็นพื้นที่ส่วนที่เปิดให้ผู้ใช้งานสามารถพิมพ์ค่าใหม่ลงไปได้ โดยมีขั้นตอนในการจัดเตรียมพื้นที่ดังนี้

- 1.1. คลิกเลือกพื้นที่ตารางส่วนที่เป็น Input (หากมีหลายพื้นที่ไม่ติดต่อกัน ให้กดปุ่ม **Ctrl** ค้างไว้ก่อนแล้วคลิกเลือกพื้นที่แต่ละส่วน แล้วอาจกำหนดสี Font ให้แตกต่างจากส่วนอื่น ซึ่งผมเองชอบใช้สีชมพูเพื่อบอกผู้ใช้งานว่าเป็นเซลล์ Input)
- 1.2. คลิกขวา สั่ง Format Cells > Protection > แล้วตัดกาช่อง Locked ทิ้งไป



- 1.3. สั่ง Review > Protect Sheet
- 1.4. หลังจากที่ได้ Protect Sheet เสร็จ เมื่อพิมพ์ข้อมูลใหม่ลงไปในพื้นที่ที่เป็น Input แล้วให้กดปุ่ม **Tab** ซึ่งจะพบว่า Excel จะกระโดดไปเลือกเซลล์ Input ต่อไปให้เอง
2. ตาราง **Calculate** และตาราง **Output** เป็นพื้นที่ซึ่งมีเซลล์ส่วนใหญ่เป็นสูตรซึ่งเมื่อสร้างเสร็จแล้ว ควรป้องกันไม่ให้ใครเข้าไปดูสูตร แก้ไขสูตร หรือลบสูตรทิ้งได้อีก โดยมีขั้นตอนในการจัดเตรียมพื้นที่ดังนี้ (ถ้าขี้หนุก Protect อยู่ต้อง Unprotect Sheet ก่อน)
 - 2.1. คลิกเลือกพื้นที่ตารางส่วนที่เป็นตาราง Calculate และตาราง Output
 - 2.2. คลิกขวา สั่ง **Format Cells > Protection > แล้วกาช่อง Hidden** เพื่อซ่อนการแสดงผลสูตรที่จะเห็นบน Formula Bar และยังป้องกันไม่ให้ Copy สูตรออกไปได้ด้วย โดยจะ Copy เฉพาะค่าผลลัพธ์ออกไปได้เท่านั้น (ห้ามตัดกาช่อง **Locked** ทิ้งอย่าง

เด็ดขาด) และต้องทำขั้นตอนต่อไปก่อนด้วยจึงจะเห็นผล



2.3. สิ่ง Review > Protect Sheet

การกำหนดตำแหน่งอ้างอิง ข้ามชีทหรือข้ามแฟ้ม

ผมเริ่มต้นใช้โปรแกรม Excel ตั้งแต่ยุคที่ในแฟ้มหนึ่งๆยังมีชีทให้ใช้งานได้เพียงชีทเดียวและยัง
ไม่สามารถใช้สูตรเชื่อมข้อมูลจากแฟ้มอื่นมาใช้งานร่วมกัน ในยุคนั้นต้องออกแบบตารางทั้งหมด
ลงไปชีทเดียว ต่อมา Excel ถูกพัฒนาให้สามารถส่งข้อมูลจากแฟ้มหนึ่งออกไปยังแฟ้มอื่นได้
แต่ก็ยังไม่สามารถทำหน้าที่ทั้งรับทั้งส่งข้อมูลกลับไปกลับมาระหว่างแฟ้มเช่นที่ Excel ในรุ่น
ปัจจุบันทำได้ ซึ่งถ้าดูให้ดีจะพบว่าสิ่งที่ Excel สามารถส่งข้อมูลกลับไปกลับมาได้นั้นเป็นเพราะ
เราถูก Excel หลอกให้ดูเหมือนว่ามีหลายชีทหรือมีหลายแฟ้ม...แค่นั้นเอง

ขอให้จำไว้ว่า ไม่ว่าจะเป็ต่างชีทหรือต่างแฟ้ม พอเปิดแฟ้มขึ้นมาใช้งานพร้อมกัน มัน
ก็คือชีทเดียวกันนั่นแหละ

การที่เราเห็นว่าในแฟ้มหนึ่งๆมีหลายชีทนั้น เป็นภาพลวงตาที่ช่วยให้ผู้ใช้งานสามารถแยกเก็บ
ข้อมูลต่างเรื่องไว้ในชีทต่างกันไป ส่วนข้อมูลที่ link ข้ามแฟ้มได้นั้น ถ้าคุณใช้ Excel รุ่นแรกๆมา
ก่อนอย่างผม พอสั่ง Unhide Sheet จะพบว่าในแฟ้มปลายทางมีชีทหนึ่งถูกซ่อนไว้ โดยชีทนี้ใช้
เก็บข้อมูลจากชีทต้นทางไว้ทั้งหมด ดังนั้นข้อมูลที่ได้มาจากสูตร link ข้ามแฟ้ม ถ้าว่ากันให้
ถูกต้อง ไม่ได้เป็นผลจากสูตรที่ link มาจากแฟ้มต้นทางโดยตรง แต่เป็นข้อมูลที่ได้มาจากชีทที่
ซ่อนไว้ภายในชีทปลายทางในแฟ้มเดียวกันกับแฟ้มที่มีสูตร link นั้นเอง โดยเรียกข้อมูลที่เก็บ
ในชีทนี้ว่า Cache Data ซึ่งส่งผลให้แฟ้มปลายทางมีขนาดแฟ้มใหญ่กว่าปกติตามไปด้วย
(ต่อมา Microsoft ได้ปรับปรุงให้ Excel ซ่อนชีทนี้ไว้โดยไม่สามารถค้นหาพบเช่นแต่ก่อน)

ถ้าคุณสามารถสร้างสูตร link ข้อมูลภายในชีทเดียวกัน ก็ขอให้ใช้วิธีเดียวกันในการ link ข้อมูล
ข้ามชีทหรือแม้แต่ข้ามแฟ้ม ขอเพียงอย่ากลัวภาพลวงตาที่ทำให้เห็นว่าเป็นต่างชีทหรือต่างแฟ้ม
กันเพราะความจริงมันก็คือชีทเดียวกันนั่นเอง ซึ่งก่อนที่จะใช้วิธีสร้างสูตร link ที่นิยมใช้กัน
ต่อไปนี้ ขอให้เปิดแฟ้มทุกแฟ้มที่ต้องการสร้างสูตร link ข้อมูลระหว่างกันขึ้นมาก่อนทั้งแฟ้มต้น
ทางและแฟ้มปลายทางแล้วจะช่วยให้สร้างสูตร link ได้ง่ายมาก

1. วิธีย้อนปลายกลับมาหาต้น

1.1. ให้เริ่มจากพิมพ์เครื่องหมายเท่ากับ = ที่เซลล์ปลายทาง

1.2.(หากไม่ต้องการ link ข้ามชีทหรือข้ามแฟ้มให้ข้ามขั้นตอนนี้) คลิกเลือกชื่อชีทต้นทาง หรือคลิกชื่อแฟ้มต้นทางที่เปิดไว้ก่อนแล้วจากคำสั่ง **View > Switch Windows**

1.3. แล้วคลิกเลือกเซลล์ต้นทาง

1.4. จากนั้นกดปุ่ม Enter เพื่อรับสูตร link ตำแหน่งเซลล์ต้นทางกลับไปบันทึกไว้ที่เซลล์ปลายทาง (วิธีนี้โดยทั่วไปจะทำได้ทีละเซลล์ เว้นแต่จะสร้างสูตรแบบ Array จึงจะสร้างพร้อมกันหลายเซลล์ทั้งตาราง และขอให้สังเกตว่าถ้าเป็นสูตร link ข้ามแฟ้มจะได้ตำแหน่งอ้างอิงแบบ Absolute เสมอ)

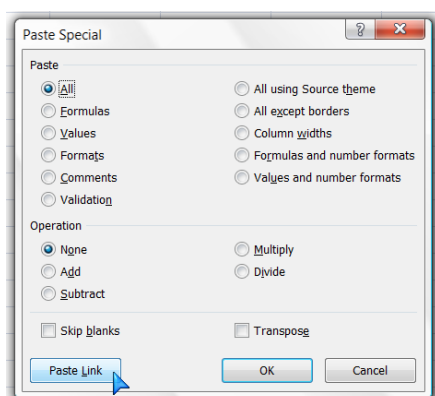
2. วิธี Copy จากต้น แล้วมา Paste Special แบบ Paste Link ที่ปลายทาง

2.1. ให้เริ่มจากเลือกพื้นที่ตารางต้นทางทั้งตารางพร้อมกัน

2.2. สั่ง Copy

2.3.(หากไม่ต้องการ link ข้ามชีทหรือข้ามแฟ้มให้ข้ามขั้นตอนนี้) คลิกเลือกชื่อชีทปลายทาง หรือคลิกชื่อแฟ้มปลายทางที่เปิดไว้ก่อนแล้วจากคำสั่ง **View > Switch Windows**

2.4. จากนั้นให้เลือกเซลล์หัวมุมตารางปลายทางเพียงเซลล์เดียวแล้วคลิกขวา สั่ง **Paste Special > กดปุ่ม Paste Link** จะได้สูตร link พร้อมกันทีเดียวทั้งตาราง



3. วิธี Cut จากต้น แล้วไป Paste ที่ปลายทาง โดยวิธีนี้ว่าไปแล้วไม่ได้เป็นการสร้างสูตร link แต่ให้เริ่มจากสร้างสูตร link ไว้ในชีทเดียวกันตามวิธีข้างต้นให้เสร็จก่อน จากนั้นจึงสั่ง Cut ตารางสูตรจากชีทต้นทางไป Paste ที่ตำแหน่งปลายทางในชีทอื่นหรือแฟ้มอื่น ซึ่ง Excel จะปรับโครงสร้างตำแหน่งอ้างอิงในสูตรให้กลายเป็นตำแหน่งอ้างอิงข้ามชีทหรือข้ามแฟ้มให้เองทันที

วิธีที่ 3 นี้เป็นเคล็ดลับสำคัญในการสร้างสูตร link ข้ามแฟ้ม โดยเริ่มจากสร้างสูตร link ทั้งหมดในชีทเดียวกันให้เสร็จก่อน จากนั้นเมื่อ Cut เซลล์สูตรไปที่ชีทอื่นจะได้สูตร link

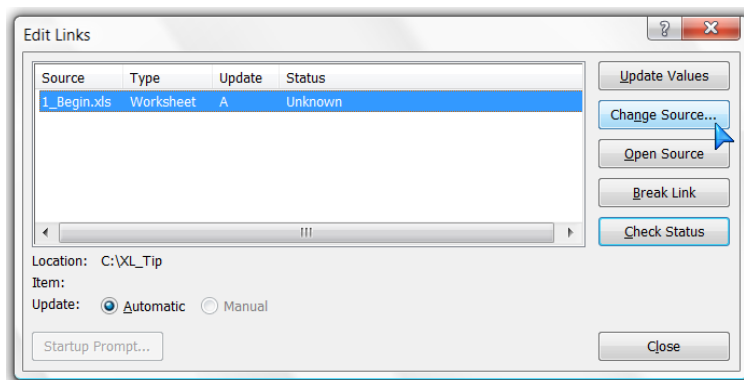
ข้ามชีท หรือถ้า Cut เซลล์สูตรไปที่แฟ้มอื่นจะได้สูตร link ข้ามแฟ้ม ช่วยให้ไม่ต้องเสียแรงเสียเวลาสร้างสูตร link ข้ามชีทหรือข้ามแฟ้มตั้งแต่แรกแต่แน่นอน เช่น

- จากเดิมสร้างสูตร link ในชีทเดียวกันเป็น =A1
- พอ Cut ไปที่ชีทอื่น สูตรนี้จะปรับตัวใหม่เป็น =Sheet1!A1
- พอ Cut ต่อไปที่แฟ้มอื่น สูตรนี้จะปรับตัวใหม่เป็น =[Source.xlsx]Sheet1!A1 โดยคำว่า Source ก็คือชื่อแฟ้มต้นทาง และคำว่า Sheet1 คือชื่อชีทต้นทาง
- ต่อมาถ้าปิดแฟ้มต้นทางที่ชื่อ Source ทั้ง สูตรนี้จะปรับตัวใหม่เป็น = 'D:\Dummy\[Source.xlsx]Sheet1!A1 ซึ่งการที่มีชื่อ Drive D และชื่อโฟลเดอร์ว่า Dummy แสดงไว้ด้านหน้าสูตร แสดงว่าแฟ้มต้นทางที่ชื่อ Source นั้น ในขณะนั้นถูกจัดเก็บไว้ที่ใดแต่ยังไม่ได้ถูกเปิดขึ้นมา

ข้อควรระวังในการรักษาสูตร link ข้ามแฟ้มให้ใช้งานได้ถูกต้องตลอดไป

1. หากต้องการทำการแก้ไขใดๆในแฟ้มที่ link กัน ต้องเปิดแฟ้มที่ link กันทุกแฟ้มขึ้นมา ทั้งแฟ้มต้นทางและแฟ้มปลายทางก่อนที่จะเริ่มทำการแก้ไขเสมอ เพื่อให้การแก้ไขที่เกิดขึ้น เช่น การย้ายตำแหน่งเซลล์ การเปลี่ยนชื่อชีทหรือแม้แต่ชื่อแฟ้ม ส่งผลให้ Excel ไล่แก้ตำแหน่งอ้างอิงในสูตร link ข้ามแฟ้มให้เอง (ถ้าต้องการให้ Excel ช่วยจำว่าต้องเปิดแฟ้มใดบ้างพร้อมกัน ให้ใช้การเปิดแฟ้ม Workspace ช่วย ซึ่งจะอธิบายในภายหลัง)
2. หากต้องการ **Save as** เพื่อเปลี่ยนชื่อแฟ้มหรือย้ายโฟลเดอร์ที่เก็บ ให้ทำการ Save แฟ้มต้นทางก่อนแล้วจึง Save แฟ้มปลายทางตามที่หลัง โดยการ Save ที่ว่านี้ให้ใช้โปรแกรม Excel เปิดแฟ้มแล้ว Save จากโปรแกรม Excel เท่านั้น เพื่อให้ Excel ปรับตำแหน่งอ้างอิงในสูตรตามตำแหน่งของโฟลเดอร์และชื่อแฟ้มที่เปลี่ยนไป (ต่างจากการใช้ Windows Explorer ในการย้ายโฟลเดอร์หรือเปลี่ยนชื่อแฟ้ม ซึ่งจะไม่ช่วยให้สูตรมีการแก้ไขตำแหน่งอ้างอิงตามแน่นอน)
3. หากต้องการแก้ไขตำแหน่งอ้างอิงเนื่องจาก Excel เตือนว่าค้นหา link ไม่พบ อาจเป็นเพราะคุณพลั้งเผลอไม่ได้ปฏิบัติตามข้อควรระวัง 2 ข้อข้างต้น ก็ยังสามารถแก้ไขให้ link ทำงานตามเดิมได้ตามขั้นตอนต่อไปนี้ โดย Excel จะไล่แก้ไขชื่อแฟ้มต้นทางในเซลล์สูตร link ทุกเซลล์ให้เองไม่ว่าอยู่ในชีทใด

- 3.1. ให้เปิดแฟ้มปลายทางขึ้นมาก่อนโดยไม่ต้องเปิดแฟ้มต้นทาง (แฟ้มปลายทางนี้ก็คือแฟ้มที่เปิดขึ้นมาแล้วพบว่า Excel มีค่าเตือนว่าหา link ไม่พบ)
- 3.2. ตรวจสอบว่าไม่มีชีทใดถูกสั่ง Protect Sheet เอาไว้ (เพราะถ้าชีทใดถูก Protect จะไม่สามารถแก้ไขสูตร link ที่มีอยู่ในชีทนั้นได้เลย แม้จะสั่งผ่าน Excel ก็ตาม)
- 3.3. จากนั้นให้ใช้คำสั่ง **Data > Edit Links**
- 3.4. คลิกเลือกชื่อแฟ้มต้นทางที่แสดงในช่องรายชื่อ Source
- 3.5. กดปุ่ม Change Source แล้วคลิกไล่หาแฟ้มต้นทางที่เก็บไว้ในโฟลเดอร์ที่ต้องการ



4. เคล็ดลับอื่นๆที่สามารถนำมาใช้ร่วมกับคำสั่ง Edit Links จากรูปข้างต้น
 - 4.1. ปุ่ม Update Values ทำหน้าที่ update ข้อมูลจากแฟ้มต้นทางที่เลือกชื่อไว้จากช่อง Source ช่วยทำให้สามารถเปิดแฟ้มขึ้นมาโดยไม่ต้อง update ในตอนเปิดแฟ้มก็ได้ (เพราะหากสั่ง update ตอนเปิดแฟ้ม จะทำให้ Excel เสียเวลาอย่างมากในการไล่ update แฟ้มต้นทางที่มีทุกแฟ้ม) แล้วต่อเมื่อต้องการดึงข้อมูลใหม่จากแฟ้มต้นทางเมื่อใด จึงค่อยคลิกปุ่ม Update Values นี้ทีละแฟ้มที่ต้องการ
 - 4.2. ปุ่ม Open Source ทำหน้าที่เปิดแฟ้มต้นทาง ตามชื่อแฟ้มที่เลือกไว้จากช่อง Source
 - 4.3. ปุ่ม Break Link ทำหน้าที่ทำลายสูตรทั้งหมดในเซลล์ทุกเซลล์ที่มีสูตร link ข้ามแฟ้ม ให้กลายเป็นค่า value หรือผลลัพธ์ ซึ่งไม่แนะนำให้ใช้วิธีนี้เพราะ Excel จะทำลายสูตร link ในแฟ้มนั้นทั้งหมดโดยเราไม่มีทางทราบได้เลยว่าเป็นเซลล์ใดบ้างและยังไม่สามารถ Undo ดังนั้นหากต้องการทำลายสูตร link ในเซลล์ใด ให้ค้นหาตำแหน่งเซลล์ที่มีสูตร link โดยใช้ชื่อแฟ้มที่แสดงในช่อง Source สำหรับค้นหาตำแหน่งเซลล์นั้นเพื่อ Copy แล้ว Paste Special แบบ Value ทับเฉพาะเซลล์นั้นๆจะเหมาะสมกว่า

ระบบการรักษาความปลอดภัย

ในการ Link File

แม้ว่าขนาดตารางในชีทของ Excel มีจำนวน 1,048,576 rows และ 16,384 columns ช่วยรองรับกับข้อมูลจำนวนมากยิ่งกว่า Excel รุ่นก่อนก็ตาม แต่ไม่ได้ทำให้วิธีบันทึกข้อมูลที่ถูกต้องต่างไปจากที่ควร กล่าวคือ ในแฟ้มหนึ่งๆควรเลือกบันทึกเฉพาะข้อมูลที่ต้องใช้งานเรื่องที่เกี่ยวข้องกันไว้ด้วยกันในชีทเดียวกันเท่านั้น เช่น แยกแฟ้มเก็บข้อมูลการขายในช่วงแต่ละไตรมาสไว้ด้วยกัน แฟ้มแรกเก็บข้อมูลเดือน 1-3 แฟ้มที่สองเก็บข้อมูลเดือน 4-6 แฟ้มที่สามเก็บข้อมูลเดือน 7-9 และแฟ้มที่สี่เก็บข้อมูลเดือน 10-12 จากนั้นเมื่อต้องการดึงข้อมูลของไตรมาสใดมาใช้ ให้สร้างสูตร link ข้ามแฟ้มนำข้อมูลมาคำนวณ และใช้คำสั่ง Change Sources เพื่อเปลี่ยนการ link จากแฟ้มเดิมไปยังแฟ้มอื่นเพื่อนำข้อมูลช่วงไตรมาสที่ต้องการมาคำนวณโดยไม่ต้องเสียเวลาสร้างสูตร link ใหม่แต่อย่างใด

ใน Excel 2010 เป็นต้นมา ไม่ได้มีวิธีการสร้างสูตร link หรือมีระบบการ Change Sources ต่างไปจาก Excel รุ่นก่อน แต่ใน Excel 2010 มีระบบการรักษาความปลอดภัยในการใช้ข้อมูลที่เคร่งครัดกว่ารุ่นก่อนและส่งผลทำให้ Excel 2010 แสดงอาการและมีค่าเตือนต่างไปจากเดิมเมื่อมีการเปิดแฟ้มที่ link กัน บทความนี้จะทดสอบระบบการ link file ของ Excel 2010 โดยจำลองสถานการณ์ในแต่ละขั้นตอนมาให้ศึกษาตั้งแต่เริ่มต้นยังไม่มีข้อมูล จวบจนสร้างสูตร link และทดลองปรับระบบการรักษาความปลอดภัยว่าจะส่งผลต่อค่าเตือนเรื่องการ link อย่างไรบ้าง

ระบบความปลอดภัยใน Excel 2010 ที่จะนำมาทดสอบกับระบบการ link เช่น Trusted Location และ Trusted Document (เชิญ Download แฟ้มที่ใช้ในการทดสอบนี้ได้จาก www.ExcelExpertTraining.com/download/testlink.zip)

1. เริ่มต้นจากทดสอบขนาดแฟ้มที่เปิดขึ้นมาโดยยังไม่มีข้อมูลอะไรเลย สั่ง Save แฟ้มตั้งชื่อว่า Target.xlsx พบว่ามีขนาดแฟ้ม 8.18 KB
2. ในแฟ้มชื่อ Target นี้ ทดลองพิมพ์คำว่า Operations ลงไปในเซลล์ B2 พบว่า จากแฟ้มว่างๆซึ่งมีขนาด 8.18 KBแฟ้มมีขนาดใหญ่ขึ้นเล็กน้อยเป็น 8.55 KB พอพิมพ์ตัวเลข 20496 ลงไปในเซลล์ C2 แฟ้มมีขนาดใหญ่เพิ่มขึ้นเป็น 8.57 KB (แสดงว่าข้อมูลตัวเลขทำให้แฟ้มมีขนาดใหญ่ขึ้นช้ากว่าข้อมูลที่เป็นตัวอักษร)

3. สร้างแฟ้มฐานข้อมูลจำนวน 200 รายการ ชื่อ Source.xlsx มีขนาด 19.1 KB

	A	B	C	D	E	F	G	H	I
1									
2									
3		1	1/1/2012	Operations	Supplies & Services	Books & Subs	2508	2562	-54
4		2	21/1/2012	Security	Compensation	Training	2520	2703	-183
5		3	10/2/2012	R&D	Compensation	Bonuses	2512	3374	-862
6		4	1/3/2012	Sales	Equipment	Other	2514	3128	-614
7		5	21/3/2012	Training	Compensation	Commissions	2500	3164	-664
8		6	10/4/2012	Training	Compensation	Conferences	2527	3029	-502
9		7	30/4/2012	R&D	Facility	Rent	2506	3253	-747
10		8	20/5/2012	R&D	Equipment	Computer Hardware	2526	3822	-1296
11		9	9/6/2012	Accounting	Facility	Utilities	2523	3812	-1289
12		10	29/6/2012	Security	Supplies & Services	Books & Subs	2518	3621	-1103
13		11	19/7/2012	Security	Equipment	Photocopiers	2511	2756	-245
14		12	8/8/2012	Public Relations	Compensation	Benefits	2519	3076	-557
15		13	28/8/2012	Security	Compensation	Benefits	2515	2531	-16
16		14	17/9/2012	Shipping	Compensation	Training	2506	3343	-837
17		15	7/10/2012	Training	Compensation	Commissions	2501	3288	-787
18		16	27/10/2012	Accounting	Equipment	Software	2512	2871	-359
19		17	16/11/2012	Advertising	Equipment	Other	2511	3787	-1276

4. ในแฟ้ม Target เซลล์ C2 เมื่อสร้างสูตร link เพื่อหาค่ารวมจากแฟ้ม Source ของรายการ Department ที่มีชื่อ Operations ว่ามียอดของ Actual รวมเท่ากับ 20496 โดยใช้สูตร Array $\{=SUM((Source.xlsx!DEPARTMENT=B2)*Source.xlsx!ACTUAL)\}$ ทำให้แฟ้ม Target มีขนาดเพิ่มขึ้นจาก 8.57 KB เป็น 12.5KB (แสดงว่าสูตรคำนวณทำให้แฟ้มมีขนาดเพิ่มขึ้นได้ง่ายกว่าข้อมูลที่เป็นตัวเลขหรือตัวอักษร และการ link ข้ามแฟ้มทำให้เกิด cache data ขึ้นในแฟ้มปลายทาง ทำให้แฟ้มปลายทางมีขนาดใหญ่ขึ้นเป็นพิเศษจากการเก็บ cache data เอาไว้)

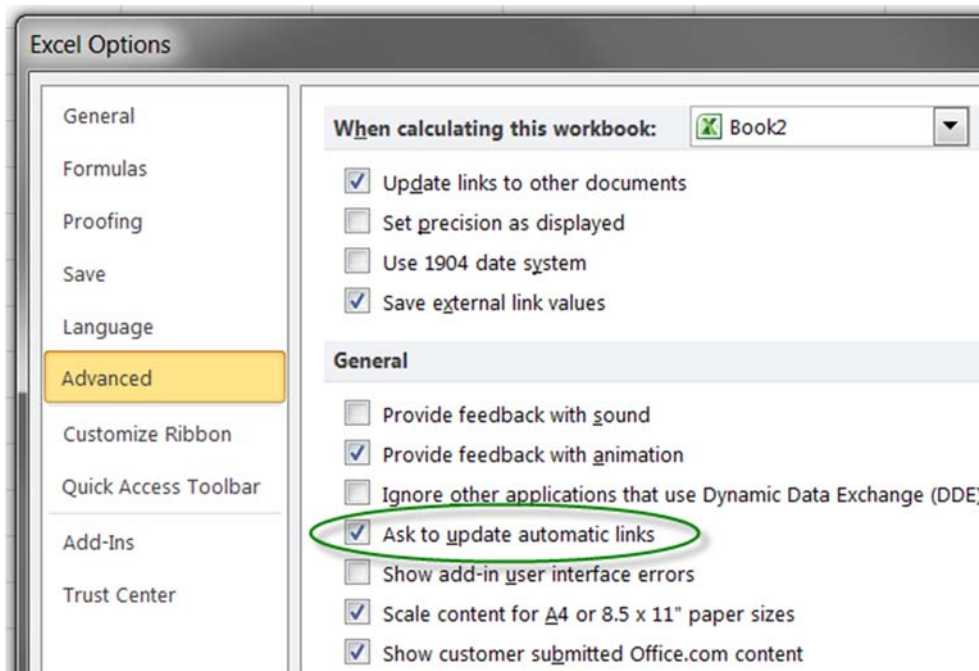
	A	B	C	D	E	F	G	H
1								
2		Operations	20496					
3								
4								

5. เมื่อปิดแฟ้ม Source ซึ่งเป็นแฟ้มต้นทาง ให้คงเหลือแฟ้มปลายทาง Target เปิดอยู่เพียงแฟ้มเดียว พบว่าแฟ้ม Target ยังคงมีขนาดเท่าเดิมเป็น 12.5 KB และเมื่อเปลี่ยนคำว่า Operations เป็น Security หรือ R&D เพื่อหาค่ารวมของรายการ พบว่าสามารถหาค่ารวมได้ตามต้องการ โดยไม่ต้องเปิดแฟ้ม Source และไม่ทำให้ขนาดแฟ้ม Target เปลี่ยนแปลงแต่อย่างใด (แสดงว่าใช้ข้อมูลจาก cache data ที่เก็บไว้ในแฟ้ม)

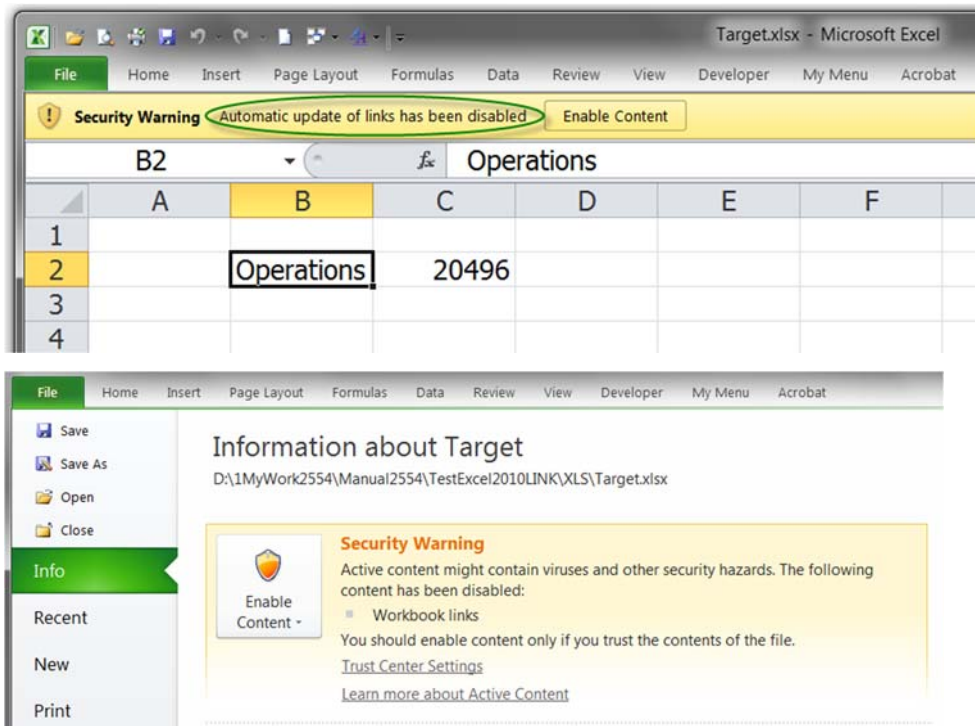
Target)

	A	B	C	D
1				
2		R&D	83592	
3	A	B	C	D
4	1			
	2	Security	92888	
	3			
	4			

6. ตรวจสอบคำสั่ง File > Options > Advanced > พบว่าระบบของ Excel กำลัง Ask to update automatic links ไว้



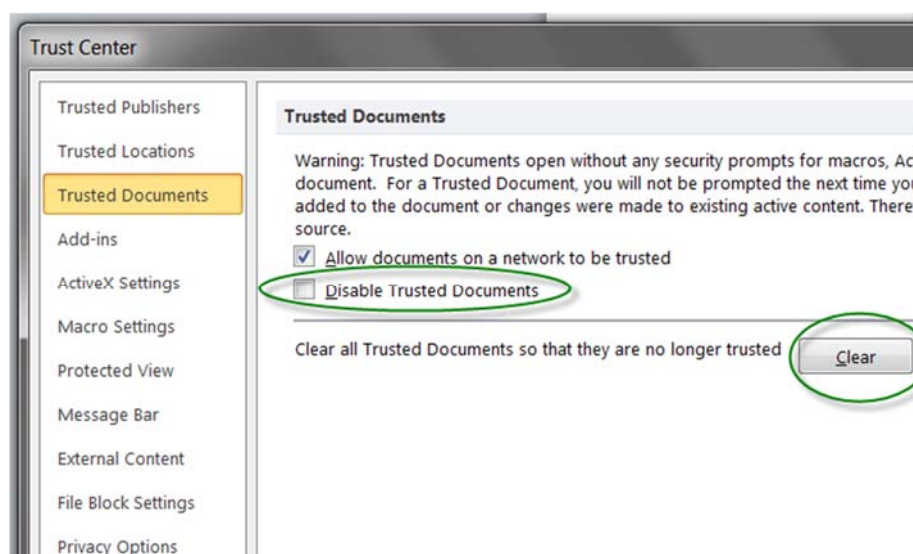
7. จากนั้นทดลองปิดแฟ้ม Target แล้วเปิดแฟ้ม Target ขึ้นมาใหม่เพียงแฟ้มเดียว ปรากฏแถบคำเตือน Security Warning ตรงด้านบนของจอเตือนว่า Automatic update of links has been disabled และเมื่อคลิกลงไปข้อความคำเตือนนี้ จะเปิดหน้าต่าง Info ขึ้นมาอธิบายว่า The following content has been disabled: Workbook links



8. ทดลองย้อนกลับไป Home โดยยังไม่ได้ Enable Content แล้วใช้คำว่า R&D หายอดรวมแทนคำว่า Operations พบว่าสามารถหายอดรวมของ R&D ได้เช่นเดิม
9. เมื่อเปิดแฟ้ม Source ขึ้นมา พบว่าคำเตือน Security Warning ในแฟ้ม Target หายไป แสดงว่ามีการ Enable Content เพื่อ Update Link ให้เองทันทีเมื่อเปิดแฟ้มต้นทางคู่กับแฟ้มปลายทาง
10. ปิดแฟ้ม Target
11. ในแฟ้ม Source ทดลองลบตัวเลข Actual ของ Department ที่เป็น Operations ให้เหลือเฉพาะรายการแรกซึ่งมีค่าเท่ากับ 2562 แล้วสั่ง Save แฟ้ม Source เก็บยอดใหม่นี้ไว้ เพื่อทดสอบว่า ถ้าเปิดแฟ้ม Target ที่ปิดไปก่อนแล้วจะแสดงยอดรวมของ Operations เท่ากับเท่าใดเมื่อเปิดแฟ้ม Target เพียงแฟ้มเดียว
12. ปิดแฟ้ม Source แล้วเปิดแฟ้ม Target ขึ้นมาเพียงแฟ้มเดียวจะพบคำเตือน Security Warning และยอดรวมของ Operations เท่าเดิมเช่นเดียวกับการทดลองในข้อ 7-8
13. คราวนี้กดปุ่ม Enable Content ทำให้คำเตือน Security Warning หายไปและได้ยอด Operations รวมเท่ากับ 2562 ตามยอดใหม่ที่เก็บไว้ในแฟ้ม Source แสดงว่าการกดปุ่ม Enable Content เป็นการสั่งให้ Excel update cache data ในแฟ้ม Target ตามข้อมูลใหม่ที่แก้ไขและเก็บไว้ในแฟ้ม Source
14. ปิดแฟ้ม Target โดยไม่ต้องสั่ง Save แล้วเปิดแฟ้ม Target ขึ้นมาอีกครั้งหนึ่ง ไม่พบว่ามี Excel 2010 เตือน Security Warning และยอดรวมของ Operations แสดงตัวเลข

เท่ากับ 2562 ทั้งนี้ แสดงว่าเมื่อเปิดเดิมแฟ้มซ้ำอีก Excel จะเปิดตามวิธีที่ผู้ใช้เคยเปิดให้ทันที

15. ตรวจสอบระบบของ Trust Center พบว่า ระบบตามปกติของ Excel 2010 กำหนดให้ใช้ Trusted Document ไว้เป็น default โดยไม่ได้กาช่อง Disable Trusted Documents เอาไว้ตั้งแต่แรกแล้ว ทำให้ Excel ถือว่าแฟ้ม Target เป็นแฟ้มที่เชื่อถือได้เพราะคราวก่อนที่เปิดแฟ้มนี้ ได้สั่ง Enable Content (Excel จะจำไว้ให้ทั้งๆที่ไม่ได้สั่ง save เก็บแฟ้ม Target ไว้ก็ตาม)



16. ตามภาพในข้อ 15 ทดลองกดปุ่ม Clear เพื่อกำหนดให้ทุกแฟ้มที่เคยเปิดถือว่าเป็นแฟ้มที่ไม่เคยไว้วางใจ แล้วเปิดแฟ้ม Target ขึ้นมาใหม่ พบว่า Excel เตือน Security Warning ตามการทดลองในข้อ 7 ตามเดิม
17. ตามภาพในข้อ 15 ทดลองกาช่อง Disable Trusted Documents เพื่อเลิกใช้ระบบ Trusted Document แล้วเปิดแฟ้ม Target ซ้ำหลายๆครั้ง พบว่า Excel เตือน Security Warning ตามข้อ 7 ทุกครั้งแม้จะกดปุ่ม Enable Content แล้วสั่ง Save แฟ้ม Target บันทึกค่าใหม่ที่ Update ไปแล้วก็ตาม
18. จากข้อ 17 ซึ่งเลิกใช้ระบบ Trusted Document ไปแล้ว ให้ทดลองใช้ Trusted Locations โดยกำหนดให้โฟลเดอร์ที่เก็บแฟ้ม Source และ Target เป็นโฟลเดอร์ที่ไว้วางใจได้ พบว่าเมื่อเปิดแฟ้ม Target จะไม่มีการเตือน Security Warning อีกต่อไป

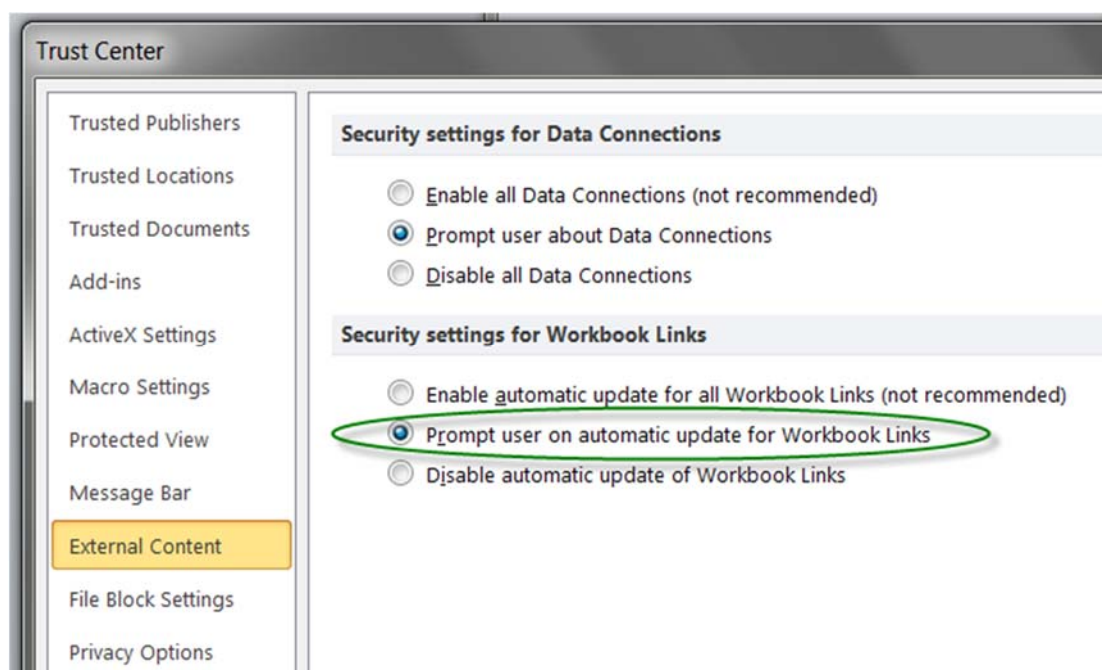
การทดลองข้างต้นทำให้ต้องเพิ่มความระมัดระวังในการใช้แฟ้ม Excel ที่ link กันมากขึ้นกว่าเดิม โดยเฉพาะอย่างยิ่งการใช้เครื่องคอมพิวเตอร์และแฟ้มร่วมกันหลายคน เพราะหากผู้ใช้ Excel คนใดคนหนึ่งเข้าไปปรับแก้ระบบการเตือน Security Warning ของ Excel 2010 ให้ต่าง

ไปจากเดิมหรือแตกต่างจากที่เคยใช้ร่วมกัน ย่อมส่งผลให้ผู้ที่ใช้ Excel คนอื่นเกิดความสับสนและนำข้อมูลไปใช้อย่างผิดพลาดขึ้นมาได้

หากจำเป็นต้องใช้เครื่องคอมพิวเตอร์และแฟ้มร่วมกันหลายคน ขอให้กาช่อง Disable Trusted Documents เพื่อเลิกใช้ระบบ Trusted Document และห้ามกำหนดโฟลเดอร์ใดเป็น Trusted Locations ไว้อย่างเด็ดขาด

นอกจากนี้เมื่อทดลองตั้งค่าช่อง Ask to update automatic links ตามภาพในข้อ 6 ทั้ง โดยจัดระบบให้ไม่ใช่ Trusted Locations และไม่ใช่ Trusted Document พบว่า Excel 2010 จะยังคงเตือน Security Warning เสมอ ทำให้เข้าใจว่าคำสั่ง Ask to update automatic links นี้ไม่ได้ส่งผลต่อการใช้งานใน Excel 2010 อีกต่อไป อาจยังคงมีประโยชน์กับแฟ้มที่สร้างขึ้นจาก Excel 2003 หรือ Excel รุ่นก่อนหน้านั้นที่นำมาเปิดด้วย Excel 2010 ก็เป็นไปได้

ในกรณีที่ไม่ต้องการให้ Excel 2010 เตือนเมื่อเปิดแฟ้มที่ link กัน สามารถปรับระบบ Trust Center ของ External Content โดยเลือกกาช่อง Enable automatic update for all Workbook Links (ซึ่งไม่แนะนำให้ใช้) แทนช่อง Prompt user on automatic update for Workbook links



หลักการตั้งชื่อและใช้ชื่อใน Excel

การตั้งชื่อและใช้ชื่อเป็นหัวใจสำคัญของการนำตำแหน่งอ้างอิงไปใช้ซ้ำ เนื่องจากหากปราศจากการตั้งชื่อที่ดีขึ้นมาใช้แทนตำแหน่งอ้างอิงของเซลล์แล้ว สูตรสั้นๆ ที่ดูไม่มีพิษภัยอะไร จะกลับกลายเป็นปัญหาอย่างมาก ยิ่งใน Excel 2007 ขึ้นไปมีพื้นที่ตารางกว้างเกิน Column และล้นกว่า Row จะทำให้ สูตร =JAN2009 ซึ่งเป็นสูตรที่นำค่ามาจากเซลล์ใน Column JAN ตัดกับ Row 2009 สามารถดูผิดๆ และถูกตีความว่าเป็นค่าของเดือน Jan ปีค.ศ.2009 แทนก็เป็นได้

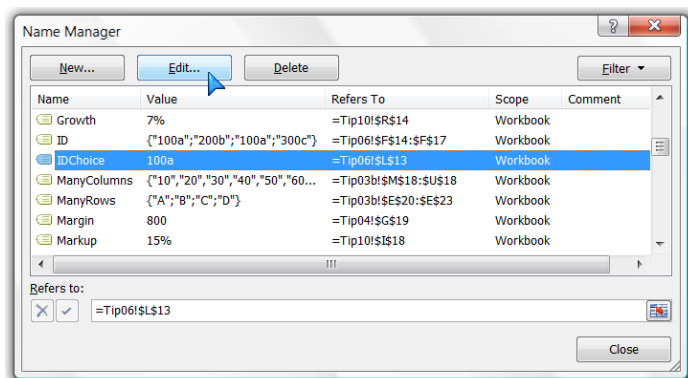
หากสังเกตขนาดของตัวอักษรแต่ละตัวในค่าแต่ละค่าที่ปรากฏในตาราง Excel จะพบว่า ชื่อใดก็ตามที่เป็นชื่อมาตรฐานของโปรแกรม Excel ซึ่งเราไม่สามารถเข้าไปแก้ไขเปลี่ยนแปลงให้เป็นชื่ออื่น ได้ถูกจัดให้ใช้ตัวอักษรตัวใหญ่ทั้งหมด (เว้นชื่อคำสั่งบน Ribbon หรือเมนู) เช่น ชื่อตำแหน่ง Column A B C D จนถึง XFD (Excel 2003 มี Column สุดท้ายชื่อ IV) และสูตรสำเร็จรูปของ Excel ทุกสูตร เช่น สูตร SUM, MAX, MIN, VLOOKUP ล้วนใช้ตัวอักษรตัวใหญ่ทั้งหมด ดังนั้นแทนที่จะพรวดใจกันตั้งชื่อแล้วใช้ชื่อที่เป็นตัวใหญ่ทั้งหมดตามแบบของชื่อของ Excel เราควรกำหนดหลักการตั้งชื่อและใช้ชื่อที่สื่อความหมายได้ดีขึ้นกว่าเดิม ดังนี้

1. หากแฟ้มใดจะถูกใช้ซ้ำ ขอให้ตั้งใจตั้งชื่อให้ดี เพื่อให้ไม่ว่าผู้ที่นำแฟ้มงานไปใช้ซ้ำนั้นจะเป็นตัวคุณซึ่งเป็นผู้สร้างแฟ้มนั้นเองหรือจะเป็นเพื่อนที่เพิ่งเปิดใช้แฟ้มนั้นเป็นครั้งแรกก็ตาม จะได้สามารถเปิดแฟ้มดูแล้วสามารถไล่ที่ไปที่มาของสูตรในเซลล์ได้ทันที แต่ถ้าแฟ้มใดถูกสร้างขึ้นมาเพื่อใช้งานง่าย ๆ เพียงครั้งเดียวแล้วทิ้ง ไม่ต้องนำกลับมาใช้อีก ก็ไม่จำเป็นต้องตั้งชื่อให้ดีก็ได้
2. ชื่อใดที่คุณตั้งขึ้นมาไม่ว่าจะเป็นชื่อชีท ชื่อแฟ้ม ชื่อโฟลเดอร์ ชื่อตำแหน่งเซลล์ (Range Name) ชื่อสูตร (Formula Name) หรือชื่อที่ตั้งให้กับค่าคงที่ (Constant Name) ให้ตั้งชื่อโดยใช้ตัวอักษรตัวใหญ่ผสมกับตัวเล็กเสมอ
3. เมื่อใดที่ต้องการสร้างสูตรหรืออ้างอิงชื่อใดก็ตาม ให้ใช้ตัวอักษรตัวเล็กทั้งหมดเท่านั้น เพราะเมื่อพิมพ์ลงไปในเซลล์แล้วกดปุ่ม Enter จะเกิดข้อสังเกตในตัวอักษรของแต่ละชื่อขึ้นมาทันทีกล่าวคือ
 - 3.1. ตัวอักษรที่กลายเป็นตัวใหญ่ทั้งหมด แสดงว่าเป็นชื่อเฉพาะของ Excel
 - 3.2. ตัวอักษรที่กลายเป็นตัวใหญ่ผสมตัวเล็ก แสดงว่าเป็นชื่อที่คุณตั้งไว้
 - 3.3. ตัวอักษรที่ยังคงเป็นตัวเล็กทั้งหมด แสดงว่าสะกดชื่อผิด ดังนั้น Excel จึงไม่ยอมรับชื่อนั้น ทำให้ขนาดตัวอักษรไม่เปลี่ยนแปลงตาม

4. ชื่อที่ตั้งขึ้นควรใช้ตัวอักษรตั้งแต่ 4 ตัวขึ้นไปเพื่อมิให้ซ้ำกับชื่อของ Column แต่ถ้าจำเป็นต้องตั้งชื่อที่มีตัวอักษร 2-3 ตัวก็ทำได้ เพราะการตั้งชื่อที่ใช้ตัวอักษรตัวใหญ่ผสมตัวเล็กจะช่วยให้การแยกแยะให้เห็นได้ชัดอยู่แล้วว่า Jan2009 มิได้มีความหมายเดียวกับ JAN2009 ที่เป็นเซลล์ใน Column JAN ตัดกับ Row 2009 (ถ้าต้องการตั้งชื่อที่เป็นตัวอักษรตัวเดียว ขอให้หลีกเลี่ยงตัว C และ R เพราะซ้ำกับตัวย่อของ Row และ Column)
5. Excel จะไม่ยอมรับชื่อ Range Name ที่มีการเว้นวรรค โดยหากสั่งให้ Excel ใช้ข้อความในเซลล์ตามขอบตารางมาเป็นชื่อโดยอัตโนมัติ จะถูกแก้ไขเครื่องหมายวรรคเป็นเครื่องหมายขีดล่าง _ (Underscore) ให้เอง และ Excel จะไม่ยอมรับชื่อ Range Name ที่เป็นตัวเลขล้วนหรือเป็นตัวเลขนำหน้าตัวอักษร
6. แม้จะไม่มีข้อห้ามในการตั้งชื่อเป็นภาษาไทย แต่ก็แนะนำให้ตั้งชื่อเป็นภาษาอังกฤษจะเหมาะสมกว่า เพราะหากใช้ Excel version ภาษาอังกฤษ เปิดแฟ้มที่มีการใช้ชื่อภาษาไทย จะอ่านชื่อภาษาไทยไม่ออก
7. ในการตั้งชื่อชื่อ ห้ามตั้งชื่อว่า History เนื่องจากเป็นชื่อที่ Excel สำรองไว้ใช้ภายใน
8. ในการตั้งชื่อ Range Name ห้ามตั้งชื่อว่า Criteria หรือ Extract เพราะชื่อ Criteria นี้จะถูกลบทิ้งหรือเปลี่ยนแปลงตำแหน่ง Extract จากเดิมได้เองทันทีที่ใช้คำสั่ง Advanced Filter
9. หากต้องการได้รับความสะดวกจากคำสั่ง Data Form ให้ตั้งชื่อตารางฐานข้อมูลว่า Database เพื่อให้ Excel ดึงข้อมูลมาไว้ใน Data Form ได้เองโดยที่คุณไม่ต้องเลือกตารางฐานข้อมูลนั้นไว้ก่อน
10. โปรดเข้าใจเพิ่มเติมว่า ในการเขียนชื่อสูตรและชื่อต่างๆที่ใช้ในการอบรมตลอดจนข้อมูลบนเว็บ www.ExcelExpertTraining.com นั้น ใช้วิธีเขียนชื่อโดยใช้อักษรตัวใหญ่ผสมตัวเล็กเสมอ เพื่อช่วยในการอ่านให้สามารถแยกแต่ละส่วนของคำได้ง่ายขึ้น เช่น VLookup, SumIF, SumProduct เป็นต้น ซึ่งเมื่อใดที่ต้องการนำชื่อไปใช้ ให้พิมพ์ตัวเล็กตลอดตามหลักการข้างต้น

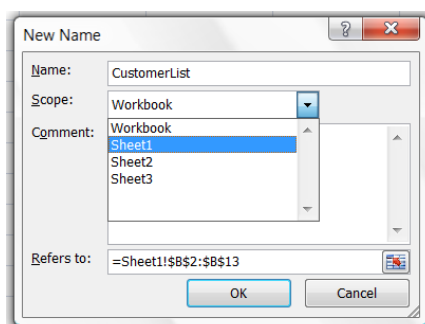
การตั้งชื่อแบบที่ตัวเอง สามารถใช้ช่อง Name Box หรือผ่านคำสั่งบนเมนูซึ่งทำได้ 2 วิธี คือ

1. หากต้องการตั้งชื่อเดียว ให้สั่ง **Formulas > Define Name**
2. หากต้องการตั้งชื่อหลายชื่อและสามารถย้อนกลับมาแก้ไขหรือตั้งชื่ออื่นใหม่อีกได้ในทันที ให้สั่งผ่าน **Formulas > Name Manager**



- 2.1. กดปุ่ม New เพื่อตั้งชื่อใหม่ หรือ
- 2.2. กดปุ่ม Edit เพื่อแก้ไขตำแหน่งอ้างอิงในชื่อ หรือ
- 2.3. กดปุ่ม Delete เพื่อลบชื่อทิ้ง

นอกจากนั้น Excel 2007 เป็นต้นมา ได้ปรับปรุงระบบการตั้งชื่อให้สนองต่อการใช้งานในระดับ File Level หรือ Sheet Level ได้สะดวกกว่าเดิมมาก กล่าวคือ



- การตั้งชื่อเพื่อใช้ในระดับ **File Level** เป็นระบบมาตรฐานที่ Excel เลือก Scope เป็น Workbook ไว้ให้ตั้งแต่ต้น เพื่อให้ชื่อที่ตั้งขึ้นในระดับนี้สามารถนำไปใช้ร่วมกันได้ทุกชีท
- การตั้งชื่อเพื่อใช้ในระดับ **Sheet Level** โดยคลิกเปลี่ยน Scope เป็นชื่อชีทที่ต้องการ ทำให้ชื่อที่ตั้งขึ้นในระดับชีทเป็นชื่อที่ใช้ในชีทนั้นและเป็นอิสระจากชีทอื่น ส่งผลให้สามารถตั้งชื่อซ้ำกันได้ในพื้นที่เดียวกัน (น่าสังเกตว่าเราสามารถกำหนด Scope ให้อยู่ต่างชีทกับชีทที่ Refers to ก็ยังได้)

การตั้งชื่อ Constant Name

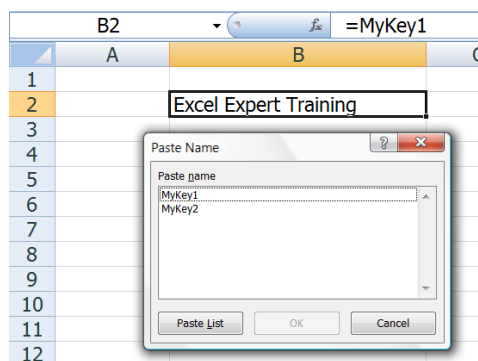
Constant Name เป็นชื่อที่ตั้งให้กับค่าคงที่ ซึ่งตามปกติในการดำเนินธุรกิจต้องเกี่ยวข้องกับค่าคงที่มากมาย นับตั้งแต่ ชื่อบริษัท ที่อยู่ ชื่อผู้บริหาร รหัสที่เป็นความลับ หมายเลขบัญชี ตัวเลขทางการเงินต่างๆ เช่น อัตราดอกเบี้ย ค่าเสื่อมราคา ตัวเลขเงินเดือน ดังนั้นแทนที่จะต้องบันทึกค่าเหล่านี้ซ้ำลงไปในเซลล์โดยตรง นอกจากจะเสียเวลาพิมพ์เองใหม่ทุกครั้งแล้ว ยังอาจพลั้งเผลอบันทึกข้อมูลต่างไปจากเดิม จึงขอแนะนำให้ตั้งชื่อให้กับค่าคงที่ทั้งหมดที่มี แทนที่จะต้องพิมพ์ชื่อยาวๆ ก็เพียงพิมพ์ชื่อสั้นๆ ที่จำง่ายกว่า และภายหลังหากมีการเปลี่ยนแปลงค่าที่เคยคิดค่าคงที่ ก็ให้ทำการแก้ไขค่าที่กำหนดให้กับชื่อที่เป็น Constant Name นี้เพียงครั้งเดียว ก็จะส่งผลให้ทุกเซลล์ที่นำ Constant Name ไปใช้มีค่าใหม่ตามการแก้ไขนั้นได้ทันที

สมมติว่า ต้องการตั้งชื่อ Constant Name ว่า "MyKey1" ให้มีค่าเท่ากับชื่อ "Excel Expert Training" ให้เริ่มจากพิมพ์ชื่อที่ต้องการตั้งลงไปในช่วง Name (โดยไม่ต้องใส่เครื่องหมายคำพูดปิดหัวท้ายชื่อ) จากนั้นพิมพ์ค่าที่ต้องการลงไปในช่อง Refers to โดยต้องมีเครื่องหมายเท่ากับนำหน้าเสมอและถ้ามีค่าเป็นตัวอักษรก็ต้องใส่ข้อความไว้ระหว่างเครื่องหมายคำพูดด้วยตามนี้

Name: MyKey1

Refers to: ="Excel Expert Training"

แต่ถ้าต้องการตั้งชื่อ Constant Name ให้มีค่าเป็นตัวเลข ก็ไม่ต้องใส่เครื่องหมายคำพูด แต่ต้องใส่เครื่องหมายเท่ากับนำหน้าเช่นกันตามนี้



Name: MyKey2

Refers to: =23187021

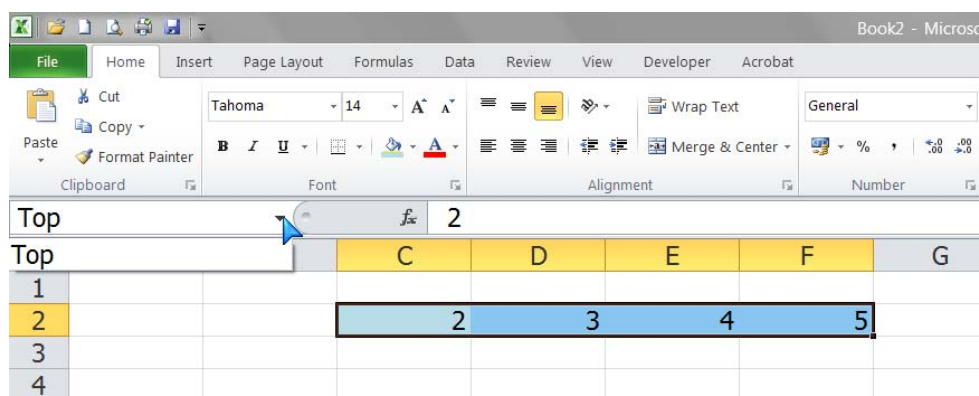
จากนั้นเมื่อต้องการนำชื่อ Constant Name นี้ไปใช้ในเซลล์ ให้พิมพ์ =mykey1 (สังเกตว่าถ้าพิมพ์เองให้พิมพ์ด้วยตัวเล็กตลอด) หรือกดปุ่ม **F3** เพื่อเปิดรายชื่อที่ตั้งไว้ในแฟม้นั้นขึ้นมาแสดงบนจอ แล้วให้ดับเบิลคลิกที่ชื่อที่ต้องการ (หรือคลิกปุ่ม OK) แล้วกดปุ่ม Enter เพื่อรับสูตร =MyKey1 ลงไปในเซลล์ จะได้คำว่า Excel Expert Training แสดงในเซลล์นั้น

หมายเหตุ ชื่อที่เป็น Range Name หรือ Formula Name มีขั้นตอนการใช้ชื่อเหมือนกับ Constant Name เช่นกัน

การตั้งชื่อ Range Name

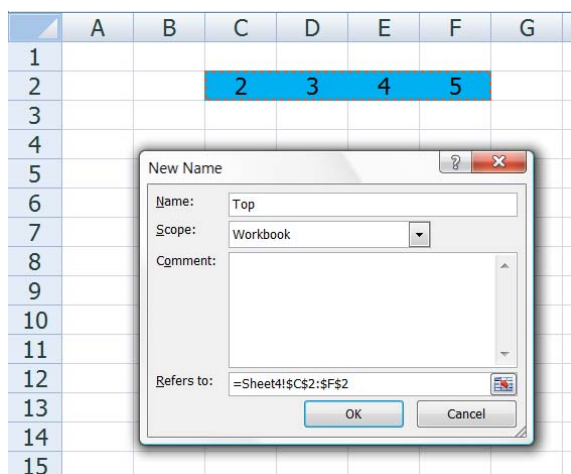
Range Name เป็นชื่อที่ตั้งให้กับเซลล์เดียวหรือพื้นที่ตาราง (ที่มีพื้นที่ติดกันหรือไม่ติดกันก็ได้) ซึ่งมีวิธีตั้งชื่อได้หลายวิธี ดังนี้

1. **วิธีใช้ Name Box** เป็นช่องด้านซ้ายมือของ Formula Bar วิธีนี้เป็นวิธีตั้งชื่อที่ง่ายที่สุด แต่ใช้กับการตั้งชื่อได้อย่างเดียว ไม่สามารถใช้ Name Box เพื่อแก้ไขหรือลบชื่อทิ้ง

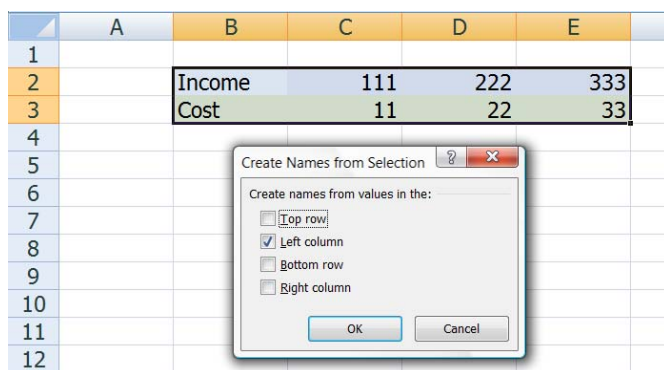


- 1.1. เริ่มจากคลิกเลือกเซลล์หรือตารางที่ต้องการ (หากต้องการเลือกเซลล์ที่ไม่ติดต่อกัน ให้กดปุ่ม Ctrl แล้วคลิกเลือกพื้นที่ตารางที่ต้องการไปเรื่อยๆ)
- 1.2. คลิกลูกศรด้านขวาของช่อง Name Box แล้วพิมพ์ชื่อ Range Name ที่ต้องการลงไป
ในช่อง Name Box ขอให้สังเกตว่า พอพิมพ์ตัวอักษรที่ละตัวของชื่อลงไป ช่อง Name Box จะเลื่อนไปหาชื่อเก่าที่ตั้งไว้ก่อนแล้วเพื่อคุณจะได้ตั้งชื่อใหม่ที่ไม่ซ้ำกับชื่อเดิมที่มีอยู่แล้ว
- 1.3. กดปุ่ม Enter เพื่อรับชื่อลงไปในช่วง Name Box
2. **วิธีใช้คำสั่ง Formulas > Define Name หรือ Formulas > Name Manager** โดยกรอกชื่อที่ต้องการตั้งลงไปในช่วง Name: แล้วคลิกลงไปในช่วง Refers to (โดยให้คลิกต่อท้ายตำแหน่งเดิมที่แสดงไว้ในช่อง Refers to: นี้) จากนั้นให้เลือกพื้นที่ตารางที่ต้องการ

แล้วกดปุ่ม OK เพื่อรับชื่อ



- 3. วิธีใช้คำสั่ง Formulas > Create from selection** วิธีนี้ช่วยนำข้อความที่อยู่ขอบตารางมาตั้งเป็นชื่อให้กับพื้นที่ตารางในฝั่งตรงข้ามกับข้อความนั้น เริ่มจากคลิกเลือกพื้นที่ตารางที่ต้องการตั้งชื่อโดยให้เลือกกรวมทั้งพื้นที่หัวตารางไว้ด้วย จากนั้นใช้คำสั่ง Formulas > Create from Selection จะพบหน้าจอต่อไปนี้เปิดขึ้นมา ให้ดูว่าตรงตามต้องการแล้วกดปุ่ม OK เพื่อรับชื่อ



Excel เลือกช่อง Left column ไว้ให้ เนื่องจาก Column ข้ายสุดของพื้นที่ตารางที่เลือกนั้นมีตัวอักษรบันทึกไว้ โดย Excel จะตั้งชื่อพื้นที่ตาราง C2:E2 ว่า Income และตั้งชื่อพื้นที่ตาราง C3:E3 ว่า Cost ตามข้อความใน Column ข้ายสุดให้ทันที ทั้งนี้ขอให้เข้าใจด้วยว่าชื่อ Income และ Cost เป็นชื่อที่ Excel ช่วยตั้งให้เพียงครั้งเดียวตอนที่ใช้คำสั่ง Create from selection นี้เท่านั้น และจะยังคงใช้ชื่อนี้อยู่โดยไม่เปลี่ยนแปลงตามข้อความใน Column ข้ายที่อาจมีข้อความต่างไปจากเดิมแต่อย่างใด

หากต้องการขยายขอบเขตของ Range Name ให้ใช้วิธีการ Insert Row หรือ Insert Column หรือใช้วิธีที่ 2 หรือวิธีที่ 3 ข้างต้นเพื่อเข้าไปกำหนดขอบเขตพื้นที่ตารางใหม่ให้มีขนาดใหญ่ขึ้นตามต้องการ

Range Name ที่ตั้งขึ้น สามารถนำไปใช้อ้างอิงผ่านสูตรสำเร็จรูปของ Excel ได้ทันที เช่น หากต้องการหาผลรวมของตัวเลขใน Range Name ชื่อ Income ให้เริ่มสร้างสูตรตามลำดับต่อไปนี้

1. พิมพ์ =sum(
2. กดปุ่ม **F3** เพื่อแสดงรายชื่อที่ตั้งไว้ในแฟ้ม (ในกรณีที่ต้องการนำชื่อมาจากแฟ้มอื่น ให้ย้ายไปอยู่ในแฟ้มที่มีชื่อ Range Name ที่ต้องการนั้นก่อนแล้วจึงกดปุ่ม F3)
3. ดับเบิลคลิกที่ชื่อ Income จะได้สูตร =sum(Income
4. กดปุ่ม Enter จะได้สูตร =SUM(Income)

แต่ถ้านำ Range Name ไปใช้อ้างอิงโดยตรงโดยไม่ได้ใช้อ้างอิงผ่านสูตรสำเร็จรูปของ Excel จะพบว่าหากตั้งชื่อให้กับตารางที่มีเซลล์ตั้งแต่ 2 เซลล์ขึ้นไป ชื่อนั้นจะนำมาใช้ได้เฉพาะในแนวคู่ขนานกับแนวนอนที่ตารางที่ตั้งชื่อไว้เท่านั้น เช่น หากตั้งชื่อตัวเลขหัวตารางในตำแหน่งจาก C2:F2 ว่า Top และตั้งชื่อตัวเลขข้างตารางตำแหน่งจาก B3:B5 ว่า Left จะส่งผลให้ใช้สูตร =Top*Left คำนวณหาค่าตอบได้เฉพาะในขอบเขตตารางที่แนวคู่ขนานของ Top ตัดกับ Left คือพื้นที่ตารางตั้งแต่ C3:F5 โดยสูตรนอกแนวคู่ขนานจะไม่คำนวณแต่คืนค่าเป็น Error #VALUE!

C6		=Top*Left						
	A	B	C	D	E	F	G	H
1								
2			2	3	4	5		
3		10	20	30	40	50	#VALUE!	#VALUE!
4		20	40	60	80	100	#VALUE!	#VALUE!
5		30	60	90	120	150	#VALUE!	#VALUE!
6			#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!
7			#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!
8								
9		C3 : =Top*Left						
10								

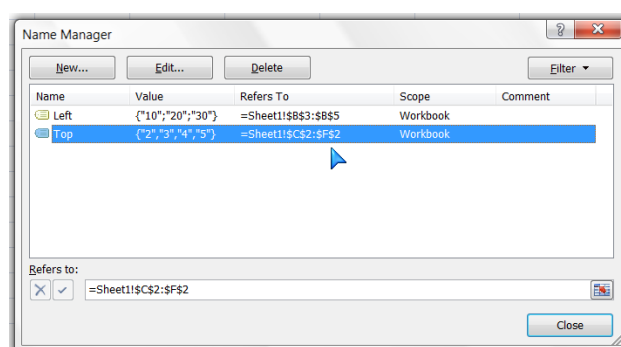
ใน Excel 365 พอสร้างสูตรลงไปเซลล์ C3 แล้วกดปุ่ม Enter จะพบว่าสูตรกระจายตัวลงไปในพื้นที่ C3:F5 ให้เอง โดยไม่ต้องเสียเวลา copy ไปวางแต่อย่างใด

วิธีตั้งชื่อตำแหน่งอ้างอิงแบบไร้ขอบเขตจำกัด

ขอย้อนนำตัวอย่างการคำนวณที่ตั้งชื่อหัวตารางด้านบนว่า Top มาคูณกับหัวตารางด้านซ้ายที่ตั้งชื่อว่า Left โดยใช้สูตร $=\text{Top} * \text{Left}$ มาวิเคราะห์ห้อย่างละเอียดว่า ด้วยเหตุใดสูตร $=\text{Top} * \text{Left}$ จึงสามารถคำนวณหาคำตอบได้เฉพาะพื้นที่ตารางในส่วนที่ตัดกันของแนวคู่ขนานของ Top ตัดกับ Left เท่านั้น ส่วนพื้นที่ตารางนอกขอบเขตแนวที่ตัดกันจะให้คำตอบเป็น Error

	A	B	C	D	E	F	G	H
1								
2			2	3	4	5		
3		10	20	30	40	50	#VALUE!	#VALUE!
4		20	40	60	80	100	#VALUE!	#VALUE!
5		30	60	90	120	150	#VALUE!	#VALUE!
6			#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!
7			#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!
8								
9			C3 : =Top*Left					
10								

เมื่อใช้คำสั่ง **Formulas > Name Manager** ย้อนไปตรวจสอบตำแหน่งอ้างอิงที่ Excel กำหนดให้กับตำแหน่งอ้างอิงที่มีชื่อว่า Top กับ Left จะพบว่า เป็นตำแหน่งอ้างอิงแบบ Absolute โดยมีเครื่องหมาย \$ ควบคุมตำแหน่งทั้งแนว Row และ Column ตามรูปต่อไปนี้



- ตำแหน่งอ้างอิงชื่อ Top มีตำแหน่งอ้างอิง Refers to: $=\text{Sheet1!}\$C\$2:\$F\2
- ตำแหน่งอ้างอิงชื่อ Left มีตำแหน่งอ้างอิง Refers to: $=\text{Sheet1!}\$B\$3:\$B\5

ด้วยสาเหตุที่ตำแหน่งอ้างอิงถูกควบคุมด้วยเครื่องหมาย \$ ทั้งแนว Row และ Column นี้เองที่ทำให้เกิดการจำกัดขอบเขตให้ใช้สูตรอ้างอิงถึงชื่อ Top กับ Left ได้เฉพาะในแนวคู่ขนาน

ดังนั้นหากคุณเข้าใจถึงเหตุและผลตามที่ได้กล่าวข้างต้น จะสามารถเลือกใช้เครื่องหมาย \$ ในการกำหนดขอบเขตแบบไม่จำกัดได้ โดยปรับตำแหน่งเครื่องหมาย \$ ตามหลักต่อไปนี้

- หากต้องการให้กำหนดตำแหน่งอ้างอิงให้หมายถึง **ทุกเซลล์ในแนวนอน** ให้ใช้ตำแหน่งอ้างอิงจากเซลล์เดียวตามแนวคู่ขนานกับเซลล์ที่จะใช้ชื่อนั้น และใส่เครื่องหมาย \$ **ควบคุมตำแหน่งเฉพาะ Row** เช่น C\$2
- หากต้องการให้กำหนดตำแหน่งอ้างอิงให้หมายถึง **ทุกเซลล์ในแนวตั้ง** ให้ใช้ตำแหน่งอ้างอิงจากเซลล์เดียวตามแนวคู่ขนานกับเซลล์ที่จะใช้ชื่อนั้น และใส่เครื่องหมาย \$ **ควบคุมตำแหน่งเฉพาะ Column** เช่น \$B3
- หากต้องการให้กำหนดตำแหน่งอ้างอิงให้หมายถึง **ทุกแนว** ให้ใช้ตำแหน่งอ้างอิงจากเซลล์เดียวตามแนวคู่ขนานกับเซลล์ที่จะใช้ชื่อนั้น และ**ไม่ใส่เครื่องหมาย \$**
- หากต้องการให้ตำแหน่งอ้างอิงใช้ได้กับทุกชีท ก็ไม่ต้องระบุชื่อชีทหน้าตำแหน่งอ้างอิงแต่ต้องทิ้งเครื่องหมาย ! ไว้ด้วยเพื่อหมายถึงทุกชีท เช่น !I5

เนื่องจากตำแหน่งอ้างอิงแบบที่ไม่ได้ใส่ \$ แบบ Absolute นี้ จะย้ายตำแหน่งไปตามแนวคู่ขนานกับตำแหน่งเซลล์สูตรที่ใช้ ดังนั้นก่อนที่จะใช้คำสั่ง **Formulas > Name Manager** เข้าไปกำหนดตำแหน่ง คุณจะต้องเลือกเซลล์ในแนวเดียวกับเซลล์ที่อ้างอิงในช่อง Refers to: ไว้ล่วงหน้าเสมอ ตามรูปตัวอย่างต่อไปนี้

H3		=TopRow*LeftColumn							
	A	B	C	D	E	F	G	H	I
1									
2			2	3	4	5		7	
3		10	20	30	40	50		70	
4		20	40	60	80	100		140	
5		30	60	90	120	150		210	
6									
7			C3 : =TopRow*LeftColumn				H3 : =TopRow*LeftColumn		
8									
9			11						
10			22		111		111	G10 : =Left2	
11			33		222				
12					333		333	G12 : =Left2	
13			66						
14					666				
15			C13 : =Sum3Above						
16					E14 : =Sum3Above				
17									

1. หากต้องการตั้งชื่อ **TopRow** และ **LeftColumn** (หรือชื่อใดก็ได้ตามใจชอบ) ให้มีตำแหน่งอ้างอิงที่หมายถึงทุกเซลล์ใน Row 2 และทุกเซลล์ใน Column B ตามลำดับ
 - 1.1. ให้คลิกเลือกเซลล์ **C3** ไว้ก่อนแล้วจึงสั่ง **Formulas > Name Manager**
 - 1.2. ตั้งชื่อ **TopRow** ให้มีตำแหน่งอ้างอิง Refers to: =Sheet1!C\$2
 - 1.3. ตั้งชื่อ **LeftColumn** ให้มีตำแหน่งอ้างอิง Refers to: =Sheet1!\$B3
2. หากต้องการตั้งชื่อ **Sum3Above** (หรือชื่อใดก็ได้ตามใจชอบ) ให้ทำหน้าที่หาผลรวมของเซลล์ 3 เซลล์ที่อยู่เหนือขึ้นไปจากเซลล์ที่ใช้ชื่อนี้โดยเว้นเซลล์ที่ติดกับเซลล์สูตร

- 2.1. ให้คลิกเลือกเซลล์ **C13** ไว้ก่อนแล้วจึงสั่ง **Formulas > Name Manager**
- 2.2. ตั้งชื่อ **Sum3Above** ให้มีตำแหน่งอ้างอิง Refers to: =SUM(Sheet1!C9:C11)
3. หากต้องการตั้งชื่อ **Left2** (หรือชื่อใดก็ได้ตามใจชอบ) ให้ทำหน้าที่หาค่าจากเซลล์ที่อยู่ด้านซ้ายถัดไป 2 เซลล์จากเซลล์ที่ใช้ชื่อนี้อยู่และให้ใช้กับทุกชีทได้ด้วย
 - 3.1. ให้คลิกเลือกเซลล์ **G10** ไว้ก่อนแล้วจึงสั่ง **Formulas > Name Manager**
 - 3.2. ตั้งชื่อ **Left2** ให้มีตำแหน่งอ้างอิง Refers to: =!E10

การตั้งชื่อ Formula Name

ในความเป็นจริงนั้นทั้ง Constant Name หรือ Range Name ต่างก็คือ Formula Name นั่นเอง เพราะในช่อง Refers to: ต้องใส่เครื่องหมายเท่ากับไว้ด้านหน้าค่าคงที่หรือตำแหน่งอ้างอิงไว้เสมอ จึงเท่ากันว่าเป็นการตั้งชื่อให้กับสูตร เพียงแต่ว่าสูตรที่เป็น Constant Name กำหนดให้มีค่าเป็นค่าคงที่ ส่วน Range Name ได้กำหนดให้มีค่าเท่ากับค่าที่เก็บไว้ในตารางตามที่ระบุไว้ในช่อง Refers to: ซึ่งนอกจากจะอ้างอิงถึงค่าคงที่หรือตำแหน่งอ้างอิงตามที่อธิบายผ่านไปแล้ว เรายังสามารถใช้ช่อง Refers to: ได้อีกหลายแบบ กล่าวคือ

- ใช้ผสมกันระหว่างค่าคงที่และตำแหน่งอ้างอิง เช่น $=5*Top*Left$ โดยเลข 5 ถือเป็นค่าคงที่ ส่วน Top กับ Left เป็นตำแหน่งอ้างอิง
- ใช้ร่วมกับสูตรสำเร็จรูปของ Excel เช่น $=SUM(Income)$
- ใช้ย่อสูตรยาวๆให้เป็นชื่อสั้นๆเพียงชื่อเดียว โดยทยอยนำชื่อที่ตั้งไว้มาคำนวณในสูตรยาวๆที่คุณสร้างขึ้น เช่น แทนที่จะตั้งชื่อเดียวให้กับสูตร $=(5*Top*Left)+Sum(Income)$ ก็สามารถตั้งชื่อให้กับสูตรแต่ละส่วนก่อนแล้วจึงนำชื่อมาใช้ต่อเป็น $=MyVar1+MyVar2$ โดยกำหนดให้ตั้งชื่อ MyVar1 มีค่า $=5*Top*Left$ และตั้งชื่อ MyVar2 มีค่า $=SUM(Income)$ ไว้ก่อน
- ใช้กำหนดตำแหน่งอ้างอิงแบบ Dynamic Range โดยใช้สูตร Offset ช่วยกำหนดขนาดตารางให้ย่อหรือขยายได้โดยอัตโนมัติตามขนาดของข้อมูล

การเปลี่ยนตำแหน่งอ้างอิงในสูตรที่มีอยู่ก่อนให้กลายเป็นชื่อที่ตั้งขึ้นในภายหลัง

	A	B	C	D	E	F
1						
2		Income	111	222	333	
3		Cost	11	22	33	
4			100	200	300	
5						

สมมติว่าสร้างสูตร $=C2-C3$ เอาไว้ในเซลล์ C4 เป็นตัวเลขกำไรขั้นต้น แล้ว Copy สูตรนี้ไปใช้ ในเซลล์ตั้งแต่ C4:E4 จากนั้นต่อมาได้ตั้งชื่อ Income และ Cost เป็น Range Name ให้กับพื้นที่ตารางตั้งแต่ C2:E2 และ C3:E3 ตามลำดับ หากต้องการให้สูตรในเซลล์ C4:E4 เปลี่ยนไปเป็นสูตร $=Income-Cost$ ตามชื่อที่ตั้งขึ้นในภายหลัง ให้เลือกพื้นที่ C4:E4 แล้วสั่ง **Formulas**

> **Define Name > Apply Names** แล้วเลือกชื่อ Income และ Cost (ซึ่ง Excel มักจะเลือกชื่อที่เกี่ยวข้องกับสูตรในพื้นที่ C4:E4 ให้เอง) จากนั้นกดปุ่ม OK

เคล็ดลับอื่นๆที่ควรทราบเกี่ยวกับชื่อที่บอกตำแหน่ง

1. ช่อง Name Box แสดงได้เฉพาะชื่อ Range Name ที่อยู่ในแฟมั้นเท่านั้น
2. การค้นหาตำแหน่งเซลล์หรือตารางที่ตั้งชื่อไว้ ให้ใช้ Name Box หรือกดปุ่ม **F5** เพื่อเลือกชื่อที่แสดงไว้หรือพิมพ์ชื่อที่ทราบลงไปเองก็ได้
3. หากต้องการ Copy ชื่อตำแหน่งอ้างอิงไปใช้ที่แฟมอื่นให้มีชื่อแบบเดียวกัน ให้เลือก Copy พื้นที่ตารางทั้งหมดที่เกี่ยวข้องและสูตรไปด้วยกัน (ซึ่งหมายถึง พื้นที่ตารางที่ระบุไว้ในตำแหน่งอ้างอิง และเซลล์สูตรที่อ้างอิงชื่อตำแหน่งอ้างอิงซึ่งอยู่ในพื้นที่ที่เลือกนั้นด้วย) จากนั้นให้ Paste ลงไปในแฟมใหม่ตามตำแหน่งเดิมของพื้นที่ตารางที่ Copy มา แล้วจึงส่งย้ายไปที่อื่นในภายหลังตามต้องการ
4. เซลล์เดียวมีชื่อได้หลายชื่อ แต่ช่อง Name Box แสดงชื่อแรกที่ตั้งตามลำดับตัวอักษร เช่น เซลล์ A1 ตั้งชื่อ Range Name ไว้หลายชื่อทั้ง Income, Sales, และ Revenue จะพบว่าเมื่อคลิกเข้าไปที่เซลล์ A1 ในช่อง Name Box จะแสดงว่ามีชื่อว่า Income เพราะตัวอักษร I มาก่อนตัวอักษร R หรือ S แต่ถ้าไล่ดูรายชื่อใน Name Box จะพบชื่ออื่นแสดงไว้ด้วย
5. ตัวชื่อ Range Name กับตำแหน่งอ้างอิงในช่อง Refers to: ไม่จำเป็นต้องอยู่ในแฟมเดียวกันก็ได้ โดยเราสามารถตั้งชื่อไว้ในแฟมหนึ่ง แต่กำหนดให้ตำแหน่ง Refers to: ไปอยู่ในแฟมอื่นได้ โดยใช้ Mouse คลิกเลือกตำแหน่งตารางในแฟมอื่นก็จะได้ตำแหน่งที่ต้องการมาใส่ในช่อง Refers to: ให้เอง
6. หาก Cut ตารางที่ตั้งชื่อ Range Name เอาไว้ไปวางไว้ในแฟมอื่น จะทำให้ตัวชื่อ Range Name ยังอยู่ในแฟมเดิม แต่ตำแหน่งอ้างอิงในช่อง Refers to: จะย้ายตามไปที่แฟมอื่น และยังส่งผลให้ช่อง Name Box ไม่แสดงชื่อ Range Name นั้นอีกต่อไป (โดยยังคงดูชื่อทุกประเภทได้จากคำสั่ง **Formulas > Name Manager**)
7. หากต้องการหาค่า ณ ตำแหน่งเซลล์ที่ตัดกันของตารางที่ตั้งชื่อไว้ (Intersection) ให้ใช้สูตรที่มีเครื่องหมายวรรคตอนระหว่างชื่อ เช่น =Top Left จะได้ค่าจากเซลล์ที่ตำแหน่งตัดกันของ Top ตัดกับ Left
8. หากต้องการสร้างสูตรที่อ้างอิงตำแหน่งอ้างอิงในตารางโดยตรง โดยไม่ต้องการให้สูตรที่เกิดขึ้นใช้ชื่อ Range Name ให้เริ่มจากเลือกเซลล์ตั้งแต่ 2 เซลล์ขึ้นไปไว้ก่อนแล้วจึงพิมพ์ = แล้วคลิกเลือกเซลล์ จะได้สูตรที่มีตำแหน่งอ้างอิงโดยไม่อ้างอิงชื่อ Range Name

9. Formula Name ที่ใช้ตำแหน่งอ้างอิงที่ไม่ใช่ Absolute Reference ในบางกรณีจะไม่คำนวณตามค่าตัวแปรที่เปลี่ยนไป ซึ่งแก้ไขได้โดยปรับสูตรให้คำนวณร่วมกับสูตร Now() หรือ Today() เช่น Sum3Above ใช้ Refers to: =SUM(Sheet1!C9:C11)*Now()/Now()
10. ชื่อที่ตั้งไว้จะถูกเก็บไว้ในแฟ้มที่ใช้ตั้งชื่อเท่านั้น หากต้องการทำให้แฟ้มใหม่มีชื่อที่ตั้งไว้แบบเดียวกับแฟ้มเดิม ให้ Copy เซลล์สูตรที่ใช้ชื่อในการอ้างอิงถึงตำแหน่งเซลล์ไปพร้อมกับเซลล์ที่ถูกตั้งชื่อไว้ไปด้วยกัน และในการสั่ง Paste ในแฟ้มใหม่ ต้อง Paste ลงไปในตำแหน่งเดิมด้วย จึงจะทำให้แฟ้มใหม่เกิดชื่อและมีตำแหน่งอ้างอิงถูกต้องตามแฟ้มเดิม

การใช้สูตรปรับตำแหน่งอ้างอิง

คำอธิบายจากนี้ไปจะมีประโยชน์ต่อเมื่อคุณมีพื้นฐานมาก่อนแล้วว่าตำแหน่งอ้างอิงมีประโยชน์อย่างไรและจะนำตำแหน่งอ้างอิงไปใช้ซ้ำโดยการ Copy หรือโดยการตั้งชื่อตำแหน่งอ้างอิงได้อย่างไร ซึ่งการนำตำแหน่งอ้างอิงไปใช้ซ้ำตามวิธีที่ใช้อยู่กันโดยทั่วไป คุณต้องเสียแรงเสียเวลา Copy หรือต้องสร้างสูตรใหม่ทุกครั้ง จึงเป็นเรื่องน่าคิดว่าจะมีวิธีใดบ้างที่จะทำให้ตำแหน่งอ้างอิงเดิมที่มีอยู่ปรับตำแหน่งได้เองโดยคุณไม่ต้องเข้าไปยุ่งเกี่ยวแก้ไขสูตรใหม่อีกเลย

วิธีที่ 1 : ใช้สูตร Indirect("Text บอกตำแหน่ง") แบบเซลล์เดียว

ลองนึกดูซิว่า ถ้าเดิมคุณสร้างสูตรเพื่อดึงค่าจากเซลล์ A1 มาใช้ด้วยสูตร =A1 จากนั้นหากต้องการทำให้สูตรในเซลล์เดิมนั้นแหละ ดึงค่าจากเซลล์ A2 มาแทน จะทำได้อย่างไรโดยไม่ต้องย้อนกลับไปแก้ไขสูตรให้เปลี่ยนเป็น =A2 แต่อย่างใด

	C7				
	A	B	C	D	E
1	11				
2	22				
3	33				
4			a		
5			1		
6			a1		C6 : '=C4&C5
7			11		C7 : =INDIRECT(C6)

แทนที่จะสร้างสูตรตรงๆลงไปว่า =A1 คราวนี้ให้ยอมเปลืองเซลล์หน่อย เริ่มจากใช้เซลล์ C4 และ C5 เป็นเซลล์รับตัวอักษร a และเลข 1 ลงไปตามลำดับ จากนั้นนำตัวอักษรกับตัวเลขมาต่อกันในเซลล์ C6 โดยใช้สูตร =C4&C5 จะได้คำว่า a1 ซึ่งเป็น Text ที่บอกตำแหน่งตามแบบมาตรฐานของตำแหน่งอ้างอิงใน Excel (ถ้าอยากลัดขั้นตอน จะใช้เซลล์ C6 สำหรับพิมพ์ตำแหน่งเซลล์ a1 ลงไปเลยก็ได้ โดยไม่ต้องเสียเซลล์ C4 และ C5 เพื่อแยกตำแหน่งแต่ละส่วนไว้ก่อน)

จากนั้นเซลล์ C7 ใช้สูตร =Indirect(C6) จะได้ค่าเป็นตัวเลข 11 ซึ่งเป็นค่าจากเซลล์ A1 นั่นเอง ที่นี้หากต้องการค่าจากเซลล์ใด ให้ใช้ตำแหน่งอ้างอิงตามค่าในเซลล์ C6 ซึ่งอาจเกิดจากการพิมพ์แยกส่วนของตัวอักษรและตัวเลขที่บอกตำแหน่ง Column และตำแหน่ง Row ลงไปใน

เซลล์ C4 และ C5 หรือจะใช้วิธีพิมพ์ตำแหน่งเซลล์ที่ต้องการลงไปเองโดยตรงในเซลล์ C6 เลยก็ได้ เช่น

- ถ้าพิมพ์คำว่า a2 ลงไปในเซลล์ C6 ก็จะได้คำตอบในเซลล์ C7 เท่ากับค่าจากเซลล์ a2
- ถ้าพิมพ์คำว่า a3 ลงไปในเซลล์ C6 ก็จะได้คำตอบในเซลล์ C7 เท่ากับค่าจากเซลล์ a3
- ถ้าพิมพ์คำว่า sheet3!a2 ลงไปในเซลล์ C6 ก็จะได้คำตอบในเซลล์ C7 เท่ากับค่าจากเซลล์ a2 แต่คราวนี้มาจากชีทชื่อ Sheet3 เป็นสูตรดึงค่าข้ามชีทตามชื่อชีทที่ต้องการ

วิธีที่ 2 : ใช้สูตร Indirect("Text บอกตำแหน่ง") แบบใช้ตำแหน่ง

ขอบเขตตาราง

วิธีนี้มีหลักการคล้ายกับวิธีแรกที่ใช้ตำแหน่งอ้างอิงจากเซลล์เดียว เพียงแต่ตำแหน่งอ้างอิงแบบขอบเขตตารางต้องอยู่ในโครงสร้างของ ตำแหน่งเซลล์ต้นทาง:ตำแหน่งเซลล์ปลายทาง เช่น A1:A3 จากนั้นจึงนำตำแหน่งอ้างอิงที่ได้ไปใช้กับสูตร Indirect

	D9			=SUM(INDIRECT(D8))		
	A	B	C	D	E	F
1	11					
2	22					
3	33					
4			a		a	
5			1		3	
6			a1	C6 : '=C4&C5	a3	E6 : '=E4&E5
7						
8				a1:a3	D8 : '=C6&":"&E6	
9				66	D9 : =SUM(INDIRECT(D8))	
10						
11				66	D11 : =SUM(INDIRECT(MyRange1))	
12				66	D12 : =SUM(MyRange2)	
13						

สมมติว่าคุณต้องการหาผลรวมของตัวเลขในเซลล์ A1:A3 ซึ่งปกติต้องใช้สูตร =Sum(A1:A3) ซึ่งสูตรนี้ยังใ้ก็หาผลรวมจากเซลล์ A1:A3 ไปตลอด แต่ถ้าต้องการให้หาผลรวมจากเซลล์ใดก็ได้ถึงเซลล์ใดก็ได้ละ จะสร้างสูตรอย่างไรครั้งเดียวแล้วไม่ต้องแก้ไขสูตรอีกเลยในภายหลัง

จากภาพข้างต้น มี Text ที่บอกตำแหน่งเซลล์ต้นทาง a1 ในเซลล์ C6 ตรงกับตัวอย่าง Indirect แบบเซลล์เดียว เพียงแต่คราวนี้ให้ใช้เซลล์ E4, E5, และ E6 สำหรับสร้าง Text ที่บอกตำแหน่งของเซลล์ปลายทาง a3

จากนั้นใช้เซลล์ D8 นำตำแหน่งเซลล์ต้นทางมาเชื่อมต่อกับเครื่องหมาย : แล้วเชื่อมต่อกับตำแหน่งเซลล์ปลายทางด้วยสูตร =C6&":"&E6 จะได้ค่า a1:a3

หากต้องการหาผลรวมก็สามารถนำ Text ที่บอกตำแหน่งในเซลล์ D8 ไปใช้ร่วมกับสูตร Sum และ Indirect เช่น เซลล์ D9 ใช้สูตร =Sum(Indirect(D8)) จะหาผลรวมตัวเลขจากตารางในขอบเขตที่ต้องการ

นอกจากนี้แทนที่จะนำค่าในเซลล์ D8 ซึ่งมีสูตรเป็น =C6&":"&E6 ไปใช้ต่อในสูตร Sum เพื่อหาผลรวม เรายังสามารถนำขอบเขตตำแหน่งอ้างอิงที่ได้ไปใช้ในสูตรอื่นได้เช่นเดียวกัน เพียงแต่ต้องใช้สูตร Indirect นำหน้าตำแหน่งอ้างอิงไว้เสมอ เช่น =Max(Indirect(D8)), =Min(Indirect(D8)), =Average(Indirect(D8)), =Index(Indirect(D8),2,1) เป็นต้น

ยิ่งกว่านั้นหากนำสูตร =\$C\$6&":"&\$E\$6 หรือ =Indirect(\$C\$6&":"&\$E\$6) ไปตั้งชื่อเป็น Formula Name ว่า MyRange1 และ MyRange2 ตามลำดับ (หรือชื่ออื่นใดก็ได้ โดยตำแหน่งอ้างอิงที่ใช้ใน Formula Name ต้องกำหนดตำแหน่งแบบ Absolute จึงจะหาค่าได้ถูกต้อง) จะช่วยให้การนำตำแหน่งอ้างอิงไปใช้ต่อไปสะดวกขึ้น เช่น จากเดิมต้องใช้สูตร =Sum(Indirect(D8)) ก็จะเทียบเท่ากับสูตร =Sum(Indirect(MyRange1)) หรือ =Sum(MyRange2) ซึ่งนอกเหนือจากจะเป็นสูตรที่ดูดีก็จริงแล้วยังไม่ขึ้นกับค่าในเซลล์ D8 ที่อาจถูกแก้ไขอย่างพลั้งเผลอให้ผิดเพี้ยนไปในภายหลังได้อีกด้วย

วิธีที่ 3 : ใช้สูตร Indirect("Text บอกตำแหน่ง") แบบใช้ชื่อ Range Name

	D9		=SUM(INDIRECT(C9))		
	A	B	C	D	E
1					
2			Bangkok	Chonburi	Lopburi
3		Year1	1	2	3
4		Year2	4	5	6
5		Year3	7	8	9
6		Year4	10	11	12
7					
8			Answer	Total	Insersection
9			Chonburi	26	5
10			Year2	15	
11					
12			D9 : =SUM(INDIRECT(C9))		
13			D10 : =SUM(INDIRECT(C10))		
14			E9 : =INDIRECT(C9) INDIRECT(C10)		

ตัวอย่างนี้ต่างจากตัวอย่างที่ผ่านมาเพียงแค่นำชื่อ Range Name มาใช้แทนตำแหน่งอ้างอิงโดยตรง เช่น หากตั้งชื่อพื้นที่ตาราง D3:D6 ว่า Chonburi และตั้งชื่อพื้นที่ตาราง C4:E4 ว่า Year2 (ให้ใช้คำสั่ง **Formulas > Create from selection** เพื่อกำหนดชื่อพื้นที่ตามค่าที่อยู่บนของตารางรวบเดียว)

จากนั้นใช้เซลล์ C9 และ C10 สำหรับพิมพ์ชื่อ Range Name ลงไป เช่น พิมพ์คำว่า Chonburi และ Year2 ลงไปตามลำดับ

- เซลล์ D9 หายอดรวมของ Chonburi ได้โดยใช้สูตร =Sum(Indirect(C9))
- เซลล์ D10 หายอดรวมของ Year2 ได้โดยใช้สูตร =Sum(Indirect(C10))
- เซลล์ E9 หาค่าของ Chonburi ในปี Year2 ได้โดยใช้สูตรหาค่าแบบ Intersection
=Indirect(C9) Indirect(C10)

ข้อควรระวังในการใช้สูตร Indirect

สูตรทุกสูตรของ Excel มีทั้งข้อดีและข้อด้อย สูตรบางสูตรที่พวกเราชอบใช้เพราะเป็นสูตรที่เขียนสั้นๆ แต่กลับทำให้ Excel ทำงานช้าลงก็มี (เช่น สูตร VLookup เป็นต้น) ซึ่งสูตร Indirect ก็มีข้อควรระวังอยู่ด้วยกล่าวคือ สูตร Indirect สามารถ link ข้อมูลข้ามแฟ้มได้ แต่แฟ้มต้นทางต้องเปิดขึ้นพร้อมกับแฟ้มปลายทางที่ใช้สูตรนี้อยู่ และเป็นสูตรหนึ่งที่ถูกเป็น Volatile Function

Volatile Function เป็นชื่อเรียกประเภทของสูตร ที่จะทำการคำนวณใหม่ทุกครั้งตามการคำนวณที่เกิดขึ้นในแฟ้ม หน้าที่เซลล์สูตรนั้นไม่ได้มีตำแหน่งอ้างอิงที่สัมพันธ์กับเซลล์อื่นมีการคำนวณ แม้แต่น้อย (ตามปกติสูตรจะคำนวณใหม่ตามต่อเมื่อมีเซลล์อื่นที่มีตำแหน่งอ้างอิงเกี่ยวข้องเกิดการคำนวณเท่านั้น) ดังนั้นหากใช้สูตรที่ทำหน้าที่แบบ Volatile นี้มากเกินไปจะทำให้แฟ้มคำนวณช้าลง แต่ถ้ามีการใช้สูตรเพียงไม่กี่เซลล์ก็ไม่ส่งผลเท่าใดนัก

สูตรที่จัดเป็นประเภท Volatile Function ได้แก่ สูตร Areas, Cell, Column, Columns, Indirect, Info, Now, Offset, Rand, Row, Rows, Today

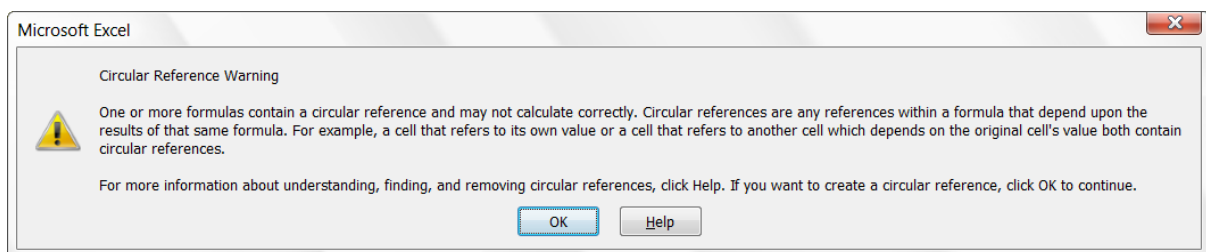
วิธีที่ 4 : ใช้สูตรเพื่อย้ายตำแหน่งอ้างอิงโดยอัตโนมัติ

แม้สูตร Indirect จะช่วยให้สามารถเปลี่ยนตำแหน่งอ้างอิงได้ก็ตาม แต่คุณยังคงต้องเป็นผู้กำหนดตำแหน่งใหม่ลงไปเองทุกครั้ง ซึ่ง Excel ยังมีสูตรอื่นอีกมากมายที่ทำหน้าที่เปลี่ยนย้ายตำแหน่งอ้างอิงได้เช่นที่สูตร Indirect ทำได้ แล้วยังมีความสามารถเหนือกว่าเพราะสามารถเปลี่ยนย้ายตำแหน่งอ้างอิงตามเงื่อนไขที่กำหนดหรือจะกำหนดให้เปลี่ยนย้ายตำแหน่งโดยอัตโนมัติได้อีกด้วย

- สูตรที่สามารถเปลี่ยนย้ายตำแหน่งอ้างอิงตามเงื่อนไขที่กำหนด เช่น IF, Choose, Index, Offset
- สูตรที่สามารถเปลี่ยนย้ายตำแหน่งอ้างอิงโดยอัตโนมัติ เช่น VLookup, Match ที่นำมาใช้ร่วมกับสูตร Index หรือ Offset, Address ที่นำมาใช้ร่วมกับสูตร Indirect

การกำหนดตำแหน่งอ้างอิงแบบวงกลม (Circular Reference)

เมื่อแรกเห็นชื่อการกำหนดตำแหน่งอ้างอิงแบบวงกลมน่าจะทำให้คิดไปว่าเป็นชื่อเรียกที่น่ารำกิด แต่เมื่อใดที่เกิดการกำหนดตำแหน่งอ้างอิงแบบวงกลมขึ้น เมื่อนั้นแฟ้มงานของคุณจะเข้าข่ายเสี่ยงที่เดียวว่าจะใช้แฟ้มนั้นต่อไปได้หรือไม่ เพราะตำแหน่งอ้างอิงแบบวงกลมอาจเกิดขึ้นโดยไม่ตั้งใจโดยไม่รู้ตัว หรือเกิดขึ้นอย่างเจตนาให้มันเกิดขึ้นก็ได้ ซึ่งถ้าใครที่ใช้ Excel เป็นประจำทุกวัน น่าจะเคยเจอคำเตือนต่อไปนี้แสดงขึ้นบนหน้าจอมาแล้วบ้าง



“การกำหนดตำแหน่งอ้างอิงแบบวงกลมอาจจะทำให้ Excel คำนวณผิดพลาด” นี่เป็นประเด็นสำคัญของคำเตือนที่แสดงขึ้นบนจอ

ที่น่ากลัวที่สุดก็คือคนบางคนที่ใช้ Excel มานาน แต่อาจไม่เคยพบคำเตือนตามหน้าจอข้างบนนี้เปิดขึ้นมาแสดงสักครั้งเดียว ซึ่งเป็นไปได้น้อยมากที่ผู้ใช้ Excel ไม่เคยพลั้งเผลอสร้างสูตรผิดแบบ Circular มาก่อน แต่อาจเป็นไปได้ว่าตัวเองได้สร้างสูตรแบบ Circular Reference นี้มาก่อนแล้ว เพียงแต่เขาเหล่านั้นสั่งให้ Excel เลิกเตือน โดยเฉพาะแฟ้มงานที่เป็นมรดกตกทอดใช้ต่อกันมาหลายต่อหลายรุ่น พอเปิดแฟ้มเข้าไปแก้ไขสูตรหรือโยกย้ายเซลล์ที่รุ่นพี่เขาสร้างไว้โดยไม่รู้จักที่ไปที่มาของสูตรในเซลล์เหล่านั้น ก็อาจทำให้เกิด Circular Reference ได้ทันที

ตัวอย่างการสร้างสูตรที่มีตำแหน่งอ้างอิงแบบวงกลมที่ง่ายที่สุด เช่น ในเซลล์ A1 สร้างสูตร =A1 ลงไป หรือในเซลล์ B3 สร้างสูตร = B1+B2+B3 ลงไป หรือสังเกตง่ายๆว่าสูตรในเซลล์ใดมีตำแหน่งอ้างอิงที่ระบุถึงเซลล์นั่นเองอยู่ด้วย ซึ่งนอกจากนี้หากมีสูตรที่ link ต่อกันไปหลายๆเซลล์แล้วกลับส่งค่ากลับมาในเซลล์ใดเซลล์หนึ่งในวงจรของเซลล์ที่ link กัน เช่น เซลล์ A1 มีสูตร =A3 และเซลล์ A2 มีสูตร =A1 แล้วเซลล์ A3 มีสูตร =A2 ย่อมทำให้เกิดตำแหน่งอ้างอิงแบบวงกลมได้เช่นกัน ซึ่งเรียกอีกแบบหนึ่งว่าเกิดการคำนวณแบบวงกลม (Circular Calculation)

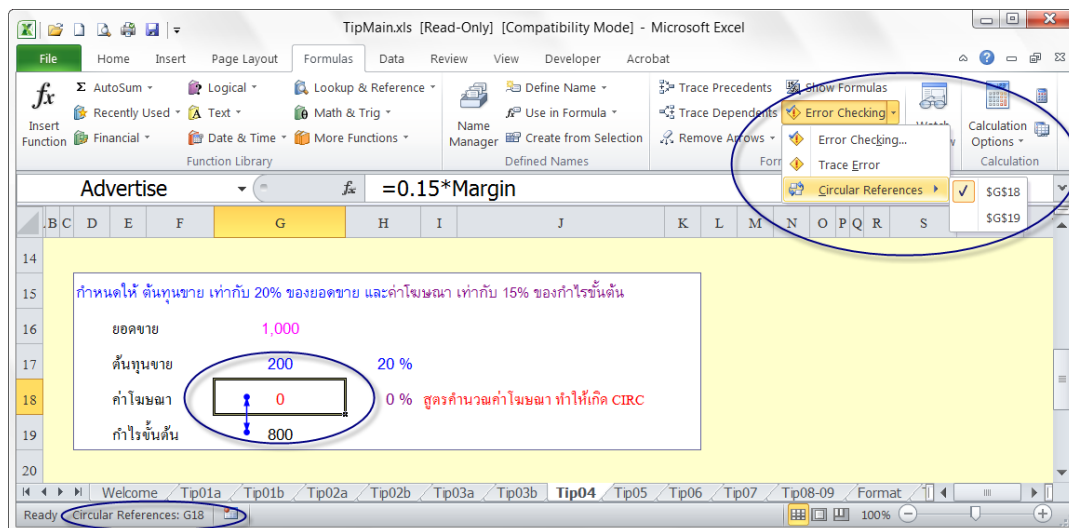
ในการดำเนินงานทางธุรกิจ มีงานหลายประเภทซึ่งอาจมีเงื่อนไขที่จำเป็นต้องสร้างสูตรที่มีการกำหนดตำแหน่งอ้างอิงแบบวงกลม อาทิเช่น

- ในเงื่อนไขการคำนวณต้นทุนสินค้า ซึ่งนำค่าใช้จ่ายทั้งทางตรงและทางอ้อมมา allocate เป็นต้นทุนของสินค้า จากนั้นหากสินค้านั้นกลับไปใช้เป็นวัตถุดิบให้กับสินค้าอื่น แล้วสุดท้ายก็นำสินค้าอื่นนั้นมาใช้เป็นองค์ประกอบร่วมกับสินค้าเดิมอีก
- หากบริษัทรับจ่ายภาษีเงินได้บุคคลธรรมดาแทนพนักงาน ซึ่งตามกฎหมายทางภาษีอากรกำหนดให้นำภาษีที่จ่ายแทนนั้นนำมาบวกเป็นเงินได้เพิ่ม แล้วเงินได้รวมที่เพิ่มขึ้นนี้ก็ต้องนำกลับไปย่อนคิดภาษีเงินได้อีก
- ค่าใช้จ่ายบางรายการ อาจมีข้อกำหนดทางภาษีว่าห้ามนำมาหักภาษีซ้ำอีก แต่ค่าใช้จ่ายนั้นกลับมีฐานการคำนวณที่เกิดขึ้นจากกำไรสุทธิ
- ในกรณีของการกู้เงิน ถ้าบริษัทนำดอกเบี้ยจ่ายเงินกู้ไปเพิ่มยอดเงินที่ต้องกู้เพิ่ม ย่อมทำให้ต้องจ่ายดอกเบี้ยเพิ่มและทำให้ต้องกู้เพิ่ม
- การของบประมาณค่าใช้จ่ายที่กำหนดว่าจะให้ตามอัตราร้อยละของกำไร ซึ่งถ้ากำไรเพิ่มก็จะทำให้ได้งบประมาณค่าใช้จ่ายเพิ่ม แต่ค่าใช้จ่ายที่เพิ่มขึ้น ย่อมส่งผลให้กำไรลดและทำให้ได้งบประมาณค่าใช้จ่ายลดลงตาม ซึ่งกลับทำให้กำไรเพิ่มขึ้น
- งานคำนวณอื่นๆ เช่น การควบคุมให้ Excel เพิ่มค่าตัวเลขครั้งละเท่าๆกัน โดยสร้างสูตร
 $\text{=Number}+\text{A1}$ ลงไปในเซลล์ A1 หรือการบันทึกประวัติตัวเลขสูงสุด โดยสร้างสูตร
 $\text{=Max}(\text{A1},\text{NumberRange})$ ลงไปในเซลล์ A1

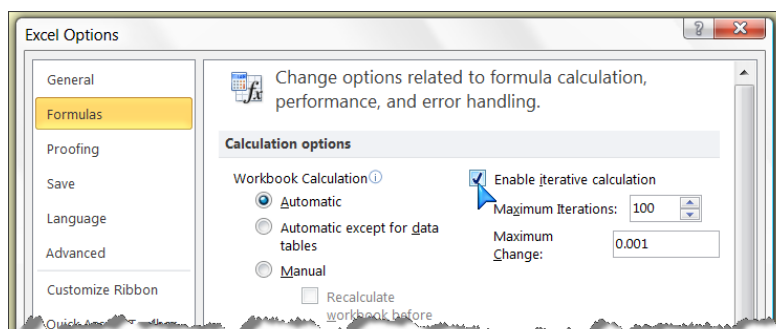
หมายเหตุ แทนที่จะใช้สูตรที่มีตำแหน่งอ้างอิงแบบวงกลม ซึ่งทำให้แฟ้มนั้นมีระบบการคำนวณแบบพิเศษที่ต้องสั่งให้ Excel ยอมรับก่อนจึงจะคำนวณได้คำตอบถูกต้อง เราสามารถใช้คำสั่งอื่นที่ทำหน้าที่สั่งให้ Excel คำนวณซ้ำได้ เช่น Goal Seek หรือใช้ Solver Add-in หรือใช้คำสั่งจาก Visual Basic ควบคุมการคำนวณแทน

ขั้นตอนการสร้างสูตรคำนวณแบบ Circular Reference

1. ให้เลือกสร้างสูตรอื่นๆที่คำนวณตามปกติทุกสูตร (ที่ไม่ต้องใช้การคำนวณแบบ Circular Reference) ให้เสร็จก่อน แล้วจึงสร้างสูตรที่ต้องคำนวณแบบ Circular Reference เป็นสูตรสุดท้าย เนื่องจากหากต้องการให้ Excel คำนวณแบบ Circular เมื่อใด คุณจะต้องสั่งให้ Excel เลิกเปิดค่าเตือน ซึ่งหากมีการสร้างสูตรผิดพลาดเปลืองไปสร้างสูตร Circular Reference ขึ้นโดยไม่เจตนา ก็ย่อมไม่มีการเตือนให้ทราบ
2. เมื่อสร้างสูตรคำนวณแบบ Circular Reference แล้วกดปุ่ม Cancel เพื่อปิดค่าเตือนบนหน้าจอตามรูปข้างต้นแล้ว จะพบคำว่า Circular References: ต่อด้วยตำแหน่งเซลล์แสดงขึ้นตรงขอบด้านล่างซ้ายของจอ พร้อมทั้งในพื้นที่ตารางก็จะมีเส้นลูกศรสีน้ำเงินชี้ตำแหน่งเซลล์ที่ทำให้เกิดการอ้างอิงแบบวงกลม ซึ่งหากต้องการให้ Excel แสดงชื่อตำแหน่งเซลล์ที่เกี่ยวข้องกับการอ้างอิงแบบวงกลมด้วย ให้สั่ง **Formulas > Error Checking > Circular References** แล้วจะพบตำแหน่งเซลล์ที่เกี่ยวข้องปรากฏต่อท้ายคำสั่งนี้



3. เมื่อต้องการให้ Excel ยอมรับการคำนวณแบบ Circular Reference ให้สั่ง **File > Options > Formulas > กางช่อง Enable iterative calculation**

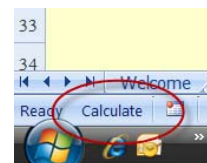


4. ช่อง Maximum Iterations และช่อง Maximum Change นั้นไม่จำเป็นต้องเข้าไปแก้ไขตัวเลขใดๆ ซึ่ง 2 ช่องนี้ทำหน้าที่สั่งให้ Excel คำนวณสูงสุดถึง 100 ครั้ง หรืออาจไม่ถึง 100 ครั้งก็ได้โดยให้คำนวณจนกว่าจะพบว่า ค่าตอบที่คำนวณได้ในแต่ละรอบให้ค่าแตกต่างกันน้อยกว่า .001 ก็จะหยุดคำนวณ ซึ่งการคำนวณเป็นรอบดังกล่าวนี้จะเกิดขึ้นทันทีที่มีการสร้างสูตรนั้นเสร็จแล้วกดปุ่ม Enter รับสูตรลงไปหรือเมื่อมีการกดปุ่ม F9 เพื่อสั่งให้ Excel คำนวณ

4.1. ให้เพิ่มตัวเลขจำนวนรอบสูงสุดในช่อง Maximum Iterations เมื่อพบว่าทุกครั้งที่กดปุ่ม F9 เพื่อสั่งคำนวณ จะได้ตัวเลขคำตอบต่างจากเดิมไปเสมอ ซึ่งหากเพิ่มตัวเลขในช่องนี้แล้วกดปุ่ม F9 พบว่าตัวเลขผลจากการคำนวณไม่เปลี่ยนแปลงต่อไปอีก ย่อมแสดงว่าได้คำตอบสุดท้ายแล้ว

4.2. ให้ปรับตัวเลขค่าความละเอียดในช่อง Maximum Change ต่อเมื่อต้องการให้ Excel คำนวณละเอียดขึ้น ซึ่งย่อมทำให้ต้องใช้จำนวนครั้งในช่อง Maximum Iterations เพิ่มขึ้นตามไปด้วย

5. หลังจากสั่งคำสั่งตามข้อ 3 แล้ว จะไม่พบข้อความหรือตำแหน่งเซลล์ที่แสดงบนหน้าจอตามตำแหน่งต่างๆที่เคยแสดงไว้ตามข้อ 2 อีกต่อไป แต่จะพบคำว่า Calculate แสดงค้างอยู่ตรงขอบจอด้านล่างซ้ายของโปรแกรม Excel แทน และขอให้สังเกตว่าทุกครั้งที่กดปุ่ม F9 จะทำให้คำว่า Calculate หายไปชั่วขณะหนึ่งแล้วกลับมาแสดงใหม่ (โดยไม่เกี่ยวข้องกับการสั่งให้คำนวณแบบ Manual ที่จะมีคำว่า Calculate แสดงขึ้นมาเช่นกัน แต่เมื่อกดปุ่ม F9 เพื่อสั่งคำนวณแบบ Manual จะทำให้คำว่า Calculate หายไปโดยไม่กลับมาใหม่อีก)



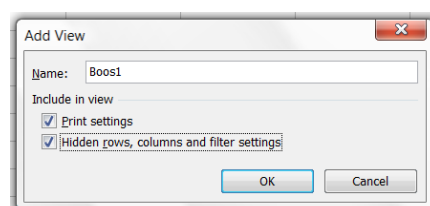
6. ในกรณีที่ต้องการตรวจสอบที่ไปที่มาของสูตรที่มีการใช้ตำแหน่งอ้างอิงแบบวงกลมว่ามีเซลล์ใดที่เกี่ยวข้องบ้าง ให้ย้อนกลับไปตัดกาช่อง **Enable iterative calculation** ทิ้งไป จากนั้น Excel จะแสดงตำแหน่งเซลล์ที่เกี่ยวข้องไว้ที่คำสั่ง **Formulas > Error Checking > Circular References** ซึ่งเซลล์เหล่านั้นอาจจะเป็นเซลล์สูตรที่สร้างไว้อย่างตั้งใจหรือเผลอแก้ไขสูตรให้ต่างจากเดิมไปจนทำให้เกิด Circular reference ขึ้นก็ได้ จากนั้นให้ทยอยลบสูตรในเซลล์ดังกล่าวทิ้งไปที่ละเซลล์ จนกว่าจะพบว่าเมื่อลบสูตรในเซลล์ใดเสร็จแล้ว ทำให้ไม่เกิดคำเตือน Circular ต่อไปอีก หลังจากนั้นจึงไล่สร้างสูตรใหม่นับพันเซลล์ทีเดียว โดยเฉพาะตารางคำนวณที่ต้องคำนวณรายการรายวันหรือรายเดือนต่อกันไปเป็นตารางขนาดใหญ่ ดังนั้นผู้ใช้ Excel ทุกคนต้องรู้ตัวทันทีเมื่อตัวเองเผลอสร้างสูตร

ที่มีตำแหน่งอ้างอิงแบบวงกลมขึ้นมาเป็นครั้งแรก หากปล่อยไว้ก็จะต้องเสียแรงเสียเวลาไล่ย้อนมาแก้ไขกันใหม่ที่เซลล์

7. เซลล์ที่ทำให้เกิด Circular นี้ ควรใส่สีให้ต่างจากเซลล์อื่น หรือทำเครื่องหมายพิเศษ หรือเขียนอธิบายกำกับไว้ด้วยเพื่อเตือนผู้ใช้งานให้แก้ไขสูตรอย่างระมัดระวัง
8. ควรตั้งชื่อแฟ้มที่มีสูตรที่มีตำแหน่งอ้างอิงแบบวงกลมให้มีชื่อที่สะดุดตา จะได้ไม่เผลอเปิดแฟ้มที่ต้องคำนวณแบบ Iterations นี้ขึ้นมาใช้งานพร้อมกับแฟ้มอื่น เพราะแฟ้มอื่นๆจะรับคำสั่ง Iterations นี้ตามไปด้วยโดยไม่จำเป็น
9. หากปิดโปรแกรม Excel แล้วเปิด Excel ขึ้นมาใหม่ จะพบว่า Excel ตัดกาช่อง Enable iterative calculation ทิ้งไปให้เสมอ

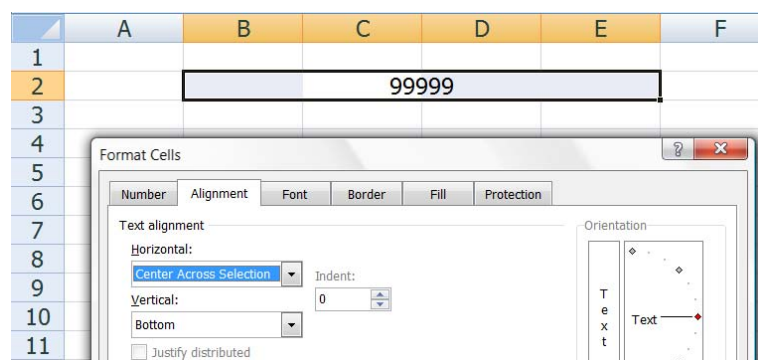
เคล็ดลับอื่นๆเกี่ยวกับการใช้ตำแหน่งอ้างอิง

1. ในกรณีต้องการหายอดรวมทุกชีตจากตำแหน่งเซลล์เดียวกัน ให้สร้างสูตรหายอดรวมแบบ **3 มิติ** ซึ่งจะอยู่ในโครงสร้างสูตร =SUM(ชื่อชีตแรก:ชื่อชีตสุดท้าย!ตำแหน่งเซลล์) เช่น =SUM(Sheet1:Sheet3!A1) ทำหน้าที่หายอดรวมของเซลล์ A1 ที่อยู่ในชีต Sheet1 ถึง Sheet3 ดังนั้นหากต้องการให้รวมข้อมูลจากชีตใด ให้ย้ายชีตนั้นไปวางไว้ระหว่าง Sheet1 ถึง Sheet3
 - 1.1. ให้เลือกเซลล์คำตอบ พิมพ์ =sum(
 - 1.2. คลิกเซลล์ในชีตแรกที่ต้องการ
 - 1.3. กดปุ่ม Shift ค้างไว้แล้วคลิกชีตสุดท้ายที่ต้องการ
 - 1.4. กดปุ่ม Enter
2. หากต้องการหายอดรวมจากทุกชีตเว้นชีตที่สร้างสูตรนี้ลงไป ให้พิมพ์สูตร =SUM('*! ตำแหน่งเซลล์) เช่น เมื่อสร้างสูตร =SUM('*!A1) ลงไปในชีต Sheet1 จะได้สูตร =SUM(Sheet2:Sheet3!A1)
3. ในกรณีที่ต้องการปรับโครงสร้างของตารางในชีตหนึ่งๆให้เป็นหลายแบบ โดยไม่จำเป็นต้องสร้างชีตใหม่สำหรับโครงสร้างตารางแต่ละแบบ แต่ให้ปรับโครงสร้างในชีตเดิมชีตเดียวนั้น แล้วสั่งให้ Excel จดจำโครงสร้างหน้าตาตารางโดยสั่ง **View > Custom Views > Add** จากนั้นให้ตั้งชื่อหน้าตาโครงสร้างตารางนั้นลงไป แล้วเมื่อใดที่ต้องการแสดงหน้าตาตารางชื่อใดให้สั่ง **View > Custom Views** แล้วเลือกชื่อ View ที่เคยตั้งชื่อไว้ (วิธีนี้สามารถตั้งชื่อ View ข้ามชีตได้ด้วย ทำให้



ไม่ต้องพิมพ์ชื่อชีท แล้วยังช่วยปรับ Print settings และ Filter settings ตามแบบที่กำหนดไว้ในขณะที่ตั้งชื่อให้ทันที)

4. ขอให้ออกแบบชีททุกชีทที่เก็บข้อมูลประเภทเดียวกันไว้ให้ใช้ตำแหน่งเซลล์เดียวกันของทุกชีท โดยไม่ต้องคำนึงว่าบางชีทจะมีข้อมูลครบหรือไม่ จากนั้นให้สั่ง Hide รายการที่ไม่มีข้อมูลนั้นออกไป และให้ใช้ Custom View ช่วยบันทึกแบบโครงสร้างตารางทั้งก่อนและหลังการ Hide ไปด้วย
5. พยายามหลีกเลี่ยงการใช้เซลล์ใดๆใน Column A และ Row 1 ทั้งนี้เพื่อให้ภาพตารางที่แสดงบนจอดูไม่อึดอัด และยังสามารถปรับขนาดความสูงความกว้างของ Column A และ Row 1 เพื่อช่วยปรับตำแหน่งตารางในหน้ากระดาษพิมพ์ได้ดีกว่าการกำหนด Margin แต่เพียงอย่างเดียว
6. ห้าม Merge เซลล์ที่เกี่ยวข้องกับการคำนวณ เพราะตำแหน่งอ้างอิงในสูตรที่อ้างอิงถึงเซลล์ที่ Merge กันไว้นั้นอาจให้ตำแหน่งอ้างอิงที่คลาดเคลื่อนจากความเป็นจริงเมื่อมีการ Copy สูตรที่อ้างอิงถึงตำแหน่งเซลล์ที่ Merge ไปใช้ที่เซลล์อื่น ดังนั้นหากต้องการปรับการแสดงข้อมูลให้เห็นว่าอยู่กลางช่วงเซลล์เช่นเดียวกับ Merge แต่ว่าเซลล์แต่ละเซลล์ยังคงเป็นเซลล์ของมันเองโดยมิได้รวมเป็นเซลล์เดี่ยวอย่าง Merge แต่อย่างใด ให้เลือกช่วงเซลล์นั้นแล้วคลิกขวา สั่ง **Format Cells > Alignment > Center Across Selection**



7. ในกรณีที่ต้องการ Cut ตารางจากชีทหนึ่งไปยังชีทอื่นหรือแฟ้มอื่น ให้เริ่มจากเลือกพื้นที่ตารางที่ต้องการจะ Cut ไปทั้งหมด (โดยต้องตรวจสอบว่าเป็นพื้นที่ตารางทั้งหมดที่มีสูตรที่เกี่ยวข้องกัน มิฉะนั้นจะมีเซลล์สูตรบางเซลล์ที่กลายเป็นสูตร link ข้ามแฟ้ม) จากนั้นเมื่อต้องการ Paste สามารถเลือก Paste ที่เซลล์ใดในชีทใหม่ก็ได้ โดย Excel จะปรับตำแหน่งอ้างอิงตามตำแหน่งในชีทใหม่ให้อัตโนมัติ
8. ในกรณีที่ต้องการ Copy ตารางจากชีทหนึ่งไปยังชีทอื่นหรือแฟ้มอื่น ให้เริ่มจากเลือกพื้นที่ตารางที่ต้องการจะ Copy จากนั้นเมื่อต้องการ Paste ให้เลือกเซลล์ในชีทใหม่ที่เป็นเซลล์เดียวกับเซลล์แรกของตารางที่ Copy ไว้ก่อน แล้วจึงสั่ง Paste

9. หากต้องการป้องกันไม่ให้ผู้อื่นนำแฟ้มไปแก้ไขสูตร ไม่จำเป็นต้อง Protect Sheet ก็ได้ แต่ให้สั่งตัดเครื่องหมาย \$ ออกจากเซลล์สูตรทุกเซลล์โดยใช้คำสั่ง Replace เพื่อให้สูตรทั้งหมดถูกแก้ไขหรือสร้างใหม่ได้ยากมาก (น่าสังเกตว่าพอตัดเครื่องหมาย \$ ทิ้งไป แฟ้มจะมีขนาดใหญ่ขึ้น)
10. หากต้องการป้องกันการ Insert Row หรือ Insert Column ในพื้นที่ช่วงใด ให้สร้างสูตร Array ลงไปพร้อมกันเป็นรั้วขอบเขตนั้น เช่น เลือกเซลล์ A1:A10 แล้วพิมพ์สูตร =0 หรือ ="" (= "" นี้ จะกลายเป็นรั้วไฟฟ้าที่มองไม่เห็นข้อมูลในเซลล์รั้วนั้นด้วย)แล้วกดปุ่ม Ctrl+Shift+Enter ลงไป จะได้รั้วป้องกันการ Insert Row ในช่วง Row 1 ถึง Row 10
11. หากต้องการใช้ Windows Explorer ในการ copy แฟ้ม Excel ที่ link กันหลายๆแฟ้มไปเก็บที่ drive อื่นหรือโฟลเดอร์อื่น ให้ Copy ทั้งโฟลเดอร์ที่เก็บแฟ้มที่ link กันนั้นไปพร้อมกันทั้งหมด เช่น ใน ถ้าเก็บแฟ้มที่ link กันไว้ในโฟลเดอร์ชื่อ MyExcelFile ก็ให้ Copy ตัวโฟลเดอร์ MyExcelFile ซึ่งมีแฟ้มที่ link กันอยู่ภายในนั้นไปทั้งหมด โดยในโฟลเดอร์ชื่อ MyExcelFile อาจมีโฟลเดอร์ย่อยสร้างไว้ด้วยก็ได้ เมื่อ Copy ไปไว้ใน drive อื่นแล้ว คุณจะเปลี่ยนชื่อโฟลเดอร์ MyExcelFile เป็นชื่ออื่นก็ได้ แต่ห้ามเปลี่ยนชื่อโฟลเดอร์ย่อยเพราะสูตร link ในแฟ้ม Excel จะจำชื่อโฟลเดอร์ย่อยไว้ในสูตร (แต่ไม่จำชื่อโฟลเดอร์ MyExcelFile)
12. แทนที่จะจัดเก็บแฟ้มไว้ในชื่อเดิม ขอให้จัดเก็บแฟ้มในชื่ออื่นทุกครั้งที่มีการแก้ไขใดๆในแฟ้มนั้น ทั้งนี้เพื่อทำให้มีแฟ้มเก่าเก็บไว้หลายๆรุ่น ดีกว่าการมีแฟ้มรุ่นเดียวแฟ้มเดียว ซึ่งเสี่ยงหากแฟ้มนั้นอาจเสียหายจนเปิดไม่ได้

แป้นพิมพ์ลัดที่เกี่ยวข้องกับตำแหน่งอ้างอิงและขอบเขตพื้นที่ตาราง

F2	ใช้แก้ไขข้อมูลหรือสูตรในเซลล์
F3	ใช้ใส่ชื่อตำแหน่งเซลล์หรือชื่อสูตรที่ตั้งไว้ในแฟ้มนั้นๆ ลงไปในสูตร หรือสั่ง Paste List ต่อ จะสรุปชื่อที่ตั้งไว้ทั้งหมดลงไปในตาราง
F4	ใช้ใส่เครื่องหมาย \$ กำกับตำแหน่งอ้างอิง หรือสั่งทวนคำสั่งสุดท้ายซ้ำ
F5	ใช้หาเซลล์ ตามตำแหน่งหรือชื่อตำแหน่งเซลล์ที่ทราบ หรือสั่ง Special ต่อ จะค้นหาตำแหน่งเซลล์ได้อีกหลายแบบ
F8	ใช้ควบคุมตำแหน่งเซลล์แรกไว้ไม่ให้ขยับเพื่อขยายตาราง (Extend)
F9	ใช้สั่งคำนวณทุกชีททุกแฟ้มที่เปิดขึ้น
Shift + F9	ใช้สั่งคำนวณเฉพาะชีทที่เลือกอยู่ชีทเดียวเท่านั้น
F2 ตามด้วย F9	ใช้แก้สูตรในเซลล์เพื่อดูว่ามีค่าที่แท้จริงเป็นอย่างไร แล้วกด Esc
Ctrl + * หรือ Ctrl+a	ใช้เลือกพื้นที่ตารางซึ่งมีข้อมูลติดต่อกันไป (Current Region)
Ctrl + .	ใช้ย้าย Active Cell ไปตามเซลล์หัวมุมของตารางที่เลือกไว้
Ctrl + Shift + → Shift + End →	ใช้เลือกพื้นที่ตารางจากเซลล์ที่เลือกไว้ไปจนสุดทางด้านขวา หรือตามทิศทางของลูกศรที่กด
Ctrl + PgUp	ใช้เลือกชีทถัดไป
Ctrl + PgDn	ใช้เลือกชีทก่อนหน้า
Ctrl + Tab	ใช้เลือกแฟ้มถัดไปที่เปิดอยู่
Ctrl + Shift + Tab	ใช้เลือกแฟ้มก่อนหน้าที่เปิดอยู่
Ctrl + Shift + Enter	ใช้สร้างสูตร Array ซึ่งจะมีเครื่องหมาย { } ปิดหัวท้ายสูตร

หลักการแยกตัวแปร

1. ตัวแปร คือ ค่าที่มนุษย์สามารถกำหนดให้มีค่าเปลี่ยนแปลงต่างไปจากเดิม ตัวแปรอาจเป็นได้ทั้งค่าที่เป็นตัวเลข ตัวอักษร หรือแม้แต่สูตร
2. เซลล์รับตัวแปรอาจใช้เซลล์เดียวหรือหลายเซลล์ ขึ้นกับประเภทของตัวแปรนั้นๆว่ามีการเปลี่ยนแปลงไปตามระยะเวลาหรือเงื่อนไขอื่นใดหรือไม่ ถ้าเป็นค่าคงที่ซึ่งใช้ค่าเดียวไปตลอด ก็สามารถให้เซลล์รับค่าเพียงค่าเดียวเซลล์เดียวในแฟ้มหนึ่ง แต่ถ้าเป็นค่าที่มีความผันผวนเปลี่ยนแปลงไปตามระยะเวลาก็จำเป็นต้องใช้หลายเซลล์ เช่น ตัวแปรประเภทราคาสินค้า ควรกำหนดให้แต่ละช่วงเวลามีราคาสินค้าแตกต่างกันไปเพื่อให้เกิดความยืดหยุ่นในการคำนวณ ก็จำเป็นต้องใช้หลายเซลล์เพื่อรับตัวแปรประเภทราคาสินค้าที่มียอดแตกต่างกันไปในแต่ละช่วงเวลา
3. ควรคิดเผื่อไว้ตั้งแต่ต้นเสมอว่า ค่าคงที่ทุกค่าเป็นตัวแปรที่ค่าอาจเปลี่ยนแปลง เพราะค่าคงที่ใดๆที่คิดว่าคงที่ในวันนี้ อาจเป็นไปได้ว่าไม่ใช่ค่าที่จะคงที่ไปตลอด ดังนั้นหากไม่มั่นใจว่าจะคงที่ไปตลอดหรือไม่ ควรออกแบบตารางตั้งแต่แรกเผื่อไว้ให้สามารถเปลี่ยนแปลงตัวแปรตามระยะเวลาได้ด้วย เช่น อัตราภาษีธุรกิจเฉพาะ ซึ่งอาจถูกปรับเปลี่ยนให้ต่างไปจากเดิมในอนาคต
4. ลูกเล่นทางธุรกิจเป็นตัวแปรได้เหมือนกัน อย่าคิดว่าวิธีดำเนินการทางธุรกิจต้องเป็นแบบปัจจุบันไปตลอด คุณต้องเผื่อเซลล์รับตัวแปรที่อาจเกิดขึ้นหรือไม่เกิดขึ้นไว้ก่อนตั้งแต่แรก ดีกว่าที่จะย้อนกลับไปแก้ไขตารางแทรกตัวแปรเพิ่มในภายหลัง ขอให้ออกแบบโครงสร้างตารางเผื่อไว้ก่อนดีกว่าขาด
5. ในกรณีที่ตัวแปรมีเงื่อนไขซับซ้อน ควรแยกตัวแปรประเภทนั้นไปเก็บเป็นตารางเฉพาะหรือเป็นชีทเฉพาะหรือแม้แต่เป็นแฟ้มเฉพาะสำหรับตัวแปรนั้นๆ โดยให้ออกแบบตารางตามแบบโครงสร้างฐานข้อมูลที่ดีด้วย
6. ควรใส่สูตรให้เห็นเซลล์รับตัวแปรแยกชัดเจนแตกต่างจากเซลล์สูตรคำนวณและกำหนดให้ Excel เปิดรับค่าใหม่ได้ โดยคลิกขวาไปในเซลล์ตัวแปรนั้น สั่ง Format Cells > Protection แล้วตัดกาช่อง Locked ทิ้งไป จากนั้นสั่ง Review > Protect Sheet แล้วจะพบว่าเมื่อกดปุ่ม Tab จะย้ายตำแหน่งเซลล์ไปตามเซลล์ที่เป็นตัวแปรให้เอง

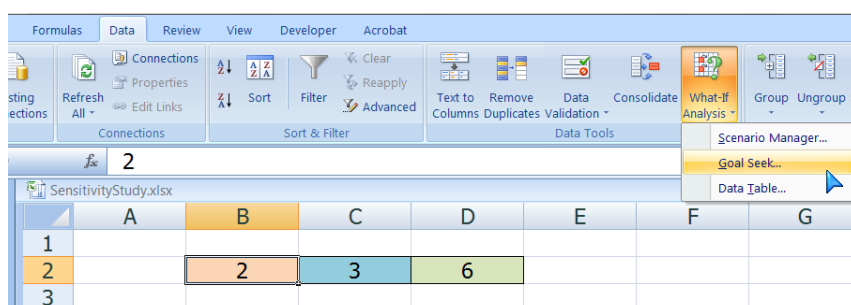
วิธีใช้ Goal Seek

เลขอะไรเอ่ยนำไปคูณกับเลข 3 แล้วได้ผลลัพธ์เป็นเลข 30 คุณก็ต้องตอบว่าเลข 10 เพราะคุณคิดสูตรคูณในใจออกมาได้ทันทีเลยใช่ไหมว่า $10 \times 3 = 30$ แต่ถ้าเปลี่ยนโจทย์ใหม่ว่า เลขอะไรเอ่ยนำไปคูณกับเลข 3 แล้วได้ผลลัพธ์เป็นเลข 77.77 คราวนี้สมองของคุณก็จะคิดในใจอย่างรวดเร็วว่า “ไม่ไหว คิดไม่ไหว” ถ้าต้องเสียเวลาคิดเครื่องคิดเลขหรือสร้างสูตรเอาเลข 77.77 ตั้งแล้วหารด้วย 3 เพื่อหาเลขอะไรเอ่ยตัวนั้นก็ได้อ ย่อมแสดงว่าคุณใช้ Goal Seek ไม่เป็น

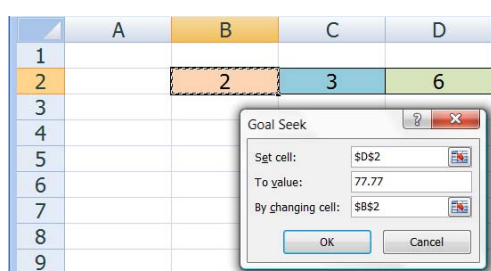
Goal Seek เป็นเครื่องมือที่ช่วยคิดย้อนโดยคุณไม่ต้องเสียเวลาสร้างสูตรใหม่เพื่อคิดย้อนเอง แม้แต่น้อย เพียงแต่ว่า Goal Seek จะทำงานได้ต่อเมื่อมีเซลล์รับตัวแปรที่สามารถถูก Excel ทำหน้าที่เปลี่ยนแปลงเป็นเลขใหม่ตามต้องการ แต่หากเป็นเซลล์สูตรจะไม่สามารถใช้ Goal Seek เข้าไปแก้ไขสูตรใดๆในเซลล์ให้เป็นค่าใหม่ได้

จากโจทย์เดิม เลขอะไรเอ่ยนำไปคูณกับเลข 3 แล้วได้ผลลัพธ์เป็นเลข 77.77

1. สร้างตัวอย่าง 2×3 ได้ 6 ตามตัวอย่างที่ผ่านมา
2. คลิกเลือกเซลล์ B2 ไว้ก่อนแล้วสั่ง Data > What-If Analysis > Goal Seek

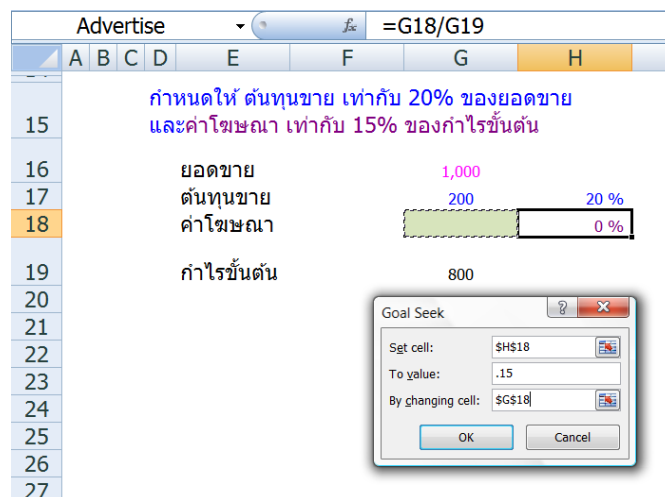


3. คลิกในช่อง Set cell: แล้วให้คลิกต่อไปที่เซลล์ D2 โดยเซลล์นี้ต้องเป็นเซลล์สูตรคำนวณ ส่วนช่อง To value: ให้พิมพ์เลข 77.77 ลงไป แล้วให้คลิกในช่อง By changing cell: แล้วคลิกต่อไปที่เซลล์ B2 โดยเซลล์ B2 นี้ต้องเป็นเซลล์รับตัวแปรที่กำลังต้องการหา ถือเป็นตัวเลขคำตอบใหม่ที่ต้องการ



กำไรลด ยอดส่งผลให้ค่าโฆษณาตาม พอค่าโฆษณาตกลง กำไรก็ต้องเพิ่ม กลายเป็นการคำนวณแบบวงกลมที่ไม่สามารถหาคำตอบสุดท้ายได้โดยวิธีธรรมดาทั่วไป

คุณสามารถแก้ปัญหานี้โดยใช้ Goal Seek โดยเริ่มต้นจากคลิกเลือกเซลล์ G18 ไว้ก่อนแล้วสั่ง Data > What-If Analysis > Goal Seek



1. คลิกในช่อง Set cell: แล้วให้คลิกต่อไปที่เซลล์ H18 โดยเซลล์นี้เป็นเซลล์สูตร คำนวณหาว่าค่าโฆษณาเป็นร้อยละเท่าใดของกำไรขั้นต้น =G18/G19
2. ช่อง To value: ให้พิมพ์เลข .15 ซึ่งเป็นตัวเลขคำตอบที่ต้องการ
3. คลิกในช่อง By changing cell: แล้วคลิกต่อไปที่เซลล์ G18 เพื่อให้หาตัวเลขค่าโฆษณา
4. กดปุ่ม OK > OK จะได้ตัวเลขคำตอบค่าโฆษณาเท่ากับ 104.50377964375 ที่ Goal Seek คิดให้

Advertise					f_x		104.50377964375	
	A	B	C	D	E	F	G	H
15	กำหนดให้ ต้นทุนขาย เท่ากับ 20% ของยอดขาย และค่าโฆษณา เท่ากับ 15% ของกำไรขั้นต้น							
16	ยอดขาย						1,000	
17	ต้นทุนขาย						200	20 %
18	ค่าโฆษณา						105	15 %
19	กำไรขั้นต้น						695	
20								

Goal Seek เป็นเครื่องมือพิสดารที่เหมาะสมในการวางแผนงบประมาณ ช่วยตั้งราคาขายสินค้าเพื่อให้ได้กำไรตามต้องการ หรือช่วยในการคำนวณย้อนกลับในงานที่คนเราคิดเองไม่ไหวได้

อย่างไม่น่าเชื่อ ซึ่งในงานวางแผนตัดสินใจนั้น แทบเป็นไปได้เลยที่เราจะมีความเชื่อมั่น 100% ต่อคำตอบที่มีอยู่เสมอไป ดังนั้นแทนที่จะเสี่ยงกับการใช้ตัวเลขเพียงตัวเดียว ขอแนะนำให้ใช้ Goal Seek คำนวณหาช่วงค่าที่เป็นไปได้ว่าอยู่ในช่วงตัวเลขระหว่างค่าใด เพื่อให้สามารถวางแผนได้อย่างยืดหยุ่นมากขึ้น เช่น เดิมเคยถามว่า ตัวเลขค่าโฆษณาในเซลล์ G18 ต้องเป็นเท่าใด จึงจะคิดเป็น 15% ของกำไรขั้นต้น ก็ให้ใช้ Goal Seek ค้นหาคำตอบที่ยืดหยุ่นกว่าเดิมว่า ตัวเลขค่าโฆษณาในเซลล์ G18 ต้องเป็นค่าระหว่างเลขใด จึงจะคิดเป็น 15% - 20% ของกำไรขั้นต้น

ข้อจำกัดของ Goal Seek และวิธีแก้ไข

ข้อจำกัดที่สำคัญของ Goal Seek เกิดจากทุกครั้งที่ต้องการคำนวณครั้งใหม่ คุณต้องเป็นผู้สั่งให้ Goal Seek ทำงานซ้ำด้วยตนเอง และ Goal Seek จะยอมให้เปลี่ยนแปลงตัวเลขตัวแปรได้เพียงเซลล์เดียว นอกจากนี้ส่วนที่ไม่ยืดหยุ่นที่สุดก็คือ คุณต้องกรอกตัวเลขที่ต้องการลงไปในช่วง To value เองเสมอ โดยไม่สามารถ link ค่าตัวเลขเป้าหมายที่ต้องการจากเซลล์เข้าไปใช้ใน Goal Seek จึงเป็นเครื่องมือที่ไม่เหมาะสำหรับผู้ที่ไม่เคยใช้ Goal Seek มาก่อน

เราสามารถแก้ไขจุดอ่อนของ Goal Seek ให้สามารถรับค่าจากเซลล์ได้โดยตรง และหาทางปรับปรุงให้ Goal Seek ทำงานหาคำตอบให้เองทันทีอย่างอัตโนมัติ โดยเริ่มจากใช้ Macro Recorder บันทึกการสั่ง Data > What-If Analysis > Goal Seek แล้วใช้งานตามตัวอย่างการคำนวณหาค่าโฆษณาในตัวอย่างที่ผ่านมา จะได้รหัส VBA ตามนี้

```
Sub SetGoal()
```

```
    Range("H18").GoalSeek Goal:=0.15, ChangingCell:=Range("G18")
```

```
End Sub
```

จากนั้นให้แก้ไขตำแหน่งอ้างอิงและค่าคงที่ในตัวรหัสให้อ้างอิงชื่อ Range Name ที่ตั้งไว้ในตารางกลายเป็นรหัสใหม่ดังนี้

```
Sub SetGoal()
```

```
    [MyTarget].GoalSeek Goal:=[TargetValue], ChangingCell:=[ChangeCell]
```

```
End Sub
```

โดยกำหนดให้ตั้งชื่อ Range Name ชื่อว่า MyTarget ให้กับเซลล์ H18 และชื่อ ChangeCell ให้กับเซลล์ G18 และเมื่อใดที่ต้องการตัวเลขตัวใหม่ ให้กรอกตัวเลขลงไปในเซลล์ที่ตั้งชื่อว่า TargetValue

หากต้องการความสะดวกมากขึ้น ควรสั่งให้ Goal Seek ทำงานเองโดยใช้ Event ก็จะได้ผลลัพธ์ทันทีโดยไม่ต้องเสียเวลามาเปิด Goal Seek ขึ้นเพื่อกรอกค่าเองอีกต่อไป เช่น หากใช้ Calculation Event ก็จะได้ผลลัพธ์ตามการคำนวณใดๆที่เกิดขึ้นในชีทนั้น

```
Private Sub Worksheet_Calculate()
```

```
    [MyTarget].GoalSeek Goal:=[TargetValue], ChangingCell:=[ChangeCell]
```

```
End Sub
```

วิธีการนี้แสดงขั้นตอนการสร้างไว้ในหลักสูตรเคล็ดลับการเพิ่มผลงาน ลดความซับซ้อนของงานด้วย Excel VBA

วิธีใช้ Data Table

ถ้าถามว่าตอนเริ่มต้นใช้สูตรคำนวณใน Excel แรกเริ่มนั้นคุณต้องรู้จักใช้อะไรให้เป็น ก็ตอบว่า ต้องรู้จักการกำหนดตำแหน่งอ้างอิงโดยใช้เครื่องหมาย \$ ควบคุมแนวของค่าที่ใช้ในการคำนวณ แล้วถ้าถามต่อไปว่า ตอนสร้างแฟ้มงานเสร็จแล้วจะต้องใช้อะไรให้เป็น คำตอบก็คือ ต้องใช้ Data Table สรุปคำตอบที่เป็นไปได้เพื่อใช้ตัดสินใจ

Data Table เป็นคำสั่งที่มีมานานแล้วแต่มักถูกละเลยไม่สนใจใช้กันนัก คงเป็นเพราะคำสั่งนี้ถูกออกแบบมาให้อยู่กับเมนู Data จึงทำให้นึกไปว่าเป็นคำสั่งที่ไม่เกี่ยวข้องกับการคำนวณ อีกทั้งตัวอย่างและคำอธิบายใน Help ก็มีแต่ให้รายละเอียดชัดเจนนัก ผู้ที่จะเข้าใจ Data Table ได้ก็ต้องลองใช้คำสั่งนี้ดูแล้วจึงจะซึ้งถึงประโยชน์ของมัน

Data Table อาศัยการออกแบบตารางแบบพิเศษที่จะช่วยให้ Excel นำค่าที่วางไว้บนหัวตาราง หรือข้างตารางด้านใดด้านหนึ่งหรือทั้งสองด้าน ส่งต่อไปแทนที่ค่าหรือสูตรที่บันทึกไว้ในเซลล์ เมื่อค่าในเซลล์เหล่านั้นเปลี่ยนไปตามคำสั่ง Data Table ก็จะทำให้เซลล์สูตรที่เกี่ยวข้อง คำนวณหาคำตอบใหม่ เพียงแต่การเปลี่ยนแปลงค่าในเซลล์และคำตอบใหม่ที่เกิดขึ้นนั้น เราจะไม่เห็นมีการเปลี่ยนแปลงใดๆเกิดขึ้นในตาราง แต่จะพบว่า Excel นำคำตอบมาสรุปไว้ในตาราง Data Table ให้เอง

สำหรับผู้ที่คุ้นเคยกับการเขียนโปรแกรมสำเร็จรูป การทำงานของ Data Table เปรียบเทียบได้กับการใช้รหัสโปรแกรมสั่งให้ Excel ทำงานซ้ำหลายครั้งเป็น loop โดย Excel จะส่งค่าไปที่เซลล์ที่เรากำหนดไว้ในแต่ละ loop ทั้งนี้ Data Table จะทำงานเป็น loop ให้อัตโนมัติโดยเราไม่ต้องเสียเวลาเขียนโปรแกรมแม้แต่หน่อย

Data Table มี 3 แบบตามจำนวนตัวแปรที่เกี่ยวข้อง ได้แก่

- Data Table แบบ 1 ตัวแปร
- Data Table แบบ 2 ตัวแปร
- Data Table แบบไม่จำกัดตัวแปร โดยใช้ร่วมกับสูตร Index (กลายเป็นวิธีที่ฉลาดกว่าการใช้ Scenarios)

โครงสร้างตาราง Data Table แบ่งเป็น 4 ส่วน ตามรูป

	A	B	C	D	E
1					
2		เซลล์หัวมุม	หัวตารางด้านบน		
3		หัวตารางด้านซ้าย			
4					
5					
6				ตารางคำตอบ	
7					
8					
9					
10					

Data Table แบบ 1 ตัวแปร

จากตัวอย่างการหาค่าโฆษณา หากกำหนดให้ค่าโฆษณาเป็นตัวเลขในช่วง 100 ถึง 400 บาท ให้หาว่าจะส่งผลต่อกำไรขั้นต้นอย่างไร

ทางออกในการแก้ปัญหานี้ หากคุณไม่รู้จัก Data Table มาก่อน ก็คงต้องสร้างตารางคำนวณซ้ำกันหลายๆตารางเพื่อทดลองใส่ตัวเลขค่าโฆษณาลงไปเพื่อหาผลลัพธ์ที่ต้องการ

แทนที่จะเสียแรงเสียเวลาสร้างตารางคำนวณซ้ำ ให้ออกแบบโครงสร้างตาราง Data Table ขึ้นมาก่อน โดยในปัญหานี้สังเกตว่าต้องการเปลี่ยนค่าโฆษณาเพียงค่าเดียว ดังนั้นจึงเข้าข่าย Data Table แบบ 1 ตัวแปร ซึ่งมีโครงสร้างแยกแยะออกไปอีก 2 แบบย่อยตามแต่ว่าการต้องการดูตัวแปรตามแนวใด ได้แก่

- Data Table แบบวางตัวแปรไว้ในหัวตารางด้านข้างตามแนวนอน ส่วนหัวตารางด้านบนเป็นสูตรคำตอบที่ link มา และเมื่อสั่ง Data Table ให้เลือกใช้ช่อง Column Input Cell
- Data Table แบบวางตัวแปรไว้ในหัวตารางด้านบนตามแนวตั้ง ส่วนหัวตารางด้านข้างเป็นสูตรคำตอบที่ link มา และเมื่อสั่ง Data Table ให้เลือกใช้ช่อง Row Input Cell

ตัวอย่างการใช้ Data Table แบบวางตัวแปรไว้ในหัวตารางด้านข้างตามแนวนอน

กำหนดให้ ต้นทุนขาย เท่ากับ 20% ของยอดขาย และค่าโฆษณา เท่ากับ 15% ของกำไรขั้นต้น

ยอดขาย 1,000
ต้นทุนขาย 200 20 %
ค่าโฆษณา 0 %
กำไรขั้นต้น 800

กำไรขั้นต้น %กำไรขั้นต้น

ค่าโฆษณา	กำไรขั้นต้น	%กำไรขั้นต้น
100	700	14%
150	650	23%
200	600	33%
300	500	60%
400	400	100%

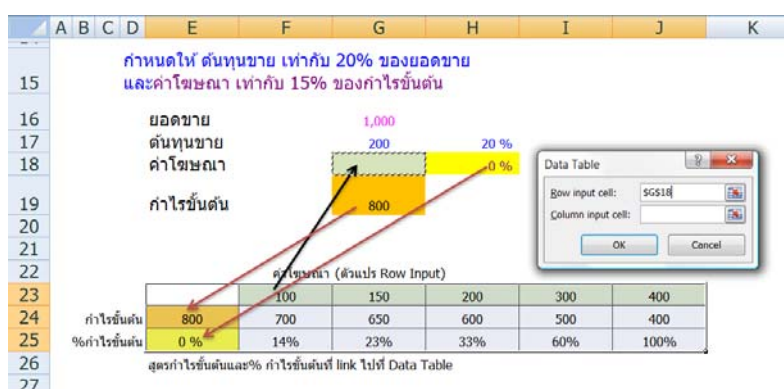
ค่าโฆษณา (ตัวแปร Column Input)

สูตรกำไรขั้นต้นและค่าโฆษณา link ไปที่ Data Table

Data Table
Row input cell: \$C\$18
Column input cell: \$C\$18

1. นำตัวเลขค่าโฆษณามาพิมพ์ลงในเซลล์ E23:E27 โดยจะกำหนดให้เป็นตัวเลขใดก่อนหลังก็ได้ ไม่จำเป็นต้องเรียงลำดับมากนัก
2. เซลล์ F22 และ G22 สร้างสูตร link ผลลัพธ์ของสูตรคำนวณมาจากเซลล์ G19 และ H18 ตามลำดับ
3. ให้เลือกพื้นที่ E22:G27 แล้วสั่ง Data > What-If Analysis > Data Table
4. คลิกลงในช่อง Column input cell แล้วคลิกต่อไปที่เซลล์ G18 (เพื่อกำหนดให้ Data Table ยิงตัวเลขค่าโฆษณาที่วางไว้ตามแนวตั้งไปที่เซลล์ G18) แล้วกดปุ่ม OK
5. จะพบคำตอบปรากฏในตาราง F23:G27 และเมื่อคลิกดูจะพบสูตร {=TABLE(,G18)} ซึ่งสูตรนี้เราไม่สามารถสร้างเอง ต้องใช้คำสั่ง Data Table เท่านั้นจึงจะเกิดสูตรนี้ขึ้น และเมื่อดูในสูตรจะพบ (,G18) แสดงถึงการไม่ใช่ Row Input แต่ใช้ Column Input มีเซลล์ G18 เป็นเซลล์รับค่า

หากใช้ตาราง Data Table แบบวางตัวแปรไว้ในหัวตารางด้านบนบนตามแนวนอน ต้องออกแบบตารางตามรูปต่อไปนี้



ข้อสังเกต

1. Data Table แบบ 1 ตัวแปร สามารถหาคำตอบมาแสดงในตารางได้หลายเรื่อง ตามแต่ที่จะ link สูตรคำตอบเรื่องใดมาแสดงในหัวตารางอีกข้างหนึ่ง จึงให้ผลลัพธ์ได้มากกว่า Data Table แบบ 2 ตัวแปร ซึ่งแม้จะใช้ตัวแปรได้ 2 ตัว แต่จะแสดงคำตอบในตารางได้เพียงเรื่องเดียวเพราะมีเซลล์หัวมุมเพียงเซลล์เดียว
2. ถ้าต้องการใช้ตัวแปรมากกว่า 2 ตัว หรือต้องการคำตอบหลายเรื่อง สามารถใช้สูตร If, Choose หรือ Index ช่วยทำให้ Data Table ทำงานได้ไม่จำกัด
3. สูตรที่ Excel สร้างให้เมื่อสร้าง Data table เป็นสูตรที่ถูกสร้างขึ้นแบบ Array พร้อมกันทุกเซลล์ โดยมีโครงสร้างสูตรดังนี้

=Table(Row Input Cell, Column Input Cell)

Data Table แบบ 1 ตัวแปร จะเกิดสูตร =Table(Row Input Cell,) เมื่อเป็นตารางที่ใช้ Row Input และจะเกิดสูตร =Table(, Column Input Cell) เมื่อเป็นตารางที่ใช้ Column Input ส่วน Data Table แบบ 2 ตัวแปรจะเกิดสูตรที่ใช้โครงสร้างแบบเต็ม =Table(Row Input Cell, Column Input Cell)

4. การอ่านผลลัพธ์จากตาราง Data Table ต้องอธิบายให้ชัดเจนว่า ในขณะที่ค่าโฆษณาเปลี่ยนแปลงไปนั้น ตัวเลขตัวแปรอื่น ได้แก่ ยอดขาย และ %ต้นทุนขาย ถือว่าคงที่
5. ค่าโฆษณาในเซลล์ G18 จะเว้นว่างไว้หรือจะใส่ตัวเลขใดๆแม้จะสร้างสูตรคำนวณไว้ก่อนหรือหลังการสั่ง Data Table ก็ได้ เพราะไม่มีผลต่อการคำนวณใน Data Table
6. ตาราง Data Table ไม่สามารถถูก Copy ออกไปใช้ที่อื่น เพราะเมื่อ Paste จะพบว่าสูตร Table ในตารางถูกเปลี่ยนเป็นตัวเลขค่าคำตอบแทนทุกเซลล์
7. เราไม่สามารถ Insertแทรกเซลล์สูตร Table และหากต้องการลบสูตร Table ทั้ง ต้องเลือกพื้นที่ตารางทั้งหมดที่มีสูตร Table แล้วจึงจะสามารถลบสูตรทิ้งพร้อมกันทุกเซลล์

Data Table แบบ 2 ตัวแปร

ในตัวอย่างเดิม หากต้องการวิเคราะห์ว่าถ้าเราเปลี่ยนแปลงต้นทุนขายและค่าโฆษณาจะมีผลอย่างไรบ้างต่อยอดกำไรขั้นต้น

ต้นทุนขาย (ตัวแปร Row Input)	ค่าโฆษณา (ตัวแปร Column Input)	ยอดขาย	กำไรขั้นต้น
800	100	200	250
100	700	650	500
150	650	600	450
200	600	550	400
300	500	450	300
400	400	350	200

1. พิมพ์ตัวแปรต้นทุนขายไว้ในเซลล์ F22:H22 ซึ่งถือเป็น Row Input
2. พิมพ์ตัวแปรต้นทุนค่าโฆษณาไว้ในเซลล์ E23:E27 ซึ่งถือเป็น Column Input

3. ในเซลล์หัวมุมตาราง E22 สร้างสูตร =G19 เพื่อ link สูตรคำนวณหากำไรขั้นต้นมาใช้
4. จากนั้นให้เลือกพื้นที่ตั้งแต่ E22:H27 แล้วสั่ง Data > What-If Analysis > Data Table
5. คลิกลงไปในช่อง Row Input Cell แล้วคลิกต่อไปยังเซลล์ G17 ซึ่งเป็นเซลล์ต้นทุนขาย
6. คลิกลงไปในช่อง Column Input Cell แล้วคลิกต่อไปยังเซลล์ G18 ซึ่งเป็นเซลล์ค่าโฆษณา แล้วกดปุ่ม OK จะพบคำตอบของกำไรขั้นต้น แสดงในตาราง F23:H27

เมื่อนำ Data Table มาเปรียบเทียบกับ Goal Seek มีข้อสังเกตดังนี้

1. Data Table สามารถส่งค่าตัวแปรใหม่ครั้งละหลายตัวไปแทนที่เซลล์ที่มีสูตรสร้างไว้ ซึ่ง Excel จะคำนวณให้ตามค่าตัวแปรใหม่นั้น โดยถือว่าไม่ได้ใช้สูตรที่สร้างไว้ก่อนแล้วในเซลล์นั้นมาใช้คำนวณร่วมด้วยแต่อย่างใด ส่วน Goal Seek สามารถส่งตัวแปรใหม่ไปได้ครั้งละ 1 ตัวและไม่สามารถส่งตัวแปรไปแทนเซลล์ที่เป็นสูตร
2. Data Table เหมาะกับการใช้วางแผนระยะยาว โดยกำหนดค่าตัวแปรบนหัวตารางให้ครอบคลุมช่วงค่าที่เป็นไปได้เพื่อดูผลลัพธ์ที่เกิดขึ้นในช่วงค่านั้นๆซึ่งอาจไม่ได้คำตอบตรงกับค่าเป้าหมายที่ต้องการ ส่วน Goal Seek เหมาะสำหรับการวิเคราะห์หาตัวเลขเฉพาะกิจ ซึ่งจะคิดย้อนกลับจนได้ตัวเลขตามต้องการ
3. เราสามารถสั่ง Formulas > Calculation Options > Automatic Except for Data Tables เพื่อกำหนดให้ใช้การกดปุ่ม F9 เพื่อสั่งให้ Data Table คำนวณเฉพาะเมื่อต้องการเท่านั้น ส่วน Goal Seek แม้ว่าจะพบว่าใช้กับการคำนวณแบบ Automatic หรือ Manual ก็ได้ แต่เราควรใช้ Goal Seek ร่วมกับการคำนวณแบบ Automatic ไว้ก่อน เพื่อช่วยทำให้มั่นใจว่าเซลล์สูตรที่เกี่ยวข้องถูกคำนวณตามกันไปครบทั้งหมดด้วย
4. เนื่องจากเราเป็นผู้กำหนดค่าตัวแปรให้ใช้กับ Data Table ดังนั้น Excel จะถูกสั่งให้คำนวณตามค่าตัวแปรเหล่านั้นจนครบทุกตัว ส่วน Goal Seek จะคำนวณย้อนกลับให้เองแต่อาจได้คำตอบเฉพาะในช่วงค่าที่ Excel ค้นหาให้เท่านั้น
5. หาก Data Table ไม่สามารถหาคำตอบที่ต้องการ ก็สามารถใช้ Goal Seek ทำงานต่อบนตาราง Data Table โดยให้ใช้ Goal Seek คิดย้อนหาค่าตัวแปรบนหัวตาราง Data Table จนกว่าจะคำนวณได้คำตอบมาปรากฏขึ้นในพื้นที่ส่วนของคำตอบในตาราง Data table

หลากหลายแนวทางการใช้สูตร IF

ในบรรดาสูตรสำเร็จรูปของ Excel นั้น สูตรที่ใช้มากที่สุดและมีประโยชน์มากที่สุดรองจากสูตร SUM เห็นจะหนีไม่พ้นสูตร IF ซึ่งแปลว่า "ถ้า" นี้แหละ โดยเฉพาะคนที่มีตำแหน่งขึ้นชื่อว่าเป็นหัวหน้า ถ้าใช้สูตร IF ไม่เป็น ก็ไม่ควรเป็นหัวหน้าหรอกนะ เปลี่ยนตำแหน่งให้ได้ชื่อตรงข้ามกับคำว่าหัวกับหน้า ให้กลายเป็นตำแหน่งเท่าหลังแทนจะเหมาะสมกว่า

ถ้าเงินเดือนข้างต้นเกี่ยวกับหัวหน้ามาเขียนด้วยสูตร IF จะได้สูตรตามนี้

=IF (หัวหน้าใช้สูตร IF เป็นหรือไม่ , ถ้าใช่เป็น ให้เป็นหัวหน้าต่อไป , ถ้าไม่ใช่เป็น ให้เป็นเท่าหลังแทน)

หรือถ้าคิดเขียนสูตร IF ตามหลักกฎแห่งกรรมของการทำดีต้องได้ดี ทำชั่วได้ชั่ว จะได้สูตรตามนี้

=IF (คุณทำดีใช่ไหม , ถ้าทำดี ต้องได้ดี , ถ้าไม่ทำดี ต้องได้รับผลอื่นๆ)

สังเกตว่าในสูตร IF ของการตรวจสอบว่าทำดีหรือไม่นั้น ถ้าไม่ได้ทำดี ไม่ใช่ว่าจะได้ชั่ว เพราะเงื่อนไขที่ใช้เป็นเพียงตรวจสอบว่าทำตัวอย่างเดียวเท่านั้น คนที่ไม่ได้ทำดี ไม่ใช่ว่าต้องทำชั่วก็ได้ อาจอยู่เฉยๆไม่ได้ทำดีไม่ได้ทำชั่ว ดังนั้นกรณีที่ไม่ได้ทำดี จึงต้องได้รับผลอย่างอื่นซึ่งไม่ใช่ผลจากการทำดี

ถ้าคุณเข้าใจวิธีคิดเงื่อนไขในใจของตัวเอง ก็จะพบว่าบริษัทไมโครซอฟท์ได้สร้าง Excel ขึ้นมาโดยใช้สูตร IF ที่มีโครงสร้างของสูตรตามหลักการเดียวกันกับที่คุณคิดได้นั่นเอง

โครงสร้างสูตร IF

=IF(การตรวจสอบเงื่อนไข, ผลลัพธ์กรณีที่ตรวจสอบเงื่อนไขว่าใช่, ผลลัพธ์กรณีที่ตรวจสอบเงื่อนไขว่าไม่ใช่)

ในส่วนของการตรวจสอบเงื่อนไข ให้ใส่สมการที่ต้องการลงไป โดยใช้เครื่องหมาย =, >, <, >=, <=, หรือไม่เท่ากับ <> หรือในส่วนของการตรวจสอบเงื่อนไขนี้ อาจใช้สูตรอื่นซึ่งคืนค่า

เป็น TRUE หรือ FALSE อย่างใดอย่างหนึ่ง เช่น AND, OR, TRUE, FALSE, ISNUMBER เป็นต้น

ตัวอย่างการใช้ IF แบบง่ายๆ

=IF(SaleChoice="n", 1000, 2000)

สูตรนี้จะคืนค่าเป็นตัวเลข 1000 ต่อเมื่อเซลล์ที่ตั้งชื่อว่า SaleChoice มีค่าเป็นตัวอักษร n เท่านั้น แต่ถ้า SaleChoice มีค่าอื่น จะคืนค่าเป็นตัวเลข 2000 แทน

เนื่องจากตัวอักษร n เป็นตัวอักษรที่ Excel ไม่รู้จัก จึงจำเป็นต้องใส่ "n" ไว้ระหว่างเครื่องหมายคำพูด ส่วนคำว่า IF และ SaleChoice นั้นแม้ว่าเป็นตัวอักษรแต่ไม่ต้องใส่ระหว่างเครื่องหมายคำพูดเพราะเป็นตัวอักษรที่ Excel รู้จักแล้วคำว่า IF คือสูตรชื่อ IF และคำว่า SaleChoice เป็น Range Name ที่ตั้งชื่อไว้ ส่วนตัวเลข 1000 และ 2000 เป็นตัวเลขที่ Excel รับทราบอยู่แล้วว่าเป็นตัวเลขจึงไม่ต้องใส่ระหว่างเครื่องหมายคำพูดแต่อย่างใด

=IF(MyMargin>0, MyMargin*0.3, 0)

สูตรนี้ใช้กับการคำนวณหายอดขาย ถ้า MyMargin หรือเซลล์กำไรขั้นต้นมีค่ามากกว่า 0 ย่อมแสดงว่ามีกำไรเกิดขึ้น จึงนำยอดกำไรขั้นต้นไปคูณด้วยอัตราภาษี 30% แต่ถ้าไม่มีกำไรก็ไม่ต้องจ่ายภาษี

แนวทางการใช้สูตร IF

1. ใช้เปลี่ยนเส้นทางการรับค่า
2. ใช้ในการตัดสินใจ
3. ใช้ในการเปลี่ยนค่าหรือแสดงค่า
4. ใช้เปลี่ยนเส้นทางการส่งค่า

การใช้สูตร IF ในการเปลี่ยนเส้นทางการรับค่า

การสร้างสูตรลิงค์เพื่อเชื่อมโยงค่าโดยทั่วไปเป็นการเชื่อมโยงกันแบบ 1 ต่อ 1 เช่น เมื่อต้องการทำให้เซลล์ A2 มีค่าเท่ากับเซลล์ A1 ก็สร้างสูตร =A1 ลงไปในเซลล์ A2 ทำให้เซลล์ A2 แสดงค่าเปลี่ยนแปลงตามค่าในเซลล์ A1 ซึ่งการลิงค์แบบนี้เซลล์ A1 เรียกว่าเป็นเซลล์ต้นทาง และเซลล์ A2 เรียกว่าเป็นเซลล์ปลายทาง

แต่ถ้าเซลล์ต้นทางมีมากกว่า 1 เซลล์ เช่นกำหนดให้เซลล์ C2 และ C3 สามารถเป็นเซลล์ต้นทางได้ทั้งคู่ โดยกำหนดให้เซลล์ E3 เป็นเซลล์ปลายทางเพียงเซลล์เดียว โดยเลือกที่จะรับค่าจากเซลล์ C2 หรือ C3 ก็ได้แล้วแต่เงื่อนไขที่เรากำหนดว่าจะให้เลือกรับค่ามาจากเซลล์ใด

E3					fx =IF(E2="A",C2,C3)	
	A	B	C	D	E	
1						
2		A	100		A	
3		B	200		100	
4						

จากภาพกำหนดให้เซลล์ E2 เป็นเซลล์สำหรับใส่ค่าเงื่อนไขลงไปว่าจะเป็นตัวอักษร A หรือ B (หรือค่าอื่นใดก็ได้) ส่วนเซลล์ E3 ซึ่งเป็นเซลล์ปลายทาง ให้สร้างสูตรต่อไปนี้

=IF(E2="A", C2, C3)

ดังนั้นเมื่อเซลล์ E2 มีค่าเป็น A จึงทำให้เซลล์ E3 รับค่า 100 มาจากเซลล์ C2 แต่ถ้าเซลล์ E2 ไม่ได้มีค่าเป็น A ก็จะทำให้เซลล์ E3 เปลี่ยนเส้นทางการรับค่า 200 มาจากเซลล์ C3 แทน

แต่ถ้ากำหนดให้มีจำนวนเซลล์ต้นทางมากมายหลายเซลล์ เช่น ตามรูปต่อไปนี้ใช้เซลล์ต้นทางถึง 14 เซลล์

F9										fx =IF(E9="A",C2,IF(E9="B",C5,IF(E9="C",C8,IF(E9="D",C11,IF(E9="E",C14,IF(E9="F",C17,IF(E9="G",F2,IF(E9="H",F17,IF(E9="I",I2,IF(E9="J",I5,IF(E9="K",I8,IF(E9="L",I11,IF(E9="M",I14,I17))))))))))))))	
	A	B	C	D	E	F	G	H	I	J	
1											
2	A	100			G	700		I	900		
3											
4	B	200						J	1000		
5											
6											
7	C	300			K	1100		K	1100		
8											
9											
10	D	400						L	1200		
11											
12	E	500						M	1300		
13											
14	F	600			H	800		N	1400		
15											
16											
17											
18											
19											

สูตรลิงค์เพื่อเลือกรับค่าในเซลล์ F9 จะกลายเป็นสูตรซับซ้อนมากขึ้น

=IF(E9="A",C2,IF(E9="B",C5,IF(E9="C",C8,IF(E9="D",C11,IF(E9="E",C14,IF(E9="F",C17,IF(E9="G",F2,IF(E9="H",F17,IF(E9="I",I2,IF(E9="J",I5,IF(E9="K",I8,IF(E9="L",I11,IF(E9="M",I14,I17))))))))))))))

ซึ่งเซลล์ E9 มีค่าเป็น K จึงทำให้สูตร IF ในเซลล์ F9 เลือกรับค่ามาจากเซลล์ I8

แต่ถ้าโครงสร้างตารางไม่ได้จัดให้เป็นระเบียบ แม้สูตร IF จะยังคงทำงานได้ตามเดิมก็ตาม แต่จะพบว่าการแกะสูตรเพื่อตรวจสอบความถูกต้องทำได้ยากขึ้นมาก

	A	B	C	D	E	F	G	H	I	J
1										
2		A	100		D	400				
3								B	200	
4										
5										
6			L	1200						
7							J	1000		
8		X	1100							
9					K	1100		E	500	
10										
11								C	300	
12			M	1300						
13										
14				H	800		G	700		
15										
16									F	600
17		N	1400			I	900			
18										
19										

สูตรในเซลล์ F9

=IF(E9="A",C2,IF(E9="B",I3,IF(E9="C",J11,IF(E9="D",F2,IF(E9="E",I9,IF(E9="F",J16,IF(E9="G",H14,IF(E9="H",E14,IF(E9="I",G17,IF(E9="J",H7,IF(E9="K",C8,IF(E9="L",D6,IF(E9="M",D12,C17))))))))))))))

แต่ช่วยตรวจสอบว่า แม้ว่า E9 มีค่าเท่ากับ K ทำให้เซลล์ F9 รับค่า 1100 มาจากเซลล์ C8 ได้ถูกต้องตามต้องการก็ตาม แต่ค่า 1100 ที่ได้รับมานั้น ไม่ใช่ตัวเลขของค่า K แต่เป็นของค่า X มิใช่หรือ

ตั้งแต่ Excel 2007 เป็นต้นมา ในวงเล็บของแต่ละสูตร เราสามารถซ้อนสูตรเข้าไปได้ถึง 64 สูตร (Excel 2003 และรุ่นก่อนนั้นจะซ้อนได้เพียง 7 สูตร) ซึ่งจากตัวอย่างข้างต้นน่าจะทำให้เราได้เรียนรู้ว่า ถึงแม้ว่าเราสามารถใส่สูตร IF ซ้อน IF เพื่อหาค่าที่กระจายกันอยู่ต่างที่ต่างชนิดต่าง

แฟ้มได้ก็ตาม แต่การกระจายกันของค่านี้เอง หากกระจายอย่างขาดระเบียบ ย่อมเพิ่มความเสี่ยงที่จะหาคำตอบผิดพลาดได้โดยไม่รู้ตัว และเมื่อจะตรวจสอบแก้ไขก็จะทำให้ทำได้ยาก

ดังนั้นโปรดจำไว้ว่า ขอให้เลือกใช้สูตร IF กับปัญหาง่ายๆ แต่เมื่อปัญหาซับซ้อนมากขึ้น เราควรปรับเปลี่ยนโครงสร้างตารางหรือเลือกใช้สูตรอื่นซึ่งทำงานได้เช่นเดียวกันกับสูตร IF จะเหมาะสมกว่า

การใช้สูตร IF ในการตัดสินใจ

โครงสร้างสูตร IF แบบง่ายๆ =IF(เงื่อนไข, ผลกรณีใช่, ผลกรณีไม่ใช่) มีพื้นฐานของตัวสูตรทำหน้าที่ตัดสินใจตามเงื่อนไขที่เรากำหนดลงไป สูตรอยู่แล้ว ไม่ว่าจะใช้สูตร IF ในการเปลี่ยนเส้นทางการรับค่าหรือใช้ในการปรับเปลี่ยนค่าก็ตาม ย่อมถือว่าเป็นการใช้สูตร IF ในการตัดสินใจนั่นเอง เพียงแต่หัวข้อนี้จะแยกแยะประเภทของการตัดสินใจให้เห็นชัดเจน

การใช้สูตร IF ในการตัดสินใจมี 2 แบบ ขึ้นกับว่าเงื่อนไขถัดไปกำหนดว่าต้องผ่านเงื่อนไขก่อนหน้าอย่างไร

1. การใช้ IF แบบกำหนดเงื่อนไขแตกต่อกรณีที่ใช่
2. การใช้ IF แบบกำหนดเงื่อนไขแตกต่อกรณีที่ใช่

การใช้ IF แบบกำหนดเงื่อนไขแตกต่อกรณีที่ใช่

เป็นเงื่อนไขที่พบเห็นกันทั่วไป เช่น สูตร IF ในตัวอย่างที่ใช้สำหรับเปลี่ยนเส้นทางการรับค่านั้นเอง เมื่อเงื่อนไขการรับค่าแรกผ่านไปแล้ว ได้คำตอบแรกตามเงื่อนไขแรกเมื่อตัดสินใจว่าใช่ไป แล้วก็จะได้คำตอบกรณีที่ใช่เรียบร้อย แต่หากเงื่อนไขแรกไม่ผ่าน ก็จะต้องพิจารณาเงื่อนไขในลำดับถัดไป ซึ่งแสดงเป็นโครงสร้างสูตร IF จากแบบง่ายๆ กลายเป็นแบบซ้อนกันดังนี้

สำหรับการตัดสินใจในเงื่อนไขแรก

=IF(เงื่อนไขที่ 1, ผลกรณีใช่ของเงื่อนไขที่ 1, ผลกรณีไม่ใช่ของเงื่อนไขที่ 1)

ถ้าเงื่อนไขที่ 1 ยังไม่ผ่าน ให้แตกผลกรณีไม่ใช่ของเงื่อนไขที่ 1 ต่อเป็น IF สูตรที่สองเพื่อคิดตามเงื่อนไขที่ 2

=IF(เงื่อนไขที่ 2, ผลกรณีใช่ของเงื่อนไขที่ 2, ผลกรณีไม่ใช่ของเงื่อนไขที่ 2)

เมื่อนำสูตรทั้งสองเงื่อนไขมาซ้อนเป็นสูตรในเซลล์เดียว จะได้สูตรดังนี้

=IF(เงื่อนไขที่ 1, ผลกรณีใช่ของเงื่อนไขที่ 1, IF(เงื่อนไขที่ 2, ผลกรณีใช่ของเงื่อนไขที่ 2, ผลกรณีไม่ใช่ของเงื่อนไขที่ 2))

ตัวอย่างการใช้ IF แบบกำหนดเงื่อนไขแตกต่อกรณีที่ไม่ใช่ในการตัดเกรด

	F3	=IF(E3>=B3, C3, IF(E3>=B4,C4,C5))					
	A	B	C	D	E	F	G
1							
2		ตารางคะแนน-เกรด			คะแนนที่ได้	เกรด	
3		90	A		75	B	
4		70	B				
5		0	C				
6							
7							

กำหนดให้เซลล์ B3:C5 เป็นตารางคะแนน-เกรด

- ถ้าสอบได้ตั้งแต่ 90 คะแนนขึ้นไปให้ได้เกรด A
- ถ้าสอบได้ตั้งแต่ 70 คะแนนขึ้นไปให้ได้เกรด B
- ถ้าสอบได้ตั้งแต่ 0 คะแนนขึ้นไปให้ได้เกรด C

สมมติว่าเซลล์ E3 มีค่าเป็น 75 เป็นคะแนนที่สอบได้ เซลล์ F3 หาคำตอบเป็นเกรด B โดยใช้สูตร IF ข้อนั้นดังนี้

=IF(E3>=B3, C3, IF(E3>=B4,C4,C5))

ซึ่งแปลสูตรออกมาตามเงื่อนไขที่กำหนดได้เป็น

=IF(คะแนนที่สอบได้ >=90, "A", IF(คะแนนที่สอบได้ >=70,"B","C"))

ข้อสังเกต

1. การเขียนสูตร IF ข้อนั้น IF มีหลักการสำคัญ คือ **ต้องจัดลำดับของเงื่อนไขที่เปิดโอกาสให้ IF ทุกตัวในสูตรมีโอกาสได้ทำงาน** ดังนั้นจึงต้องใช้เงื่อนไข คะแนนที่สอบได้ >=90 ขึ้นก่อนเป็นเงื่อนไขแรก หากเงื่อนไขนี้ไม่ผ่านจึงปล่อยให้เงื่อนไขถัดไปทำงาน (แต่ถ้าสร้างผิดโดยใช้เงื่อนไข >=70 ขึ้นก่อน จะปิดโอกาสของเงื่อนไขเกรด A เพราะ >=70 ย่อมหมายถึง >=90 อยู่แล้ว)
2. แม้ต้องการเกรด A, B, C แต่เราใช้ IF ข้อนั้นเพียง 2 IF เท่านั้น เพราะถ้าไม่ใช่ทั้งสองเงื่อนไข ก็ต้องเป็นเกรด C นั่นเอง

การใช้ IF แบบกำหนดเงื่อนไขแตกต่อกรณีที่ใช่

เป็นเงื่อนไขที่กำหนดซ้อนเข้าไปเพื่อตัดสินใจในทุกเงื่อนไขพร้อมกันไป เกิดเป็นโครงสร้างสูตรตามนี้

=IF(เงื่อนไขที่ 1, IF(เงื่อนไขที่ 2, ผลกรณีใช่ของเงื่อนไขที่ 1 และ 2, ผลกรณีไม่ใช่ของเงื่อนไขที่ 2), ผลกรณีไม่ใช่ของเงื่อนไขที่ 1)

หากผลกรณีที่ใช่ของทุกเงื่อนไขเป็นค่าเดียวกัน ส่วนผลกรณีที่ไม่ใช่ของเงื่อนไขที่ 1 และผลกรณีที่ไม่ใช่ของเงื่อนไขที่ 2 เป็นค่าเดียวกัน แทนที่จะใช้สูตร IF ซ้อนกันหลายสูตร ให้ใช้สูตร And หรือ Or มาช่วยในสูตร IF เพียงสูตรเดียวได้ดังนี้

=IF(**And(เงื่อนไขที่ 1,เงื่อนไขที่ 2)**, ผลกรณีใช่, ผลกรณีไม่ใช่)

=IF(**OR(เงื่อนไขที่ 1,เงื่อนไขที่ 2)**, ผลกรณีใช่, ผลกรณีไม่ใช่)

หมายเหตุ

1. สูตร And จะคืนค่าเป็นจริงต่อเมื่อทุกเงื่อนไขเป็นจริงพร้อมกัน หากเงื่อนไขใดเงื่อนไขหนึ่งเป็นเท็จ จะคืนค่าเป็นเท็จ (คำว่า False เป็นลักษณะเด่น)
2. สูตร OR จะคืนค่าเป็นเท็จต่อเมื่อทุกเงื่อนไขเป็นเท็จพร้อมกัน หากเงื่อนไขใดเงื่อนไขหนึ่งเป็นจริง จะคืนค่าเป็นจริง (คำว่า True เป็นลักษณะเด่น)

ตัวอย่างการใช้ IF แบบกำหนดเงื่อนไขแตกต่อกรณีที่ใช้ในการป้องกันสูตรลับ

	D2				=IF(B4=123, IF(B5=999, B2+C2, 0), 0)			
	A	B	C	D	E	F	G	H
1								
2		1	2	3				
3				3				
4	รหัส1	123			D2 : =IF(B4=123, IF(B5=999, B2+C2, 0), 0)			
5	รหัส2	999			D3 : =IF(AND(B4=123,B5=999), B2+C2, 0)			
6								

สมมติว่าเราต้องการบวกเลข 1 กับเลข 2 ในเซลล์ B2 และ C2 เข้าด้วยกัน แต่ก่อนจะยอมให้สูตร B2+C2 ทำงานได้ตามปกติ ต้องผ่านการกรอกรหัสลงไปเซลล์ B4 และ B5 เป็นเลข 123 และ 999 ตามลำดับให้ถูกต้องเรียบร้อยแล้ว

ถ้าสร้างสูตร IF ซ้อน IF จะได้สูตรตามเซลล์ D2

=IF(B4=123, IF(B5=999, B2+C2, 0), 0)

หรือใช้สูตร And มาใช้งานร่วมด้วย กลายเป็นสูตร IF เพียงสูตรเดียวดังนี้

=IF(**AND(B4=123,B5=999)**, B2+C2, 0)

ในแง่ของการคำนวณแม้สูตร IF ซ้อน IF กับสูตร IF ซ้อน And ให้คำตอบแบบเดียวกันก็ตาม แต่ในแง่ของความเป็นจริงนั้นการใช้ IF ซ้อน IF ตรวจสอบเงื่อนไขทีละขั้น เมื่อพบว่า B4=123 จริงจึงจะไปตรวจสอบ B5=999 ในลำดับต่อไป ส่วนเงื่อนไขในสูตร And ถือว่าเงื่อนไขแต่ละส่วนทำงานร่วมกันหรือจะเรียกว่าต้องตรวจเงื่อนไขให้ครบทั้งหมดก่อน

ยกตัวอย่างการตัดสินใจเกี่ยวข้องกับการสอบข้อเขียน พอสอบข้อเขียนผ่านถึงคะแนนที่กำหนดแล้วจึงจะเข้าสอบสัมภาษณ์ กรณีแบบนี้ควรใช้สูตร IF ซ้อน IF ซึ่งจะทำให้ Excel ทำงานเร็วขึ้นด้วยเพราะสูตรจะคัดกรองเฉพาะคนที่ผ่านข้อเขียนให้เหลือรายการน้อยลงก่อน

ตัวอย่างการใช้ IF แบบกำหนดเงื่อนไขแตกต่อกรณีที่ใช้ในการรับเงิน ในช่วงวันที่กำหนด

E3	=IF(Date>=From, IF(Date<=To,Amount,0), 0)									
A	B	C	D	E	F	G	H	I	J	K
1	Date									
2	From	To	Amount	1	2	3	4	5	6	7
3	2	5	100		100	100	100	100		
4	2	5	100		100	100	100	100		
5										
6	2	5	100	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
7	2	5	100	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE
8	2	5	100	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE
9										
10										
11										
12										
13										
14										
15										

E3 : =IF(Date>=From, IF(Date<=To,Amount,0), 0)

E4 : =IF(AND(Date>=From,Date<=To), Amount, 0)

E6 : =Date>=From

E7 : =Date<=To

E8 : =AND(Date>=From,Date<=To)

หากต้องการรับเงิน 100 บาท ระหว่างวันที่ 2-5 กำหนดให้ใช้ Range Name ในตารางต่อไปนี้

- B3:B8 ชื่อ From
- C3:C8 ชื่อ To
- D3:D8 ชื่อ Amount
- E2:K2 ชื่อ Date

เซลล์ E3 เป็นสูตรรับเงินในช่วงวันที่ต้องการตามนี้

=IF(Date>=From, IF(Date<=To,Amount,0), 0)

หรือ

=IF(**AND(Date>=From,Date<=To)**, Amount, 0)

ข้อสังเกต

ตัวอย่างนี้ใช้สูตร And มาช่วยเพราะต้องการให้คืนค่าเป็น True เฉพาะในช่วงวันที่ 2-5 เท่านั้น (ดูเซลล์ E8:K8)

การใช้สูตร IF ในการเปลี่ยนค่าหรือแสดงค่า

ในกรณีที่ผลลัพธ์จากการคำนวณหรือค่าที่ได้จากการลิงค์ต่อมาจากเซลล์อื่น อาจไม่ใช่ค่าที่ต้องการแสดงให้ผู้ใช้เห็นเสมอไป เราสามารถนำสูตร IF มาช่วยเปลี่ยนค่าเดิมนั้นให้กลายเป็นค่าใหม่หรือเปลี่ยนจากตัวเลขหรือค่าเตือน Error ให้กลายเป็นตัวอักษรที่สื่อความหมายได้ชัดเจนกว่าแทน

	A	B	C	D	E
1					
2		100	5	20	
3				D2 : =B2/C2	
4					
5		100	0	#DIV/0!	
6				D5 : =B5/C5	
7					
8		100	0	0	
9				D8 : =IF(C8=0, 0, B8/C8)	
10					
11		100	o	0	
12				D11 : =IF(ISERROR(B11/C11), 0, B11/C11)	
13					

โปรดดูตัวอย่างจาก Row 2 ข้างต้น ในการนำเลขมาหารกันนั้น หากเป็นตัวเลขที่นำมาหารกันได้ ก็ย่อมคืนค่าที่ต้องการออกมาได้ทันที เช่น เมื่อนำเลข 100 จากเซลล์ B2 มาหารด้วยเลข 5 จากเซลล์ C2 โดยสร้างสูตร = B2/C2 ลงไปในเซลล์ D2 จะได้คำตอบเป็นตัวเลข 20

ใน Row 5 ถ้าเปลี่ยนเลข 5 ที่เป็นตัวหาร ให้เป็นเลข 0 แทน จะพบว่าสูตร =B5/C5 ไม่สามารถคำนวณได้ โดยแสดง Error #DIV/0! ออกมาแทน (#DIV/0! ย่อมาจากคำว่า Error from Divided by Zero หรือ Error เนื่องจากการถูกหารด้วยเลข 0)

เราสามารถเปลี่ยนการแสดงผล Error ให้เป็นเลข 0 แทน โดยใช้สูตรตามตัวอย่างใน Row 8 ดังนี้

=IF(C8=0, 0, B8/C8)

หรืออ่านสูตรเป็นคำแปลว่า ถ้าเซลล์ตัวหารมีค่าเท่ากับ 0 ให้ผลลัพธ์แสดงเลข 0 แทนการแสดงผล Error แต่ถ้าตัวหารไม่ใช่เลข 0 ก็ให้คำนวณหารกันต่อไปตามปกติ

ถ้าตัวหารไม่ใช่เลข 0 แต่กลับกลายเป็นตัวอักษร o จะพบว่าสูตร IF ข้างต้นนี้คือค่าออกมาเป็น #VALUE! แทน เพราะ Excel ไม่ยอมรับการนำตัวอักษรไปหารตัวเลข

ใน Row 11 กรณีที่ตัวหารเป็นตัวอักษร o (หรือตัวอักษรอื่นใด) เราสามารถป้องกันการเกิด Error จากการหารโดยใช้สูตร IsError มาช่วยตรวจสอบการคำนวณว่าจะเกิด Error หรือไม่

=IF(**ISERROR(B11/C11)**, 0, B11/C11)

หรือตั้งแต่ Excel 2007 เป็นต้นมา มี IfError เป็นสูตรใหม่ที่สั้นลง

=IFERROR(B11/C11,0)

หมายเหตุ

1. ก่อนที่จะใช้สูตร IsError ควรหาทางตรวจสอบสูตรคำนวณที่สร้างก่อนว่าสามารถคำนวณได้คำตอบที่ต้องการหรือไม่ เพราะหากใช้สูตร IsError เข้ามาช่วยร่วมกับสูตร IF แล้ว เราจะไม่เห็น Error เกิดขึ้นอีกเลย
2. ควรเลือกนำเฉพาะบางส่วนของสูตรคำนวณจากสูตรที่จะนำมาตรวจสอบด้วยสูตร IsError โดยไม่จำเป็นต้องนำสูตรยาวๆทั้งหมดมาใส่ลงไปในวงเล็บของสูตร IsError
3. ในกรณีที่เกิด Error ขึ้น ในกรณีที่ผลลัพธ์เดิมเป็น Text ควรเปลี่ยน Error เป็นค่าเตือน หรือกรณีที่ผลลัพธ์เดิมเป็นเลข ควรเปลี่ยน Error เป็นเลข 0 แทน จากนั้นให้ใช้ Format 0;-0; ซ่อนเลข 0 ให้ดูเหมือนเป็นช่องว่าง
4. ในการเปลี่ยนการแสดงผล Error ให้แสดงเป็นช่องว่าง พยายามหลีกเลี่ยงการใช้ Null Text หรือ "" (แทนการใช้เลข 0) เพราะ "" มีสภาพเป็น Text ซึ่งหากนำค่านี้ไปบวกลบคูณหารต่อจะเกิด Error ต่อไปอีก กลายเป็นภาระให้เราต้องสร้างสูตร =IF(Cell="", "", Cell) ต่อไปอีก ซึ่งทำให้แฟ้มคำนวณช้าลงและมีขนาดใหญ่ขึ้นโดยไม่จำเป็น
5. ในกรณีที่ต้องการแสดง #N/A ให้ใช้สูตร NA()
6. นอกเหนือจากการใช้สูตร IF ช่วยในการเปลี่ยน Error ให้เป็นค่าอื่นแล้ว ในหน้ากระดาษที่ถูกพิมพ์ สามารถใช้คำสั่ง Page Setup > Sheet > Print > Cells error as เพื่อเปลี่ยนเซลล์ที่มีค่าเป็น Error ให้แสดงเป็นช่องว่าง, --, #N/A แทนได้อีกด้วย



สูตรกลุ่ม Is ที่ใช้ช่วยในการตรวจสอบ

- IsNA ตรวจสอบ Error #N/A "Not Available" ซึ่งเกิดจากสูตรกลุ่ม Lookup ในกรณีที่หาค่าที่ต้องการไม่พบ
- IsError ตรวจสอบ Error ทุกประเภท
- IsErr ตรวจสอบ Error เกือบทุกประเภท เว้น NA
- IsBlank ตรวจสอบเซลล์ว่าง

- IsText ตรวจสอบเซลล์มีค่าเป็น Text หรือค่าที่จะขีดซ้ายให้เองเมื่อพิมพ์ลงไป เช่น '123 ถือว่าเป็น Text
- IsNumber ตรวจสอบเซลล์มีค่าเป็น Number หรือค่าที่จะขีดขวาให้เองเมื่อพิมพ์ลงไป เช่น 10/4/2010 จะขีดขวาเพราะถือว่าเป็น Number

การใช้สูตร IF ในการเปลี่ยนเส้นทางการส่งค่า

ตามปกติสูตรที่สร้างลงไปในเซลล์จะทำหน้าที่คำนวณหรือรับค่าที่ลิงค์มาจากเซลล์ต้นทาง ส่วนการส่งค่าจากเซลล์ต้นทางไปยังเซลล์ปลายทางโดยไม่ต้องสร้างสูตรลงไปนั้นไม่สามารถทำได้โดยวิธีปกติทั่วไป แต่ต้องเขียนรหัส VBA เข้ามาช่วย

	Source			222	
	A	B	C	D	E
1					
2		Source	222		
3					
4					
5		Target1	-111		
6		Target2	222		
7					
8		Source	=SendData!\$C\$2		
9		Target	=IF(Source<0,Target1,Target2)		
10		Target1	=SendData!\$C\$5		
11		Target2	=SendData!\$C\$6		
12					

ตัวอย่าง กำหนดให้ส่งค่าจากเซลล์ Source ไปยังเซลล์ Target1 หรือ Target2 ตามเงื่อนไขต่อไปนี้

- หาก Source มีค่าน้อยกว่า 0 เช่นมีค่าเป็นเลข -111 ให้ส่งค่า -111 ไปยังเซลล์ Target1
- หาก Source มีค่ามากกว่าหรือเท่ากับ 0 เช่นมีค่าเป็นเลข 222 ให้ส่งค่า 222 ไปยังเซลล์ Target2

เริ่มต้นจากให้ตั้งชื่อ Source, Target1, Target2 ให้กับเซลล์ C2, C5 ,C6 ตามลำดับและตั้งชื่อ Target เป็น Formula Name มีค่าเป็นสูตร

=IF(Source<0, Target1, Target2)

จากนั้นให้สร้างชุดคำสั่งนี้ลงใน Visual Basic Editor

Sub SendData()

MyVar = [Source]

[Target] = MyVar

End Sub

1. ชุดคำสั่งนี้จะรับค่าที่เก็บไว้ในเซลล์ที่ตั้งชื่อไว้ว่า Source ไปเก็บไว้ที่ตัวแปรที่มีชื่อว่า MyVar
2. จากนั้นค่าที่เก็บไว้ใน MyVar จะถูกส่งต่อไปยัง Formula Name ที่ตั้งชื่อไว้ว่า Target
3. เนื่องจาก Target เป็นสูตร =IF(Source<0, Target1, Target2) จึงทำหน้าที่ตรวจสอบค่าที่รับมาจาก Source ก่อนแล้วจึงเลือกส่งค่าไปยังเซลล์ที่ตั้งชื่อว่า Target1 หรือ Target2 ตามเงื่อนไขที่กำหนดไว้ข้างต้นต่อไป

วิธีการใช้ VBA นี้อยู่ในหลักสูตรเคล็ดการเพิ่มผลงาน ลดความซับซ้อนของงานด้วย Excel VBA

หลบร้อนหนีสูตร IF ไปหาทางออกอื่น

สูตร IF เป็นสูตรที่มีความยืดหยุ่นอย่างมาก เราสามารถใช้ทั้งตัวเลข ตัวอักษร สูตรอื่น หรือ ตำแหน่งอ้างอิงเพื่อใช้ในการกำหนดเงื่อนไข และสามารถใช้อนุโลมเปรียบเทียบค่าได้ทั้ง เท่ากับ มากกว่า น้อยกว่า หรือไม่เท่ากับ ซึ่งใน Excel ตั้งแต่รุ่น 2007 เป็นต้นมาได้ปรับปรุงให้ สามารถนำสูตรมาซ้อนเข้าไปได้ถึง 64 สูตร ทำให้ใช้สูตร IF ได้กว้างขวางกว่า Excel 2003 หรือรุ่นเก่าก่อนนั้นที่ซ้อนสูตรได้อีกเพียง 7 สูตรเท่านั้น

ไม่ว่าจะใช้สูตร IF ให้เต็มที่ตามความสามารถของ Excel รุ่นเก่าหรือรุ่นใหม่ ถ้าเขียนสูตร IF ซ้อนกันจนกลายเป็นสูตรยาวเหยียดตามสูตรในเซลล์ D9 ต่อไปนี้ คุณคิดว่าอยากจะใช้สูตร IF ต่อไปอีกหรือ

D9		fx		=IF(F9>C9,1,IF(SUM(F9:G9)>C9,2,IF(SUM(F9:H9)>C9,3,IF(SUM(F9:I9)>C9,4,IF(SUM(F9:J9)>C9,5,IF(SUM(F9:K9)>C9																			
	E	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
6	จงคำนวณหา วันที่ใดในแต่ละเดือนซึ่งสินค้าที่เก็บไว้ใน Stock จะเริ่มไม่เพียงพอต่อการเบิกใช้																						
7	ปริมาณการเบิกในแต่ละวัน >>																						
8	Stock	Short Due	Total	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
9	500	4	600	100	200	100	200																
10	500	5	600	100	200	100	100	100															
11	500	6	600	100	200	100	0	100	100														
12																							

=IF(F9>C9,1,IF(SUM(F9:G9)>C9,2,IF(SUM(F9:H9)>C9,3,IF(SUM(F9:I9)>C9,4,IF(SUM(F9:J9)>C9,5,IF(SUM(F9:K9)>C9,6,IF(SUM(F9:L9)>C9,7,0))))))

+IF(SUM(F9:L9)>C9,0,IF(SUM(F9:M9)>C9,8,IF(SUM(F9:N9)>C9,9,IF(SUM(F9:O9)>C9,10,IF(SUM(F9:P9)>C9,11,IF(SUM(F9:Q9)>C9,12,IF(SUM(F9:R9)>C9,13,0))))))

+IF(SUM(F9:R9)>C9,0,IF(SUM(F9:S9)>C9,14,IF(SUM(F9:T9)>C9,15,IF(SUM(F9:U9)>C9,16,IF(SUM(F9:V9)>C9,17,IF(SUM(F9:W9)>C9,18,IF(SUM(F9:X9)>C9,19,0))))))

+IF(SUM(F9:X9)>C9,0,IF(SUM(F9:Y9)>C9,20,IF(SUM(F9:Z9)>C9,21,IF(SUM(F9:AA9)>C9,22,IF(SUM(F9:AB9)>C9,23,IF(SUM(F9:AC9)>C9,24,IF(SUM(F9:AD9)>C9,25,0))))))

+IF(SUM(F9:AD9)>C9,0,IF(SUM(F9:AE9)>C9,26,IF(SUM(F9:AF9)>C9,27,IF(SUM(F9:AG9)>C9,28,IF(SUM(F9:AH9)>C9,29,IF(SUM(F9:AI9)>C9,30,IF(SUM(F9:AJ9)>C9,31,0))))))))

สูตรข้างต้นเป็นสูตรคำนวณหาวันที่ใดในแต่ละเดือนซึ่งสินค้าที่เก็บไว้ใน Stock จะเริ่มไม่เพียงพอต่อการเบิกใช้ โดยต้องนำสูตร IF มาซ้อนกันเพื่อตรวจสอบยอดเบิกใช้สะสมตั้งแต่วันที่แรกไปจนถึงสิ้นเดือน เริ่มจากยอดเบิกวันแรกในเซลล์ F9 กลายเป็นยอดเบิกสะสม 2 วันจากเซลล์ F9:G9 หรือยอดเบิกสะสม 3 วันจากเซลล์ F9:H9 เรื่อยไปจนถึง F9:AJ9 ซึ่งเป็นยอดเบิกใช้สะสมถึง 31 ครั้งตามจำนวนวันในแต่ละเดือนมาเทียบกับปริมาณ Stock ในเซลล์ C9

ถ้าใช้ Excel 2003 หรือรุ่นก่อนนั้น ต้องแยกสูตรที่ซ้อนกันชุดละ 7 วัน โดยกำหนดเงื่อนไขในแต่ละชุดว่า หากไม่เป็นจริงให้คืนค่าเท่ากับ 0 โปรดสังเกตว่าในวงเล็บของ IF ที่ซ้อนกันแต่ละชุดนั้นจะลงท้ายด้วย ,0)))))) แล้วจึงนำสูตร IF ที่ซ้อนกันมาบวกเข้าด้วยกัน

โจทย์เดียวกันนี้ หากใช้ Excel 2007 เป็นต้นมา เราสามารถซ้อน IF ลงไปในวงเล็บของสูตร IF ร่วมกันได้สูงสุดถึง 64 สูตรได้เลย ทำให้สูตรสั้นลงบ้างเหลือสูตรตามนี้

=IF(F9>C9,1,IF(SUM(F9:G9)>C9,2,IF(SUM(F9:H9)>C9,3,IF(SUM(F9:I9)>C9,4,IF(SUM(F9:J9)>C9,5,IF(SUM(F9:K9)>C9,6,IF(SUM(F9:L9)>C9,7,IF(SUM(F9:M9)>C9,8,IF(SUM(F9:N9)>C9,9,IF(SUM(F9:O9)>C9,10,IF(SUM(F9:P9)>C9,11,IF(SUM(F9:Q9)>C9,12,IF(SUM(F9:R9)>C9,13,IF(SUM(F9:S9)>C9,14,IF(SUM(F9:T9)>C9,15,IF(SUM(F9:U9)>C9,16,IF(SUM(F9:V9)>C9,17,IF(SUM(F9:W9)>C9,18,IF(SUM(F9:X9)>C9,19,IF(SUM(F9:Y9)>C9,20,IF(SUM(F9:Z9)>C9,21,IF(SUM(F9:AA9)>C9,22,IF(SUM(F9:AB9)>C9,23,IF(SUM(F9:AC9)>C9,24,IF(SUM(F9:AD9)>C9,25,IF(SUM(F9:AE9)>C9,26,IF(SUM(F9:AF9)>C9,27,IF(SUM(F9:AG9)>C9,28,IF(SUM(F9:AH9)>C9,29,IF(SUM(F9:AI9)>C9,30,IF(SUM(F9:AJ9)>C9,31,0))))))))))))))))))))))))))))))))))))

แม้สูตร IF สามารถคำนวณหาคำตอบได้ตามต้องการ แต่จากโครงสร้างยาวเหยียดนอกจากจะทำให้แฟ้มมีขนาดใหญ่ขึ้นแล้ว ยังเพิ่มความเสี่ยงในการติดตามแก้ไขหรือตรวจสอบอีกด้วย ดังนั้นแทนที่จะใช้สูตร IF เราสามารถใช้สูตรต่อไปนี้แทน

=(F9<=C9)+(SUM(F9:G9)<=C9)+(SUM(F9:H9)<=C9)+(SUM(F9:I9)<=C9)+(SUM(F9:J9)<=C9)+(SUM(F9:K9)<=C9)+(SUM(F9:L9)<=C9)+(SUM(F9:M9)<=C9)+(SUM(F9:N9)<=C9)+(SUM(F9:O9)<=C9)+(SUM(F9:P9)<=C9)+(SUM(F9:Q9)<=C9)+(SUM(F9:R9)<=C9)+(SUM(F9:S9)<=C9)+(SUM(F9:T9)<=C9)+(SUM(F9:U9)<=C9)+(SUM(F9:V9)<=C9)+(SUM(F9:W9)<=C9)+(SUM(F9:X9)<=C9)+(SUM(F9:Y9)<=C9)+(SUM(F9:Z9)<=C9)+(SUM(F9:AA9)<=C9)+(SUM(F9:AB9)<=C9)+(SUM(F9:AC9)<=C9)+(SUM(F9:AD9)<=C9)+(SUM(F9:AE9)<=C9)+(SUM(F9:AF9)<=C9)+(SUM(F9:AG9)<=C9)+(SUM(F9:AH9)<=C9)+(SUM(F9:AI9)<=C9)+(SUM(F9:AJ9)<=C9)

F9:R9)<=C9)+(SUM(F9:S9)<=C9)+(SUM(F9:T9)<=C9)+(SUM(F9:U9)<=C9)+(SUM(F9:V9)<=C9)+(SUM(F9:W9)<=C9)+(SUM(F9:X9)<=C9)+(SUM(F9:Y9)<=C9)+(SUM(F9:Z9)<=C9)+(SUM(F9:AA9)<=C9)+(SUM(F9:AB9)<=C9)+(SUM(F9:AC9)<=C9)+(SUM(F9:AD9)<=C9)+(SUM(F9:AE9)<=C9)+(SUM(F9:AF9)<=C9)+(SUM(F9:AG9)<=C9)+(SUM(F9:AH9)<=C9)+(SUM(F9:AI9)<=C9)+(SUM(F9:AJ9)<=C9)+1

ถ้ายังยาวเกินไปอีก ก็ต้องหันมาใช้สูตร Array กันตามนี้

{=SUM((SUBTOTAL(9,INDIRECT(ADDRESS(ROW(F9),COLUMN(F9))&":"&ADDRESS(ROW(F9),ROW(INDIRECT(COLUMN(F9)&":"&COLUMN(AJ9))))))<=C9)*1)+1}

หนังสือ IF ไปใช้สูตร Choose หรือ Index

Choose เป็นสูตรซึ่งทำหน้าที่เลือกค่าที่ใส่ไว้ในวงเล็บของตัวสูตร Choose เองตามเลขลำดับที่เรากำหนด โดยใช้เลขลำดับตั้งแต่เลข 1 - 254 ตามโครงสร้างสูตร ดังนี้

=CHOOSE(เลขลำดับ, ค่าที่1, ค่าที่2, ค่าที่ 254)

สูตร Choose กับสูตร IF มีความเหมือนกันตรงที่ 2 สูตรนี้เหมาะกับการหาคำตอบจากค่าที่กระจายตัวโดยไม่จำเป็นต้องอยู่ในตารางที่เป็นพื้นที่ต่อเนื่องกันไป เพียงแต่สูตร Choose ต้องใช้ตัวเลขลำดับเป็นเลขจำนวนเต็มตั้งแต่เลข 1 - 254 เป็นเงื่อนไขในการเลือกคำตอบ ส่วนสูตร IF สามารถรับเงื่อนไขเปรียบเทียบได้ทั้งตัวเลขและตัวอักษรและไม่จำกัดว่าต้องใช้เงื่อนไขกรณีเท่ากันเพียงอย่างเดียวเช่นที่สูตร Choose กำหนดไว้

สมมติว่าสูตร IF ที่ใช้อยู่เป็นสูตรตามนี้

=IF(E9="A",C2,IF(E9="B",C5,IF(E9="C",C8,IF(E9="D",C11,IF(E9="E",C14,IF(E9="F",C17,IF(E9="G",F2,IF(E9="H",F17,IF(E9="I",I2,IF(E9="J",I5,IF(E9="K",I8,IF(E9="L",I11,IF(E9="M",I14,I17))))))))))))))

สูตรข้างต้นใช้ค่าในเซลล์ E9 เป็นเงื่อนไขในการเปรียบเทียบกับตัวอักษร A - M เช่น ถ้าเซลล์ E9 มีค่าเป็นตัวอักษร A จะคืนค่าเป็นค่าจากเซลล์ C2 แต่ถ้าเซลล์ E9 มีค่าเป็นตัวอักษร B จะคืนค่าเป็นค่าจากเซลล์ C5 เป็นต้น

เนื่องจากเงื่อนไขที่ใช้ในสูตร IF ใช้เทียบกับตัวอักษร จึงไม่มีทางที่จะนำสูตร Choose มาใช้แทน แต่หากเราสามารถเปลี่ยนเงื่อนไขจากตัวอักษรที่ใช้เดิม หันมาใช้เปรียบเทียบกับตัวเลขแทน กลายเป็นสูตรตามนี้

=IF(E9=1,C2,IF(E9=2,C5,IF(E9=3,C8,IF(E9=4,C11,IF(E9=5,C14,IF(E9=6,C17,IF(E9=7,F2,IF(E9=8,F17,IF(E9=9,I2,IF(E9=10,I5,IF(E9=11,I8,IF(E9=12,I11,IF(E9=13,I14,I17)))))))))))))

สูตร IF ที่แก้ไขใหม่นี้ ใช้ค่าในเซลล์ E9 เป็นเงื่อนไขในการเปรียบเทียบกับตัวเลข 1 - 13 เช่น ถ้าเซลล์ E9 มีค่าเป็นตัวเลข 1 จะคืนค่าเป็นค่าจากเซลล์ C2 แต่ถ้าเซลล์ E9 มีค่าเป็นตัวเลข 2 จะคืนค่าเป็นค่าจากเซลล์ C5 เป็นต้น

เนื่องจากสูตร IF ใช้เงื่อนไขเทียบกับตัวเลข และเป็นตัวเลขจำนวนเต็มตั้งแต่เลข 1 - 13 ซึ่งไม่เกิน 254 และเป็นเงื่อนไขที่ใช้การตรวจสอบว่าเท่ากับหรือไม่เท่ากัน (ไม่ได้ตรวจสอบว่ามากกว่าน้อยกว่า) จึงสามารถใช้สูตร Choose ที่สั้นกว่าแทนได้ด้วยตามนี้

=CHOOSE(E9, C2,C5,C8,C11,C14,C17,F2,F17,I2,I5,I8,I11,I14,I17)

ถ้าเซลล์ E9 มีค่าเป็นตัวเลข 1 จะคืนค่าเป็นค่าจากเซลล์ C2 แต่ถ้าเซลล์ E9 มีค่าเป็นตัวเลข 2 จะคืนค่าเป็นค่าจากเซลล์ C5 แต่ถ้า E9 มีค่าเป็นตัวเลข 14 จะคืนค่าเป็นค่าจากเซลล์ I17

สูตร Choose สามารถใช้เลือกค่าที่กระจาย ซึ่งการกระจายตัวนี้เองที่จะกลายเป็นหนามยอกอก หากมีการกระจายตัวกันมากเกินไปเช่นสูตรต่อไปนี้

=CHOOSE(E9, C2,C5,C8,C11,C14,C17,F2,F17,I2,I5,I8,
I11,I14,I17,D3,D6,D9,D12,D15,D18,G3,G18,J3,J6,J9,
J12,J15,J18,E4,E7,E10,E13,E16,E19,H4,H19,K4,K7,
K10,K13,K16,K19)

เพียงเห็นสูตรแวนเดียวก็รู้สึกเหนื่อยแล้วใช่ไหม แม้ว่าสูตร Choose จะสามารถใช้เลือกค่าที่กระจายตัวกันอยู่ได้สูงสุดถึง 254 ค่าก็ตาม หากเราเผลอจัดลำดับค่าคำตอบสลับที่กันเพียงค่าเดียว จะส่งผลทำให้สูตร Choose คืนค่าคำตอบผิดพลาดทันที

ถ้าค่าที่ต้องการเลือกนั้นถูกนำมาจัดลำดับใหม่ไว้ในตารางให้เป็นพื้นที่ซึ่งต่อเนื่องกันไป เราสามารถนำสูตร Index มาใช้แทน Choose เพราะสูตร Index มีโครงสร้างสูตรดังนี้

=INDEX(พื้นที่ตารางที่เป็น Single Row, เลขลำดับ)

หรือ

=INDEX(พื้นที่ตารางที่เป็น Single Column, เลขลำดับ)

Single Row หรือ Single Column หมายถึงพื้นที่ตารางที่มีความสูงหรือความกว้างเป็นเซลล์เดียวตามแนวนอนหรือแนวตั้ง

ดังนั้นแทนที่จะใช้สูตร IF หรือสูตร Choose ตามตัวอย่างที่กล่าวไปแล้วนั้น หากเราสามารถจัดโครงสร้างตารางใหม่ให้อยู่ในแนวนอนหรือแนวตั้ง และใช้ตัวเลขจำนวนเต็มตั้งแต่เลข 1 เป็นต้นไปเป็นตัวชี้ตำแหน่ง ก็ควรหันมาใช้สูตร Index แทนดีกว่า กลายเป็นสูตรดังนี้

=INDEX(A1:A10000, E9)

ถ้าเซลล์ E9 มีค่าเป็นตัวเลข 1 จะคืนค่าเป็นค่าจากเซลล์ A1 แต่ถ้าเซลล์ E9 มีค่าเป็นตัวเลข 2 จะคืนค่าเป็นค่าจากเซลล์ A2 แต่ถ้า E9 มีค่าเป็นตัวเลข 10000 จะคืนค่าเป็นค่าจากเซลล์ A10000 ซึ่งการนำค่าที่ต้องการมาจัดไว้ในพื้นที่เซลล์ A1:A10000 ทำให้ง่ายในการค้นหาตำแหน่งเซลล์เพื่อบันทึกหรือเปลี่ยนแปลงค่าใหม่ในอนาคต

การเลือกใช้สูตรได้อย่างเหมาะสม สูตรนั้นต้องสามารถรองรับกับเงื่อนไขที่อาจเปลี่ยนแปลงได้ในอนาคตโดยเราไม่ต้องย้อนกลับมาแก้ไขสูตรเก่าที่เคยสร้างไว้อีกเลย ดังนั้นจะเลือกใช้สูตร IF หรือ Choose หรือ Index ก็ต้องคิดพิจารณาให้เหมาะสมด้วยว่าโครงสร้างตารางและค่าที่ใช้เป็นเงื่อนไขนั้นเป็นอย่างไร

แทนที่จะปล่อยให้บันทึกค่าอย่างตามใจหรือไม่ได้ใส่ใจว่าจะใช้ตำแหน่งเซลล์ที่ใดในการบันทึกค่า หากรู้จักออกแบบตารางให้เหมาะสมจะช่วยให้เราสามารถเลือกใช้สูตรได้ยืดหยุ่นมากขึ้น

หนังสือ IF ไปใช้สูตร VLookup

VLookup ย่อมาจากคำว่า Vertical Lookup ทำหน้าที่ตรงตามคำแปลที่หมายถึงการมองตามแนวตั้ง โดยเริ่มจากใช้ค่าที่ต้องการค้นหา นำไปเปรียบเทียบกับค่าที่เก็บไว้ในตารางจาก Column ซ้ายสุดว่า ค่าที่ใช้ค้นหานั้นอยู่ในแนวของค่าใดใน Column ซ้ายสุดนั้น แล้วคืนค่าเป็นคำตอบจาก Column ตามเลขที่เราต้องการ โดย VLookup มีโครงสร้างสูตร 2 แบบ

โครงสร้างสูตรแบบ Approaching Match

เหมาะกับตารางข้อมูลที่ Column ซ้ายสุดเป็นตัวเลขต่อเนื่องที่เรียงค่าจากน้อยไปมาก
โครงสร้างนี้ทำงานตามเงื่อนไขแบบน้อยกว่าหรือเท่ากับ

=VLOOKUP(ค่าที่ใช้หา, ตารางที่เก็บค่า, เลขที่ Column ของคำตอบ)

โครงสร้างสูตรแบบ Exact Match

เหมาะกับตารางข้อมูลที่ Column ซ้ายสุดเป็นรหัส ซึ่งไม่จำเป็นต้องเรียงค่า โครงสร้างนี้ทำงานแบบเงื่อนไขเท่ากับ

=VLOOKUP(ค่าที่ใช้หา, ตารางที่เก็บค่า, เลขที่ Column ของคำตอบ,
FALSE หรือ 0)

ในกรณีที่ใช้สูตร IF หาคำตอบเป็นเกรด A B หรือ C โดยใช้สูตร =IF(E3>=B3, C3, IF(E3>=B4,C4,C5))

	F3						
	A	B	C	D	E	F	G
1							
2							
3		ตารางคะแนน-เกรด			คะแนนที่ได้	เกรด	
4		90	A		75	B	
5		70	B				
6		0	C				
7							

หากต้องการเปลี่ยนมาใช้สูตร VLookup ต้องเรียงคะแนนในตารางจากน้อยไปมาก ตามรูปต่อไปนี้นี้แทน

	F3						
	A	B	C	D	E	F	
1							
2		ตารางคะแนน-เกรด			คะแนนที่ได้	เกรด	
3		0	C		75	B	
4		70	B				
5		90	A				
6							
7							

เซลล์ F3 สร้างสูตร =VLOOKUP(E3,B3:C5,2) ซึ่งเป็นสูตร VLookup แบบ Approaching Match เนื่องจากค่าที่ใช้หาเป็นตัวเลข และตัวเลขนี้สามารถเป็นตัวเลขที่ต่อเนื่องกันไป และคืนค่าได้คำตอบเป็นเกรด B โดยมีลำดับการทำงานของสูตร ดังนี้

1. VLookup นำคะแนนที่สอบได้ 75 คะแนน ไปตรวจสอบกับตารางข้อมูล B3:C5
2. ใน Column ซ้ายสุดที่เรียงเลขจากน้อยไปมากนั้น พบว่าเลขคะแนน 70 เป็นเลขที่มากที่สุดที่ยังน้อยกว่าหรือเท่ากับเลข 75 ซึ่งเป็นคะแนนที่สอบได้ แสดงว่าคำตอบที่ต้องการอยู่ในแถวเดียวกับเลขคะแนน 70 นั่นเอง
3. เลข 2 ที่ใส่ลงไปในวงเล็บของสูตร VLookup เป็นตัวเลขแทน Column ที่สองในพื้นที่ตาราง B3:C5 ซึ่งเป็น Column เรื่องเกรดที่เป็นคำตอบ จึงคืนค่าออกมาเป็นเกรด B

หากเลือกใช้สูตรแบบ Exact Match =VLOOKUP(E3,B3:C5,2,0) ซึ่งเพิ่มเลข 0 ต่อท้ายลงไป ในวงเล็บ จะได้คำตอบเป็น #N/A เนื่องจาก Column ซ้ายสุดที่เป็นตัวเลขคะแนน ไม่มีเลข 75 แสดงไว้ จึงคืนค่าคำตอบออกมาว่า Not Available แสดงว่าหาค่าที่ต้องการไม่พบ

แม้การใช้สูตร VLookup จะช่วยให้ได้สูตรที่สั้นลงก็ตาม แต่จะทำให้ Excel คำนวณช้ากว่าในกรณีที่ใช้สูตร IF หากตารางที่ใช้เก็บข้อมูลมีขนาดใหญ่ขึ้น ก็จะส่งผลให้สูตร VLookup คำนวณช้าลง เราจึงต้องพึงระวังไว้เสมอว่าทุกอย่างมีทั้งข้อดีข้อเสีย ได้อย่างไรก็ต้องเสียอย่างเช่นกัน

ห้สูตร IF ไปใช้สูตร Max Min หรือแค่ใช้วงเล็บช่วย

ในบรรดาเงื่อนไขที่ใช้กันมากที่สุดในชีวิตประจำวันเห็นจะหนีไม่พ้นเงื่อนไขเกี่ยวกับการตัดสินใจในเรื่องห้ามเกินกว่าหรือห้ามต่ำกว่า ยกตัวอย่างที่ใกล้ตัว เช่น เกณฑ์การคำนวณภาษีเงินได้บุคคลธรรมดา มีหลักคำนวณภาษีจากเงินได้ว่า เงินได้ช่วงที่ไม่เกินกว่ายอดนั้นยอดนี้ให้ใช้อัตราภาษีเท่านั้นเท่านี้ หรือภาษีเงินได้ของบริษัทต้องคำนวณจากยอดกำไร แต่ถ้าขาดทุนก็ไม่จำเป็นต้องจ่ายภาษี เทียบได้กับเงื่อนไขที่ห้ามคิดภาษีถ้ามีตัวเลขต่ำกว่า 0 นั่นเอง

ถ้ามีตัวเลขใดๆที่อาจเป็นไปได้ทั้งค่าบวกลบและศูนย์ แล้วกำหนดให้ใช้สูตร IF ปรับค่านี้ให้เป็นค่าใหม่ที่ห้ามต่ำกว่า 0 แต่ถ้าตัวเลขนั้นมีความมากกว่า 0 อยู่แล้ว ก็ขอให้คงค่าเดิม จะต้องสร้างสูตรตามนี้

=IF(เซลล์ตัวเลข<0, 0, เซลล์ตัวเลข)

หรือ

=IF(เซลล์ตัวเลข>=0, เซลล์ตัวเลข, 0)

แทนที่จะใช้สูตร IF ในเงื่อนไขที่เกี่ยวข้องกับข้อกำหนดห้ามต่ำกว่าหรือห้ามเกินกว่า เราสามารถนำสูตร Max หรือ Min มาใช้แทนโดยใช้หลักช่วยจำ ดังนี้

- สูตร Max หรือ Min นี้ทำงานตรงข้ามกับความเข้าใจ
- ปกติสูตร Max ทำหน้าที่หาค่าสูงสุด แต่ให้ใช้กับเงื่อนไขห้ามต่ำกว่า โดยใช้สูตร
=MAX(ตัวเลขตามเกณฑ์ต่ำสุดที่เป็นไปได้, ตัวเลขที่ต้องการเทียบ)
- ปกติสูตร Min ทำหน้าที่หาค่าต่ำสุด แต่ให้ใช้กับเงื่อนไขห้ามเกินกว่า โดยใช้สูตร
=MIN(ตัวเลขตามเกณฑ์สูงสุดที่เป็นไปได้, ตัวเลขที่ต้องการเทียบ)

ตามตัวอย่างข้างต้นที่ใช้สูตร IF ปรับตัวเลขไม่ให้ต่ำกว่า 0 นั้น สามารถเปลี่ยนมาใช้สูตร Max ต่อไปนี้แทน

=MAX(0, เซลล์ตัวเลข)

ส่วนในกรณีที่ต้องการสร้างสูตรควบคุมตัวเลขไม่ให้เกินเลข 100 สามารถใช้สูตร IF หรือ Min ได้ตามนี้

=IF(เซลล์ตัวเลข>100, 100, เซลล์ตัวเลข)

หรือ

=IF(เซลล์ตัวเลข<=100, เซลล์ตัวเลข, 100)

หรือ

=MIN(100, เซลล์ตัวเลข)

นอกจากนี้เรายังสามารถใช้วงเล็บช่วยในการตัดสินใจโดยไม่จำเป็นต้องพึ่งพาสูตรใดๆ เช่น กรณีที่ต้องการเปลี่ยนตัวเลขให้ไม่ต่ำกว่า 0 ตามตัวอย่างข้างต้น ให้ใช้สูตรที่ใช้วงเล็บช่วยต่อไปนี้ได้เลย

= (เซลล์ตัวเลข<=0)*0 + (เซลล์ตัวเลข>0)*เซลล์ตัวเลข

การใช้วงเล็บช่วยในการตัดสินใจ มีหลักการดังนี้

1. สามารถใช้ได้เฉพาะเมื่อผลลัพธ์ที่ต้องการเป็นตัวเลขเท่านั้น เนื่องจากต้องนำตัวเลขมาคำนวณต่อกันไปเรื่อยๆ
2. หากมีหลายวงเล็บ ต้องมีเงื่อนไขเดียวในวงเล็บหนึ่งเท่านั้นที่เป็นจริง และเงื่อนไขในวงเล็บอื่นต้องเป็นเท็จทั้งหมด
3. เงื่อนไขที่ใช้ตรวจสอบนั้น ถ้าคืนค่าเป็นเท็จ ถือว่ามีค่าเท่ากับเลข 0 แต่ถ้าคืนค่าเป็นจริง ถือว่ามีค่าเท่ากับเลข 1
4. ให้นำเงื่อนไขมาบวกกันในกรณีที่กำหนดให้เงื่อนไขเดียวเท่านั้นเป็นจริง หรือนำเงื่อนไขมาคูณกันในกรณีที่ต้องการตรวจสอบว่าทุกเงื่อนไขเป็นจริงพร้อมกัน

จากตัวอย่างข้างต้นในกรณีที่ห้ามต่ำกว่า 0 เช่น เซลล์ตัวเลขมีค่าเท่ากับ -5 เมื่อนำมาแทนค่าลงไปในสูตร = (เซลล์ตัวเลข<=0)*0 + (เซลล์ตัวเลข>0)*เซลล์ตัวเลข แล้ว Excel จะคิดทีละขั้นตามนี้

1. แทนค่าตัวเลขลงไปในสูตรเป็น $=(-5 \leq 0) * 0 + (-5 > 0) * -5$
2. วงเล็บของ $(-5 \leq 0)$ จะคืนค่า TRUE หรือเทียบเท่ากับ 1
3. วงเล็บของ $(-5 > 0)$ จะคืนค่า FALSE หรือเทียบเท่ากับ 0
4. จะได้สูตร $= 1 * 0 + 0 * -5$
5. จะได้สูตร $= 0 + 0$
6. ซึ่งปรับค่า -5 ออกมาเป็น 0

แต่ถ้าเซลล์ตัวเลขมีค่าเท่ากับ 5 เมื่อนำมาแทนค่าลงไปในสูตร $=(\text{เซลล์ตัวเลข} \leq 0) * 0 + (\text{เซลล์ตัวเลข} > 0) * \text{เซลล์ตัวเลข}$ แล้ว Excel จะคิดทีละขั้นตามนี้

1. แทนค่าตัวเลขลงไปในสูตรเป็น $=(5 \leq 0) * 0 + (5 > 0) * 5$
2. วงเล็บของ $(5 \leq 0)$ จะคืนค่า FALSE หรือเทียบเท่ากับ 0
3. วงเล็บของ $(5 > 0)$ จะคืนค่า TRUE หรือเทียบเท่ากับ 1
4. จะได้สูตร $=0 * 0 + 1 * 5$
5. จะได้สูตร $=0 + 5$
6. ซึ่งปรับค่า 5 ออกมาเป็น 5 เท่ากับค่าบวกตามค่าเดิมนั่นเอง

แม้การใช้วงเล็บช่วยในการตัดสินใจจะเกิดสูตรที่มีความยาวซึ่งดูเหมือนไม่จำเป็นก็ตาม แต่โครงสร้างสูตรภายในวงเล็บทุกวงเล็บมีความคล้ายคลึงกัน จึงสามารถตรวจสอบแก้ไขได้ง่าย และยังเป็นพื้นฐานของการคำนวณแบบ Array อีกด้วย

หนังสือ IF ไปใช้คำสั่ง Format

สมัยที่ผมใช้ Excel รุ่นแรกๆ Excel ไม่ได้มีคำสั่งบนเมนูให้เลือกใช้ได้มากมายอย่างเช่นทุกวันนี้ แต่ก็ได้ฟังก์ชัน IF นี้แหละช่วยในการตรวจสอบค่าต่างๆ อย่างเช่นถ้าต้องการตรวจสอบค่าที่พิมพ์ลงไปว่าเป็นตัวเลขอย่างเดียวเท่านั้น ก็ต้องใช้สูตร IF มาผสมกับสูตร IsNumber เพื่อตรวจสอบค่าที่พิมพ์ว่า ถ้าไม่ใช่ตัวเลข ให้แสดงค่าเตือนออกมาว่า Input is Wrong ตามสูตรต่อไปนี้

=IF(ISNUMBER(Cell), "OK", "Input is Wrong")

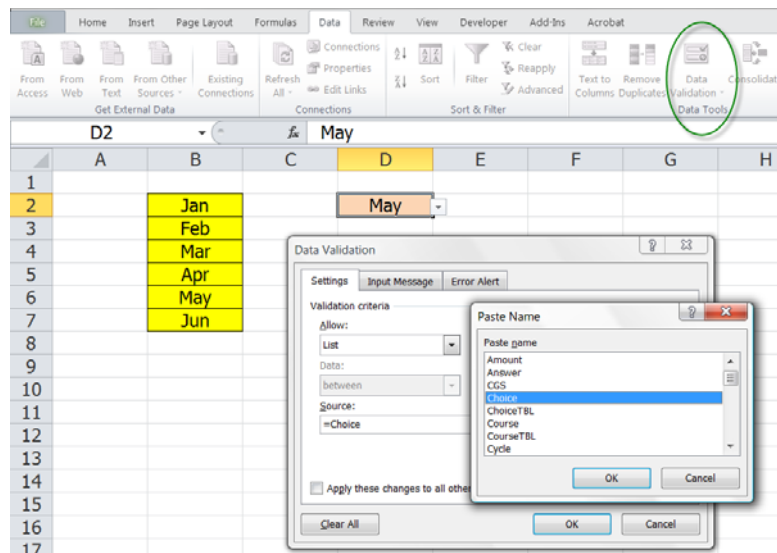
การใช้สูตรดังกล่าวจำเป็นต้องสร้างสูตรลงไปอีกเซลล์หนึ่งเพื่อเป็นเซลล์ที่แสดงค่าเตือนออกมาให้เห็น นอกจากสูตร IF ก็มีแต่รูปแบบจากคำสั่ง Format เป็นอีกเครื่องมือหนึ่งที่กำหนดลงไปในเซลล์ เช่น ถ้าหากต้องการเตือนว่า Input is Wrong ในเซลล์ที่ควบคุมว่าต้องเป็นตัวเลขเท่านั้น ก็สามารถกำหนด Format ตามนี้

0;-0;0;"Input is Wrong"

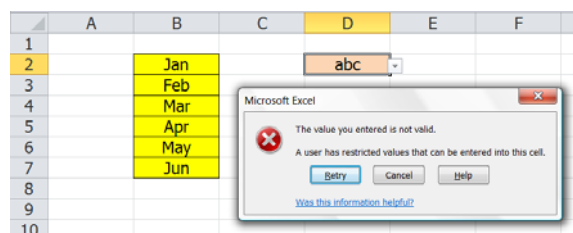
แม้ Format จะแสดงคำว่า Input is Wrong ให้เห็นในเซลล์ที่บันทึกค่าลงไปก็ตาม แต่ก็ถือว่าได้เกิดค่าที่ผิดพลาดขึ้นมาแล้ว ต่างจาก Excel รุ่นใหม่ที่มีคำสั่งบนเมนู Data Validation กับ Conditional Formatting ให้เลือกใช้ ซึ่งสามารถแสดงค่าเตือนหรือเปลี่ยนสีให้เห็นชัดในเซลล์นั้นๆเลย

วิธีใช้ Data Validation

สมมติว่าต้องการพิมพ์ชื่อเดือนลงไปในเซลล์ ซึ่งที่ผ่านมามีเคยพิมพ์ผิดซ้ำแล้วซ้ำอีก จึงต้องการ
สั่งให้ Excel แสดงค่าเตือนขึ้นมาเมื่อพิมพ์ชื่อเดือนผิด

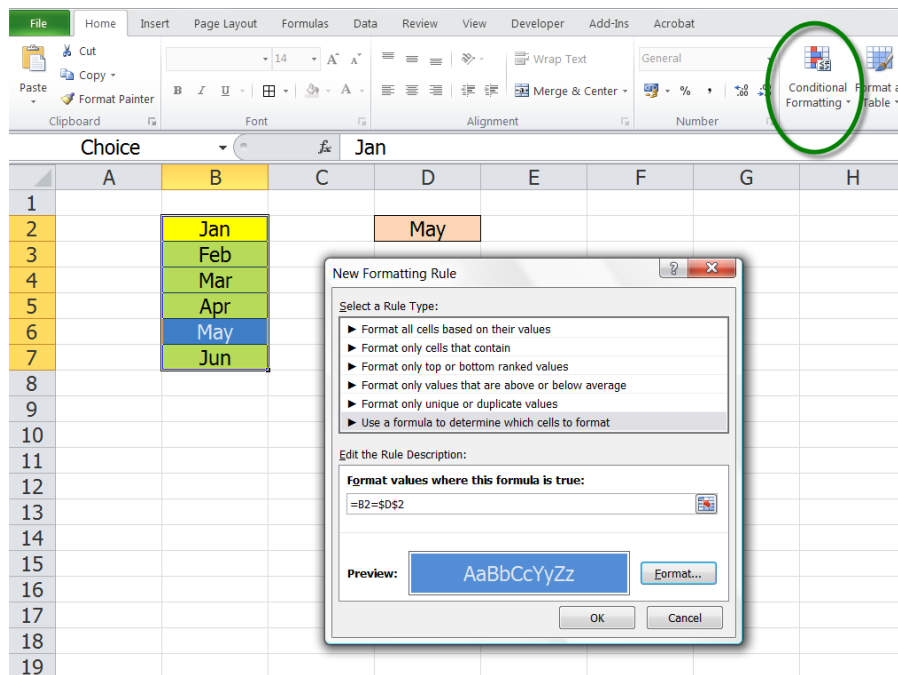


1. ตั้งชื่อ Range Name ให้กับเซลล์ที่เก็บชื่อเดือนในเซลล์ B2:B7 ให้มีชื่อว่า **Choice**
2. เลือกเซลล์ D2 ที่เป็นเซลล์ซึ่งต้องการพิมพ์ชื่อเดือนที่ต้องการแล้วสั่ง **Data > Data Validation**
3. ในช่อง **Allow** คลิกเลือก **List** (ในช่อง Allow นี้เองที่เราสามารถเลือกประเภทของข้อมูลที่ยอมรับได้อีกหลายอย่างตามแต่จะกำหนด หรือเลือก Allow แบบ Custom เพื่อใช้สูตรตรวจสอบการบันทึกค่า)
4. ให้คลิกลงไปในช่วง Source แล้วกดปุ่ม F3 เพื่อเปิดรายชื่อ Range Name ที่มีอยู่ขึ้นมา
5. ให้เลือกชื่อ Choice เพื่อเลือกแล้วกดปุ่ม OK 2 ครั้ง
6. จะพบว่าเซลล์ D2 มีปุ่มลูกศรแสดงขึ้นมาทางขอบด้านขวา และเมื่อคลิกปุ่มลูกศรนี้ก็จะมีรายชื่อเดือนแสดงให้คลิกเลือกก็ได้ หรือหากทดลองพิมพ์ชื่อเดือนลงไปในเซลล์ D2 เอง ถ้าเราพิมพ์ชื่อเดือนผิด พอกดปุ่ม Enter เพื่อรับชื่อเดือนลงไป จะมีค่าเตือนแสดงขึ้นมาให้เห็นว่า The value you entered is not valid.



วิธีใช้ Conditional Formatting

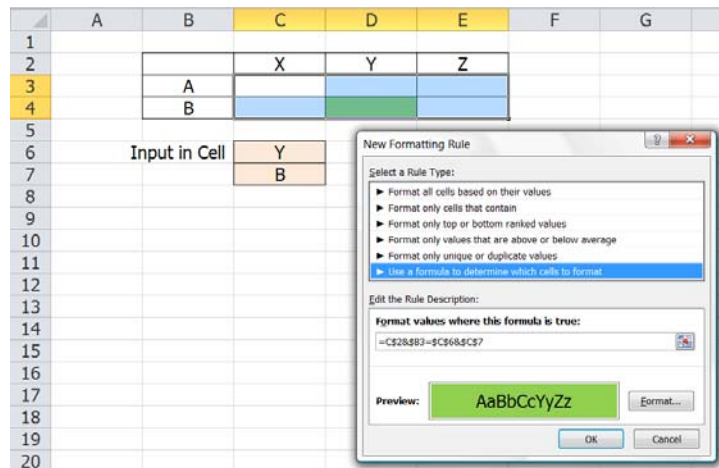
สมมติว่าพอลิกลีเลือกชื่อเดือน May ในเซลล์ D2 แล้ว ต้องการเปลี่ยนสีเซลล์ในตารางที่ตั้งชื่อว่า Choice ในเซลล์ที่มีคำว่า May ตรงตามทีเลือก ให้ทำดังนี้



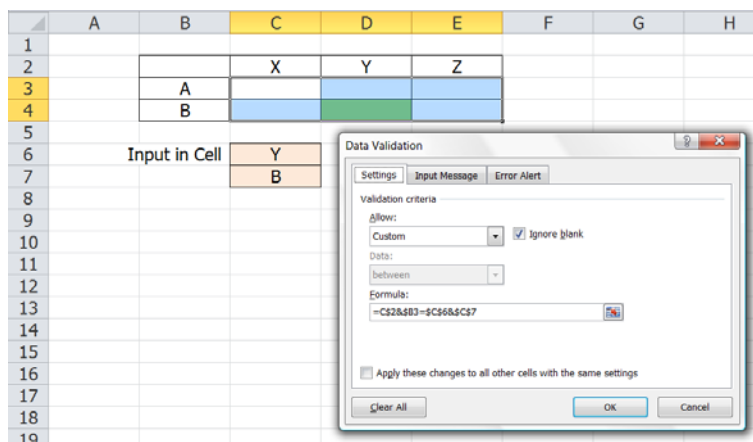
1. เลือกพื้นที่ตารางที่ตั้งชื่อว่า **Choice**
2. สั่ง Home > Conditional Formatting > New Rules > Use a formula to determine which cells to format
3. สร้างสูตร **=B2=\$D\$2** ลงไปในช่อง Edit the Rule Description
4. กดปุ่ม Format แล้วกำหนดรูปแบบที่แตกต่างไปจากเดิมตามต้องการ
5. Data Validation กับ Conditional Formatting ถือเป็นเครื่องมือซึ่งควรนำมาใช้พร้อมกันเสมอ โดย Validation จะทำหน้าที่เสมือนยามเฝ้าประตูตรวจสอบแขกที่จะยอมให้ผ่านประตูเข้ามา แต่ถ้ามีแขกเดินผ่านประตูเข้ามาแล้วก็ต้องใช้ Conditional Formatting เปลี่ยนสีแสดงตัวแขกผู้นั้นขึ้นมา

การใช้ Validation และ Conditional Formatting ควบคุมตำแหน่งของ เซลล์ที่จะเปิดให้พิมพ์ค่า

จากรูปต่อไปนี้ หากต้องการควบคุมให้เปิดบันทึกค่าได้ตามตำแหน่งตัวขึ้นบนหัวตารางที่เป็น Y และตามตำแหน่งข้างตารางที่เป็น B และให้เปลี่ยนสีบอกตำแหน่งอีกด้วย จะพบว่าเซลล์ D4 เปลี่ยนสีและเป็นเซลล์เดียวเท่านั้นที่จะเปิดให้พิมพ์ค่าใหม่ลงไปได้



1. คลิกเลือกพื้นที่เซลล์ C3:E4 ซึ่งเป็นพื้นที่ในตาราง
2. จากนั้นใช้คำสั่ง Conditional Formatting ในทำนองเดียวกับตัวอย่างข้างต้น ต่างกันเพียงใช้สูตร **=C\$2&\$B3=\$C\$6&\$C\$7**
3. ตามด้วยคำสั่ง Data Validation โดยใช้ Allow แบบ Custom โดยกำหนดให้ใช้สูตร **=C\$2&\$B3=\$C\$6&\$C\$7** ซึ่งเป็นสูตรเดียวกันกับที่ใช้กับ Conditional Formatting นั้นเอง (หรือจะใช้สูตร **=And(C\$2=\$C\$6,\$B3=\$C\$7)**)



สูตร Array

เรื่องสูตร Array เป็นเรื่องที่ผมเคยเขียนมาแล้วหลายครั้ง เคยอธิบายมาแล้วหลายรอบ ทุกครั้งที่จัดอบรมก็พูดเรื่องสูตร Array อยู่เสมอ คนที่ฟังเรื่องสูตร Array แล้วเข้าใจก็จะเข้าใจในทันที แล้วก็ต้องมานั่งเบื่อนั่งทนฟังผมอธิบายเรื่องสูตร Array นี้ซ้ำแล้วซ้ำอีก เพราะคนส่วนใหญ่เขาฟังก็ทีก็ทีก็ไม่เข้าใจเรื่องสูตร Array นี้สักที ไม่ใช่ว่าฟังแบบเข้าหูซ้ายแล้วทะลุหูขวาหรอก แต่แย่ยิ่งกว่านั้นอีก พออธิบายเรื่องสูตร Array ที่ไร ปรากฏว่าเสียงพูดอธิบายของผมมันเดินทางไปไม่ถึงหูคนฟังเสียด้วยซ้ำ เพราะคนฟังพอฟังแล้วไม่รู้เรื่องก็จะมีอาการหงุดหงิดขึ้นเรื่อยๆ แล้วหลับหูหลับตาไม่ได้ฟังทุกครั้งไป

"ถ้าคุณใช้สูตร Array เป็น จะแก้ปัญหาอะไรได้บ้าง" ผมคิดว่าต้องเริ่มอธิบายให้พวกเราเห็นคุณประโยชน์ของสูตร Array กันก่อน ถ้าเห็นประโยชน์ของสูตร Array ก็จะช่วยแก้ปัญหาของตัวเองได้มากมายขนาดไหน จะได้เกิดความอยากใช้แล้วจะได้มีความตั้งใจตั้งใจมาเรียนรู้สูตร Array

ประโยชน์ของสูตร Array

สูตร Array จะช่วยให้คุณสามารถทำสิ่งที่ไม่นึกไม่ฝันว่า Excel จะทำได้มาก่อน ได้แก่

1. ทำให้แฟ้มมีขนาดเล็กลง เนื่องจากแทนที่จะต้องสร้างตารางหลายๆตารางเพื่อแยกคำนวณทีละชั้น เราสามารถสร้างสูตร Array ลงไปในเซลล์เพียงเซลล์เดียวก็ได้คำตอบที่ต้องการแล้ว
2. ทำให้เก็บข้อมูลไว้ในตารางเดียวชี้ทีเดียว โดยไม่ต้องแยกตารางฐานข้อมูลออกเป็นหลายตารางหรือหลายชีทเพียงเพื่อต้องการหายอดรวมแยกประเภทตามข้อมูลที่จัดแยกไว้
3. ทำให้ Excel ทำงานแบบคอมพิวเตอร์ได้อย่างแท้จริง พอกรอกค่าลงไปก็จะได้คำตอบที่ต้องการทันที ไม่ต้องเสียแรงและเสียเวลาคลิกเลือกใช้คำสั่งบนเมนู เช่น ไม่ต้องสั่ง Sort ตารางก็จัดเรียงให้เองแล้ว หรือไม่ต้องสั่ง AutoFilter ก็สามารถหายอดรวมแยกประเภทได้แล้ว
4. ทำให้สูตรสำเร็จรูปของ Excel มีความสามารถเหนือชั้นขึ้นกว่าเดิม สามารถใช้สูตรในเซลล์เดียวลัดหาคำตอบ

ในการอบรมครั้งหนึ่งมีผู้เข้าอบรมคนหนึ่งมีท่าทีไม่สนใจเรียนรู้สูตร Array โดยอ้างว่า ทำไมต้องใช้สูตร Array ด้วยในเมื่อ Excel ก็มีสูตรและคำสั่งบนเมนูอยู่แล้ว ถ้าอยากได้อะไรก็แค่คลิกคำสั่งบนเมนู ก็จะได้คำตอบที่ต้องการแล้วมิใช่หรือ

ผมจึงเรียนถามกลับว่า แล้วมือที่ใช้คลิกคำสั่งบนเมนูที่ว่านั้นเป็นมือของใคร ถ้าคุณส่งแฟ้มไปให้หัวหน้าที่ใช้ Excel ไม่เป็น หัวหน้าจะรู้วิธีเลือกคลิกเมอนั้นเมนูนี้ด้วยหรือ เห็นที่หัวหน้าต้องพึ่งพามือของคุณไปชั่วชีวิตหรือไม่ แต่ถ้าคุณสร้างสูตร Array เป็น แฟ้มนั้นๆจะทำงานต่อเองอย่างอัตโนมัติโดยไม่ต้องห่วงว่าใครจะเป็นผู้ใช้แฟ้มของคุณ

อย่างเรื่องการใช้ Pivot Table เหมือนกันที่หัวหน้ามักชอบใช้เป็นชีวิตจิตใจแล้วผลักดันให้ลูกน้องใช้ Pivot Table ตามกันไปทั้งบริษัท โดยหารู้ไม่ว่าถ้าใช้ Array เป็นก็แทบไม่จำเป็นต้องใช้ Pivot Table เสียด้วยซ้ำ

ข้อเสียของสูตร Array

1. ทำให้แฟ้มคำนวณช้าลง แต่เนื่องจากการใช้สูตร Array จะช่วยทำให้แฟ้มมีขนาดเล็กลงซึ่งโดยทั่วไปย่อมส่งผลให้แฟ้มเล็กคำนวณเร็วกว่าแฟ้มใหญ่ ถ้าคุณออกแบบโครงสร้างตารางให้เหมาะกับสูตร Array ด้วย ปัญหาเรื่องนี้ก็จะขดเขยกันไป
2. ทำให้แก้ไขสูตรยากขึ้น เพราะคนทั่วไปมักไม่เข้าใจเรื่องสูตร Array เป็นพื้นฐานอยู่แล้วและไม่ทราบว่าขั้นตอนการสร้างสูตร Array ต้องกดปุ่ม Ctrl+Shift+Enter (CSE) พร้อมกันแทนที่จะกดปุ่ม Enter
3. คนที่ติดนิสัยชอบสร้างสูตร Array มักหลงลืมมองข้ามสูตรสำเร็จรูปอื่นที่มีอยู่แล้วในโปรแกรม Excel ซึ่งสูตรสำเร็จรูปจะใช้เวลาคำนวณเร็วกว่าสูตร Array
4. ใน Excel 365 รุ่นล่าสุดได้พัฒนา Dynamic Array ช่วยทำให้ไม่ต้องกดปุ่ม CSE อีกต่อไป โดย Excel จะกระจายค่าให้เห็นถ้าเป็น Array ที่คืนค่าหลายค่า หรือเมื่อใส่เครื่องหมาย @ จะช่วยทำให้สูตรคืนค่าเดียวออกมา

เบื้องต้นของ Array

ขนาดผมใช้สูตร Array มานานนับสิบปี สร้างสูตร Array เพื่อใช้กับงานมาสารพัดชนิด ตอบคำถามเรื่อง Array ใน Excel Expert Forum ซ้ำแล้วซ้ำอีกนับเป็นพันๆครั้ง แต่ก็ยังมีสูตร Array อีกเยอะที่ผมเห็นแล้วต้องถอนหายใจ ไม่สามารถแกะที่ไปที่มาของลำดับการคำนวณที่ใช้ในสูตร Array เหล่านั้น เพราะเป็นสูตรที่คนอื่นสร้างขึ้น ถ้าคนที่สร้างสูตร เขาไม่ได้เขียนอธิบายแนวคิดประกอบสูตรไว้ด้วย ย่อมทำให้ผู้อื่นคิดตามในใจว่า “เขาสร้างสูตรยาวๆแบบนี้ขึ้นมาได้อย่างไรกัน” แล้วถ้าเราได้แต่ลอกเอาสูตรของคนอื่นมาใช้ ก็ได้แต่ใช้ ไม่สามารถดัดแปลงแก้ไขสูตรที่ลอกมาเพื่อปรับให้เข้ากับเงื่อนไขในงาน วันหนึ่งเราก็จะหน้าแตก เมื่อถูกจับได้ว่าไม่ได้เก่งจริง “อ้อ ลอกสูตรเขามานี่เอง” ดังนั้นถ้าจะคิดใช้สูตร Array ต้องหาทางสร้างสูตรเองให้เป็น แม้สูตร Array ที่สร้างขึ้นเองเป็นสูตรยาวกว่าสูตรของคนอื่นก็ตาม

สูตร Array สูตรแรกๆที่ทำให้ผมเกิดความประทับใจแล้วเกิดแรงบันดาลใจให้อยากเรียนรู้เรื่องสูตร Array มากขึ้น เป็นสูตรที่ใช้คำนวณหาจำนวน Unique Items หรือนับจำนวนชื่อรายการที่ไม่ซ้ำกัน

$$\{=SUM(1/CountIF(DataRange,DataRange))\}$$

สมมติว่าใน DataRange บันทึกชื่อลูกค้าไว้ตามนี้ aa, bb, aa, cc, bb สูตรนี้จะหาคำตอบเป็นเลข 3 ซึ่งหมายถึงการนับชื่อลูกค้าว่ามีอยู่ 3 คน คือ aa, bb, และ cc แต่กว่าจะหาทางสร้างสูตรให้ทำงานได้ก็ต้องทราบก่อนว่าเครื่องหมายวงเล็บปีกกา { } ที่เห็นอยู่ด้านหน้าสูตรและด้านท้ายสุดของสูตรนั้น ไม่ใช่เครื่องหมายวงเล็บปีกกาที่พิมพ์เอง แต่เกิดจากการกดปุ่ม Ctrl+Shift+Enter พร้อมกันและห้ามมีช่องว่างใน DataRange ด้วย

ตอนที่ผมเห็นสูตรนี้ก็คิดไม่ออกว่ามันใช้หลักการใดในการคำนวณ แต่พออ่านคำอธิบายที่มากับสูตรจึงเข้าใจ สูตรนี้เป็นการหายอดรวมของค่าเฉลี่ยต่อตัวของแต่ละรายการใน **DataRange** กล่าวคือ ถ้านับจำนวนชื่อแต่ละชื่อใน DataRange aa, bb, aa, cc, bb จะพบจำนวนของการซ้ำกันดังนี้

- มี aa ซ้ำกันอยู่ 2 ตัว ดังนั้นค่าเฉลี่ยต่อตัวของ aa แต่ละตัวคือ 1/2
- มี bb ซ้ำกันอยู่ 2 ตัว ดังนั้นค่าเฉลี่ยต่อตัวของ bb แต่ละตัวคือ 1/2
- มี cc อยู่ 1 ตัว ดังนั้นค่าเฉลี่ยต่อตัวคือ 1

เมื่อรวมค่าเฉลี่ยต่อตัวเข้าด้วยกัน จาก DataRange aa, bb, aa, cc, bb
 $=1/2+1/2+1/2+1+1/2$ จะได้คำตอบเป็นจำนวนของ Unique Items เท่ากับ 3

จวบจนปัจจุบันนี้ยังตามหาคนที่คิดสร้างสูตรยอดเยี่ยมนี้ขึ้นมาเป็นครั้งแรกไม่พบว่าเป็นใคร ยังดีที่มีคำอธิบายประกอบสูตรนี้ไว้ด้วย

อยากเก่งสูตร Array ต้องสร้างไปแกะไป

จากคำอธิบายที่เขียนไว้เกี่ยวกับที่ไปที่มาของสูตรคำนวณหาจำนวน Unique Items ว่า เป็นการหาอัตรรวมของค่าเฉลี่ยต่อตัวของแต่ละรายการใน DataRange ย่อมทำให้คนส่วนใหญ่ที่เพิ่งรู้จักสูตร Array เป็นครั้งแรกคงยังไม่เข้าใจลำดับในการคำนวณที่ใช้ในสูตรนี้ชัดเจนนัก ถ้าอยากจะเข้าใจอย่างถ่องแท้ว่าสูตร Array ทำงานได้อย่างไร ต้องเรียนรู้จากการปฏิบัติเท่านั้น โดยใช้วิธีทดลองสร้างสูตรไปแกะไปซึ่งมีอยู่ 2 วิธี

1. สร้างสูตร Array ลงไปในเซลล์เดียวแล้วแกะค่าในแต่ละส่วนของสูตร โดยการกดปุ่ม F9 เพื่อดูในช่อง Formula Bar หรือ
2. สร้างสูตร Array ลงไปในตารางพร้อมกันทีเดียวหลายๆเซลล์หรือใช้ตารางแยกคำนวณทีละชั้น เพื่อดูผลลัพธ์กระจายออกมาให้เห็นในตาราง (Excel 365 จะกระจายค่าให้เอง)

โดยทั่วไปจะใช้วิธีแรกสำหรับแกะสูตรที่ไม่ซับซ้อนหรือไม่ยาวนาน พอแกะสูตรโดยการกดปุ่ม F9 แล้วต้องอย่าลืมกดปุ่ม Esc เพื่อย้อนหลังกลับคืนสู่สภาพเดิมก่อนแกะ (ถ้าแกะสูตรแล้วเผลอกด Enter ลงไปจะทำลายสูตรให้กลายเป็นค่าคงที่แทน)

วิธีที่สอง เหมาะสำหรับคนที่เพิ่งเรียนรู้เรื่อง Array หรือใช้แกะสูตรยากๆยาวๆ หรือใช้ทดลองสร้างสูตร Array ขึ้นใหม่ตั้งแต่แรก เพราะวิธีนี้จะเห็นผลลัพธ์แต่ละชั้นของการคำนวณแสดงกระจายให้เห็นในเซลล์แต่ละเซลล์ในตาราง จากนั้นพอซ้อนสูตรต่อกันไปจนได้คำตอบที่ต้องการแล้วจึงนำสูตรแต่ละชั้นมาซ้อนต่อกันเป็นสูตรเดียวเพื่อนำไปใช้งานต่อไป

วิธีแกะสูตร Array วิธีที่ 1

	D3						
	A	B	C	D	E	F	G
1				จำนวน			
2		DataRange		Unique Items			
3		aa		3			
4		bb					
5		aa					
6		cc					
7		bb					
8							
9							

DataRange คือ ตารางช่วง B3:B7

เซลล์ D3 มีสูตร Array **`{=SUM(1/COUNTIF(DataRange,DataRange))}`** โดยเครื่องหมายวงเล็บปีกกาเกิดจากการกดปุ่ม Ctrl+Shift+Enter พร้อมกัน (ไม่ได้พิมพ์ {} เอง)

1. เริ่มจากคลิกลงไปในช่วง Formula Bar คลิกลากหับส่วนของสูตร
COUNTIF(DataRange,DataRange)
2. กดปุ่ม F9 จะพบว่าสูตร CountIF นับจำนวนซ้ำของค่าแต่ละตัว ทำให้สูตรทั้งหมดเปลี่ยนเป็นสูตร =SUM(1/{2;2;2;1;2})
3. ตัวเลข {2;2;2;1;2} หมายถึง จำนวนซ้ำของค่าแต่ละค่า กล่าวคือ aa มี 2 ค่า, bb มี 2 ค่า, aa มี 2 ค่า, cc มี 1 ค่า, และ bb มี 2 ค่า
4. คลิกลากหับส่วนของสูตร 1/{2;2;2;1;2}
5. กดปุ่ม F9 จะพบว่าสูตร 1/{2;2;2;1;2} หาค่าเฉลี่ยจากจำนวนซ้ำของค่าแต่ละตัว ทำให้สูตรทั้งหมดเปลี่ยนเป็นสูตร =SUM({0.5;0.5;0.5;1;0.5})
6. ตัวเลข {0.5;0.5;0.5;1;0.5} เกิดจากส่วนของหารหาค่าเฉลี่ยโดย 1/2 ทำให้เกิดเลข 0.5 และ 1/1 ทำให้เกิดเลข 1
7. คลิกลากหับสูตร SUM({0.5;0.5;0.5;1;0.5}) แล้วกดปุ่ม F9 จะได้คำตอบเป็นสูตร =3
8. กดปุ่ม Esc เพื่อทำให้สูตรแปลงกลับไปเป็นสูตรแรกเริ่มตามเดิม

หมายเหตุ การคลิกแล้วลากหับส่วนของสูตรที่แสดงในช่วง Formula Bar นี้ ต้องคลิกแล้วลากหับส่วนของสูตรที่ครบและพร้อมจะคำนวณได้ เช่น COUNTIF(DataRange,DataRange) แต่ถ้าลากหับขาดหรือเกิน เช่น ลากหับ /COUNTIF(DataRange,DataRange) ซึ่งมีเครื่องหมายหารเกินเข้ามาด้วย จะทำให้เมื่อกดปุ่ม F9 จะพบว่าไม่สามารถแกะสูตรส่วนนั้นได้

วิธีแกะสูตร Array วิธีที่ 2

วิธีนี้ใช้ตารางเป็นตัวช่วยหาผลลัพธ์ของการคำนวณที่ละชั้นแสดงให้เห็นได้ทันทีว่า หลังจากคำนวณชั้นแรกเสร็จแล้ว ส่งผลไปคำนวณต่อในการคำนวณชั้นถัดไปอย่างไรและเกิดผลลัพธ์อย่างไร ซึ่งพอลำดับคำนวณต่อกันไปจนเกิดผลลัพธ์ได้ถูกต้องตามที่ต้องการแล้ว จึงนำสูตรแต่ละชั้นมาซ้อนกันเป็นสูตร Array เพียงสูตรเดียว

	D3						
	A	B	C	D	E	F	G
1							
2		DataRange		Step1	Step2	Step3	
3		aa		2	0.5	3	
4		bb		2	0.5		
5		aa		2	0.5		
6		cc		1	1		
7		bb		2	0.5		
9				D3 : =COUNTIF(DataRange,B3)			
10				E3 : =1/D3			
11				F3 : =SUM(E3:E7)			
12							

ให้เริ่มจากแยกสูตร =SUM(1/COUNTIF(DataRange,DataRange)) ออกเป็นส่วนๆตามลำดับการคำนวณ โดยไล่จากสูตรที่อยู่ภายในวงเล็บในสุดออกมา

- Step1** เซลล์ D3 สร้างสูตร =COUNTIF(DataRange,B3) แล้ว copy ลงมาตลอดแนวจนถึง เซลล์ D7
- ตัวเลขที่ได้จากเซลล์ D3:D7 เป็นตัวเลขการนับว่าข้อมูลแต่ละตัวมีบันทึกไว้ซ้ำกันกี่ครั้งภายใน DataRange
- Step2** เซลล์ E3 สร้างสูตร =1/D3 แล้ว copy ลงมาตลอดแนวจนถึง เซลล์ E7
- ตัวเลขที่ได้จากเซลล์ E3:E7 เป็นตัวเลขค่าเฉลี่ยต่อตัวจากการนับว่าข้อมูลแต่ละตัวมีบันทึกไว้ซ้ำกันกี่ครั้งภายใน DataRange
- Step3** เซลล์ F3 สร้างสูตร =SUM(E3:E7) เพื่อหายอดรวมของค่าเฉลี่ย ได้คำตอบเป็นจำนวนของ Unique Items

หมายเหตุ ใน Step1 และ Step2 แทนที่จะสร้างสูตรลงไปในเซลล์ D3 และ E3 เพียงเซลล์เดียว ให้ทดลองสร้างสูตรคำนวณแบบ Array ลงไปในตารางหลายเซลล์พร้อมกันแทนก็ได้

- Step1** ให้เลือกเซลล์ D3:D7 แล้วพิมพ์สูตร =COUNTIF(DataRange,DataRange) แล้วกดปุ่ม Ctrl+Shift+Enter เพื่อสร้างสูตรนี้ลงไปพร้อมกันทีเดียว จะพบว่าได้คำตอบเป็นการนับค่าซ้ำเช่นเดียวกัน เนื่องจากสูตร CountIF จะใช้แต่ละค่าใน DataRange

ตรวจสอบจำนวนการบันทึกซ้ำจากพื้นที่ DataRange ทั้งหมดเช่นเดียวกับสูตรที่นับแต่ละตัว

- **Step2** ให้เลือกเซลล์ E3:E7 แล้วพิมพ์สูตร $=1/D3:D7$ แล้วกดปุ่ม Ctrl+Shift+Enter เพื่อสร้างสูตรนี้ลงไปพร้อมกันทีเดียว จะพบว่าได้คำตอบเป็นค่าเฉลี่ยเช่นเดียวกัน

Array คืออะไร

บทความนี้ยกความหมายของ Array มาไว้ตอนหลัง เพื่อให้เห็นตัวอย่างการใช้สูตร Array ไว้ก่อนบ้าง ทำให้ได้เห็นเราว่า เรานำสูตร Array ไปใช้งานกันอย่างไร แล้วมันคำนวณหาอะไร พอได้ผ่านการลองทำกับมือมาแล้ว คงพอเดากันได้บ้างว่า Array คืออะไร

Array คือ ค่าตั้งแต่สองค่าขึ้นไป แทนที่จะใช้เซลล์เดียวรับค่าเดียวตามวิธีปกติทั่วไป เราสามารถใช้เซลล์เดียวรับค่าหลายค่าลงไปก็ได้ แทนที่จะใช้พื้นที่ตารางของจริงนับร้อยนับพันเซลล์เพื่อกรอกค่าหรือสร้างสูตรให้คำนวณต่อๆกันไปทีละขั้น ด้วยการใช้ Array จะช่วยให้เราสามารถใส่เซลล์เพียงเซลล์เดียวแทนพื้นที่ตารางขนาดใหญ่ กลายเป็นเซลล์ที่ทำหน้าที่แทนตารางของจริงซึ่งเรียกว่าเป็นตารางในอุดมคติก็ได้

ค่าตั้งแต่สองค่าขึ้นไปจะเรียงตามแนวนอน หรือแนวตั้ง หรือเรียงต่อกันไปเป็นพื้นที่ตารางที่มีความสูงความกว้างหลาย row หลาย column ก็ได้ โดย Excel กำหนดให้ใช้เครื่องหมายในการแบ่งลำดับของค่า ดังนี้

- ถ้าเป็น Array ในแนวนอน ให้ใช้เครื่องหมาย comma , คั่นระหว่างค่าแต่ละค่า เช่น 11,22,33
- ถ้าเป็น Array ในแนวตั้ง ให้ใช้เครื่องหมาย semi-colon ; คั่นระหว่างค่าแต่ละค่า เช่น 11;22;33
- ถ้าเป็น Array ที่มีขนาดความสูง 2 row และมีความกว้าง 3 column จะแสดงค่าที่มีเครื่องหมาย comma และ semi-colon ผสมกัน เช่น 1,2,3;11,22,33 โดย Array จะไล่ลำดับจากแนวนอน 1,2,3 ก่อนแล้วจึงใช้ ; ขึ้นแนวนอนถัดไปเป็น 11,22,33

การนำค่าแบบ Array ไปใช้ในเซลล์ ต่างจากการใช้วิธีพิมพ์ 1,2,3 ลงไปในเซลล์ตามปกติ โดยต้องใช้ Array แบบสูตรที่มีเครื่องหมายเท่ากับนำหน้าสูตรเท่านั้น ดังนี้

1. **การใช้แบบสูตร Array Constant** เช่น พิมพ์ $=\{1,2,3\}$ ลงไปในเซลล์ใดๆ จะพบว่าในเซลล์แสดงค่าแรกคือเลข 1 เพียงค่าเดียว ซึ่งถ้าต้องการแสดงให้ครบทุกค่า ต้องเลือกเซลล์ 3 เซลล์ติดกันตามแนวนอนแล้วพิมพ์ $=\{1,2,3\}$ แล้วกดปุ่ม Ctrl+Shift+Enter เพื่อบันทึก จะพบว่าสูตร $=\{1,2,3\}$ ซึ่งมีวงเล็บปีกกาอยู่ด้านหลังเครื่องหมายเท่ากับ จะมีวงเล็บปีกกาเพิ่มด้านหน้าเครื่องหมายเท่ากับอีกชั้นหนึ่งเป็น $\{=\{1,2,3\}\}$
2. **การใช้แบบสูตร Array** ซึ่งรับค่ามาจากการอ้างอิงกับพื้นที่ในตาราง เช่น รับค่า Array มาจากเซลล์ตามแนวนอน $\{=A1:C1\}$ หรือรับค่า Array มาจากเซลล์ตามแนวตั้ง $\{=A1:A3\}$ หรือรับค่า Array มาจากตารางขนาดความสูง 2 row และมีความกว้าง 3 column $\{=A1:C2\}$ ทั้งนี้โปรดสังเกตว่าวิธีนี้จะมีเครื่องหมายวงเล็บปีกกาแสดงไว้หน้าเครื่องหมายเท่ากับ แสดงว่าเป็นวงเล็บปีกกาที่เกิดจากการกดปุ่ม Ctrl+Shift+Enter
3. **การใช้แบบสูตร Array** ซึ่งรับค่ามาจากสูตรที่คืนค่าแบบ Array เช่น $\{=Offset(A1,0,0,2,3)\}$ ซึ่งเทียบเท่ากับการอ้างอิงมาจากเซลล์ A1:C2 เป็นต้น

การนำค่าแบบ Array ทั้ง 3 แบบไปใช้ ขอให้เลือกพื้นที่ตารางให้มีขนาดพอดีกับจำนวนของค่าที่มีอยู่ใน Array หรือให้มีขนาดตารางเท่ากับจำนวนผลลัพธ์ที่ต้องการไว้ก่อนที่จะสร้างสูตรลงไป กล่าวคือ ถ้าในสูตร Array คืนค่าสุดท้ายเพียงค่าเดียวก็ให้สร้างสูตรลงไปเพียงเซลล์เดียว แต่ถ้าสูตร Array คืนค่าผลลัพธ์ที่มีจำนวนค่ามากกว่าค่าเดียว ก็ต้องเลือกพื้นที่ตารางให้มีขนาดความสูงและความกว้างเท่ากับจำนวนและลำดับค่าที่มีอยู่ จากนั้นให้กดปุ่ม Ctrl+Shift+Enter ในการรับสูตร Array ลงไป จะพบว่า Excel กระจายค่าที่มีอยู่ลงไปในพื้นที่ตารางที่เตรียมไว้ให้เห็นได้ครบทุกค่า

ตัวอย่างการสร้างตารางสูตรคูณแบบ Array

C3		{=Top*Left}						
	A	B	C	D	E	F	G	H
1								
2			2	3	4	5		
3		10	20	30	40	50	#N/A	#N/A
4		20	40	60	80	100	#N/A	#N/A
5		30	60	90	120	150	#N/A	#N/A
6			#N/A	#N/A	#N/A	#N/A	#N/A	#N/A
7			#N/A	#N/A	#N/A	#N/A	#N/A	#N/A
8								
9		Total	840	C9 : {=SUM(Top*Left)}				
10								
11		Top	=Sheet2!\$C\$2:\$F\$2					
12		Left	=Sheet2!\$B\$3:\$B\$5					
13								

เมื่อตั้งชื่อ Range Name ให้กับตัวเลขบนหัวตารางและข้างตารางว่า Top และ Left เสร็จแล้วให้ทดลองเลือกพื้นที่เซลล์ C3:H7 ซึ่งเกินกว่าแนวขอบเขตตารางตัวเลขที่มี แล้วสร้างสูตร =Top*Left แล้วกดปุ่ม Ctrl+Shift+Enter จะได้สูตร Array {=Top*Left}

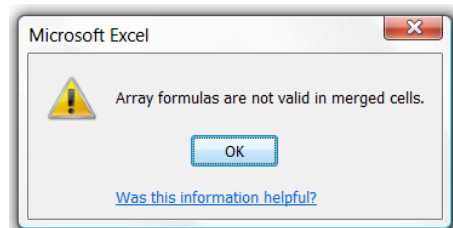
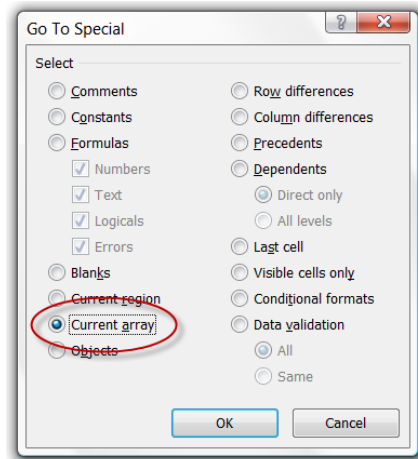
สังเกตว่าเฉพาะพื้นที่ตาราง C3:F5 ซึ่งอยู่ภายในขอบเขตของ Top กับ Left เท่านั้นสามารถคำนวณหาผลคูณเป็นตารางสูตรคูณที่ต้องการ ส่วนพื้นที่นอกเหนือจาก C3:F5 ที่มีสูตร Array จะคืนค่าเป็น Error #N/A

จากนั้นให้คลิกเซลล์ใดก็ได้ที่มีสูตร {=Top*Left} แล้วกดปุ่ม F2 ตามด้วยปุ่ม F9 จะแสดงผลเป็น Array Constant {20,30,40,50;40,60,80,100;60,90,120,150} โดยผลลัพธ์นี้มีเครื่องหมาย ; คั่นอยู่ 2 ตัว แสดงว่ามี 3 row เพราะมีการขึ้น row ใหม่ 2 ครั้ง และในช่วงที่คั่นด้วย ; นั้น มีเครื่องหมาย , คั่นอยู่ช่วงละ 3 ตัว แสดงว่าในแต่ละ row มีค่าอยู่ 4 column หรืออีกนัยหนึ่งแสดงว่า พื้นที่ตารางที่เหมาะสมจะสร้างสูตร {=Top*Left} ลงไปนั้น ต้องมีความสูง 3 row และกว้าง 4 column ดังนั้นหากเลือกพื้นที่เกินกว่าที่จำเป็นก็จะได้อาคตอบเป็น Error #N/A

แต่ถ้าต้องการหาผลรวมของผลคูณของ =Top*Left ในเซลล์ C9 ให้สร้างสูตร =SUM(Top*Left) แล้วกดปุ่ม Ctrl+Shift+Enter จะได้สูตร Array {=SUM(Top*Left)} ซึ่งสาเหตุที่คราวนี้สร้างสูตรลงไปในพื้นที่ C9 เพียงเซลล์เดียว เพราะคำตอบของการหาผลรวมมีเพียงค่าเดียวเท่านั้น

เรื่องอื่นๆเกี่ยวกับ Array ที่ควรทราบ

1. ใน Excel Help เรียกพื้นที่ใดๆที่มีจำนวนตั้งแต่ 2 เซลล์ขึ้นไปว่า Array
2. การแกะดูค่าที่คำนวณได้จากสูตร Array ให้เริ่มจากคลิกลงไปในเซลล์สูตรแล้วกดปุ่ม **F2** แล้วตามด้วยปุ่ม **F9** เมื่อเห็นค่าที่คำนวณได้แล้วให้กดปุ่ม Esc เพื่อกลับเป็นสูตรตามเดิม ทั้งนี้ถ้าค่าที่แกะได้จากสูตร Array คำนวณค่าจนเกินกว่าที่เซลล์หนึ่งจะรับได้ จะพบว่า Excel ไม่ยอมแสดงค่าให้เห็น
3. วิธีค้นหาพื้นที่ของสูตร Array ที่เกิดจากการสร้างขึ้นหลายเซลล์พร้อมกัน ให้เริ่มจากคลิกเลือกเซลล์ใดเซลล์หนึ่งซึ่งใช้สูตร Array แล้วกดปุ่ม **F5 > Special > กาช่อง Current array**
4. บางคนกดปุ่ม Ctrl+Shift+Enter แต่ไม่ได้เครื่องหมายวงเล็บปีกกา เพราะไม่ได้กดทั้งสามปุ่มพร้อมกันจริง ถ้าอยากให้ได้ { } อย่างแน่นอน ให้ใช้มือซ้ายกดปุ่ม Ctrl+Shift แยกไว้ก่อน จากนั้นใช้มือขวา กดปุ่ม Enter ลงไป
5. สูตร Array ที่เกิดจากการสร้างขึ้นหลายเซลล์พร้อมกันโดยการกดปุ่ม Ctrl+Shift+Enter จะไม่สามารถ Insert หรือ Delete เฉพาะบางเซลล์ หากต้องการแก้ไขหรือลบทิ้ง ต้องเลือกพื้นที่ทั้งหมดที่ใช้สูตร Array นั้นก่อนแล้วจึงแก้ไขสูตรหรือลบสูตรทิ้ง (ดังนั้นหากสามารถลบสูตร Array เพียงเซลล์เดียวได้ ย่อมแสดงว่าเป็นสูตร Array ที่สร้างที่ละเซลล์)
6. สูตร Array ที่เราสร้างขึ้นโดยการกดปุ่ม Ctrl+Shift+Enter จะไม่สามารถสร้างลงไปในเซลล์ที่ถูก Merge ไว้ก่อนแล้ว จะถูกเตือนว่า Array formulas are not valid in merged cells (แต่เราสามารถ Merge เซลล์ที่สร้างสูตร Array ไว้ก่อน)
7. สูตร Array ที่เกิดจากการสร้างขึ้นหลายเซลล์พร้อมกันโดยการกดปุ่ม Ctrl+Shift+Enter จะใช้เวลาคำนวณเร็วกว่าการสร้างสูตรคำนวณเพื่อหาคำตอบที่ละเซลล์แยกจากกัน
8. สูตร Array ที่เราสร้างขึ้นโดยการกดปุ่ม Ctrl+Shift+Enter ทั้งที่สร้างในเซลล์เดียวหรือหลายเซลล์พร้อมกัน จะใช้เวลาคำนวณช้ากว่าสูตรสำเร็จรูปที่มีอยู่ใน Excel ดังนั้นจึง



แนะนำให้ใช้สูตร Array ต่อเมื่อไม่มีสูตรสำเร็จรูปอื่นซึ่งสามารถคำนวณหาคำตอบที่ต้องการได้แล้วเท่านั้น

9. ใน Excel รุ่น 2003 และรุ่นเก่าก่อนนั้น สูตร Array ที่เราสร้างขึ้นโดยการกดปุ่ม Ctrl+Shift+Enter จะไม่ยอมรับการอ้างอิงพื้นที่ตารางที่มีขนาดใหญ่เต็มความสูงทั้ง 65,536 row ของตาราง เช่น

$$\{=\text{MIN}(\text{IF}(\text{A:A}<>0,\text{ROW}(\text{A:A})))\}$$
 หรือ

$$\{=\text{MIN}(\text{IF}(\text{A1:A65536}<>0,\text{ROW}(\text{A1:A65536})))\}$$

 แต่ถ้าแก้เป็น $\{=\text{MIN}(\text{IF}(\text{A1:A65535}<>0,\text{ROW}(\text{A1:A65535})))\}$ จะใช้ได้
10. นอกจากสูตร Array ที่เราสร้างขึ้นโดยการกดปุ่ม Ctrl+Shift+Enter แล้ว ใน Excel ยังมีสูตรสำเร็จรูปอื่นอีกที่ทำงานแบบ Array โดยไม่ต้องกดปุ่ม Ctrl+Shift+Enter เช่น สูตร SumIF, CountIF, SumIFs, CountIFs, AverageIfs, SumProduct ซึ่งถ้าพิจารณาตามหลักการแล้วสูตรใดก็ตามซึ่งรับค่าจากเซลล์ตั้งแต่ 2 เซลล์ขึ้นไปได้ ย่อมเป็นสูตรที่ทำงานแบบ Array จึงยังมีสูตรอีกมากที่สามารถทำงานแบบ Array ได้ เช่น สูตร Sum, Max, Min, And, Or, Index, หรือ Offset เป็นต้น
11. SumIFs, CountIFs, AverageIfs เป็นสูตรที่เกิดขึ้นใน Excel 2007 เป็นต้นมา ดังนั้น หากยังจำเป็นต้องใช้ Excel 2003 หรือรุ่นเก่ากว่านี้อยู่อีก แนะนำให้หลีกเลี่ยงสูตร SumIFs, CountIFs, AverageIfs ไปก่อน โดยหันไปใช้สูตร Sum-IF-Array, Count-IF-Array, Average-If-Array ซึ่งสามารถใช้งานใน Excel ได้ทุกรุ่น แต่จำเป็นต้องสร้างโดยการกดปุ่ม Ctrl+Shift+Enter
12. Max, Min, And, Or เมื่อนำมาใช้แบบ Array จะไม่สามารถคืนค่าหลายค่าแบบ Array เช่น ถ้าสร้าง $\{=\text{Max}(\text{RangeA},\text{RangeB})\}$ จะไม่ได้คำตอบเป็นค่าสูงสุดของแต่ละตำแหน่งใน RangeA เมื่อนำมาเทียบกับแต่ละตำแหน่งใน RangeB แต่จะได้คำตอบเป็นค่าสูงสุดเพียงค่าเดียวจากค่าทั้งหมดใน RangeA และ RangeB
13. สูตร Array บางสูตรไม่สามารถหาคำตอบมาแสดงให้เห็นในเซลล์ แต่ถ้านำสูตรนั้นไปซ่อนในสูตรอื่นจะสามารถทำงานร่วมกับสูตรอื่นได้
14. สูตร Array ที่คืนค่าคำตอบหลายค่า ไม่ควรสร้างสูตรนั้นลงไปในเซลล์เดียวเพราะจะได้คำตอบไม่ครบทั้งหมดหรือได้คำตอบ error
15. ถ้าสร้างสูตร Array ที่คืนค่าคำตอบหลายค่า ลงไปในเซลล์เดียวในแนวเดียวกับคำตอบจากตารางฐานข้อมูล จะได้คำตอบจากเซลล์ในแนวคู่ขนานกับตำแหน่งเซลล์ที่สร้างสูตร

ลงไป ซึ่งถือเป็นความบังเอิญเท่านั้น ถ้าย้ายสูตรไปนอกแนวคู่ขนานจะได้คำตอบ error

D2		=OFFSET(\$B\$4,0,0,5,1)						
	A	B	C	D	E	F	G	H
1								
2				#VALUE!	D2 : =OFFSET(\$B\$4,0,0,5,1)			
3		DataRange						
4		aa		aa	D4 : '=OFFSET(\$B\$4,0,0,5,1)			
5		bb						
6		aa		aa	D6 : '=OFFSET(\$B\$4,0,0,5,1)			
7		cc						
8		bb		bb	D8 : '=OFFSET(\$B\$4,0,0,5,1)			
9								

16. ถ้าสร้างสูตร Array ลงไปในตารางหลายเซลล์พร้อมกัน จะพบว่าการสร้างสูตรตอนที่คลิกเลือกพื้นที่เซลล์ที่เคยตั้งชื่อ Range Name ไว้ก่อนหน้านี้ Excel จะไม่ยอมรับชื่อ Range Name มาใส่ในสูตร เช่น เดิมตั้งชื่อเซลล์ B4:B8 ว่า DataRange จากนั้นเมื่อเลือกเซลล์อื่นใดเพื่อจะสร้างสูตรลงไปพร้อมกัน พอพิมพ์ = แล้วคลิกเลือกเซลล์ B4:B8 จะไม่ได้ชื่อ DataRange มาใส่ในสูตร แต่จะได้ตำแหน่งเซลล์ B4:B8 แสดงในสูตร เหมือนว่าไม่เคยมีชื่อ DataRange ตั้งไว้ก่อนแต่อย่างใด หากต้องการนำชื่อ Range Name มาใส่ในสูตร ให้กดปุ่ม **F3** เพื่อเลือกชื่อที่ต้องการมาใช้แทนการคลิกเลือกเซลล์
17. ถ้าอยากจะเก่ง Excel ให้ทดลองสร้างสูตรกับตารางขนาดเล็กให้ผ่านก่อน แล้วให้ลองกดปุ่ม Enter ตามธรรมดาบ้าง หรือกดปุ่ม Ctrl+Shift+Enter บ้าง เพราะบางครั้งจะได้คำตอบต่างไปจากเดิม
18. ไม่ควรปล่อยให้คนที่ไม่รู้จักรหัสสร้างสูตร Array ที่สร้างขึ้นโดยการกดปุ่ม Ctrl+Shift+Enter มาแก้ไขแฟ้มที่มีสูตร Array เพราะเขาจะกดปุ่ม Enter แทนแล้วอาจทำให้ได้คำตอบผิดเพี้ยนไปหรือได้คำตอบ error ขึ้นมาแทน

สูตร Array IF

เพื่อใช้หายอดรวมแยกประเภท

สูตร Array IF เพื่อใช้หายอดรวมแยกประเภท ถือเป็นสูตรที่สำคัญเป็นอย่างยิ่งและเป็นสูตรที่ใช้บ่อยที่สุดสูตรหนึ่งทีเดียว จนใน Excel 2007 เป็นต้นมาบริษัทไมโครซอฟท์ได้เพิ่มสูตร SumIFs, CountIFs, และ AverageIFs เพื่อตอบสนองต่อความต้องการของผู้ใช้ เพราะใน Excel รุ่นก่อนนั้นมีเพียงสูตร SumIF และ CountIF ซึ่งไม่สามารถรับเงื่อนไขในการคำนวณเกินกว่า 1 เงื่อนไข

แต่ถ้าคุณนำแฟ้มที่ใช้สูตร SumIFs, CountIFs, และ AverageIFs ไปเปิดด้วย Excel 2003 หรือรุ่นเก่ากว่า จะพบว่าสูตรใหม่เหล่านี้ไม่ทำงานเพราะ Excel รุ่นที่จะใช้สูตรใหม่ได้ก็ต้องเป็น Excel รุ่นใหม่เช่นกัน

บทความนี้จะแนะนำแนวทางการสร้างสูตร Array IF ว่ามีที่ไปที่มาอย่างไร เพื่อช่วยให้สามารถสร้างสูตรหายอดรวมแยกประเภทได้โดยไม่ต้องอาศัยสูตร SumIFs, CountIFs, และ AverageIFs แต่อย่างใด แล้วเมื่อคุณทราบหลักการทำงานของการใช้ Array IF หรือ Array แบบเงื่อนไข ยังจะช่วยให้สามารถนำไปพัฒนาสร้างสูตรหาค่าตอบประเภทอื่นๆได้ไม่ยาก

สูตร Array IF คืออะไร

สูตร Array IF คือ สูตร IF ที่ทำหน้าที่ตรวจสอบค่าหลายค่าว่าตรงตามเงื่อนไขที่กำหนดหรือไม่ โดยจะได้ผลลัพธ์จากการตรวจสอบเป็น True หรือ False หลายค่าตามลำดับตำแหน่งของค่าที่ใช้ ซึ่งต่างจากสูตร IF ธรรมดาที่มีการตรวจสอบค่าเพียงค่าเดียวและเกิดผลลัพธ์เป็น True หรือ False อย่างใดอย่างหนึ่งเพียงค่าเดียว

โดยลำพังของสูตร Array IF จะไม่สามารถหายอดรวมแยกประเภทได้ แต่เมื่อนำผลลัพธ์ True หรือ False ที่ได้นั้นไปเทียบกับ Array ของตัวเลขคำตอบที่ต้องการ จะทำให้เราสามารถเลือกเก็บตัวเลขเฉพาะตำแหน่งค่าที่ตรงกับตำแหน่งของ True แล้วจึงนำผลที่ได้ไปหายอดรวมกับสูตรอื่นๆก็จะกลายเป็นสูตร Array เพื่อหาค่าตามสูตรนั้นๆ เช่น เมื่อนำผลที่ได้หลังจากการเทียบตำแหน่ง True/False ไปซ่อนในสูตร Sum ก็จะกลายเป็นสูตร Array Sum IF ซึ่งหายอดรวม

แยกประเภท หรือถ้าซ่อนในสูตร Average ก็จะกลายเป็นสูตร Array Average IF ซึ่งหาค่าเฉลี่ยแยกประเภท

โครงสร้างของสูตร Array IF ประกอบด้วย 2 ส่วนใหญ่ๆ ซึ่งโดยทั่วไปทั้ง 2 ส่วนต้องมีขนาดความสูงความกว้างของ Array เท่ากัน กล่าวคือ

1. ส่วนของ Array ที่ทำหน้าที่คืนค่า True หรือ False
2. ส่วนของ Array ที่เป็นตัวเลขหรือค่าคำตอบที่ต้องการ

(ในเบื้องต้นขอให้จำไว้ว่าขนาด Array ทั้งสองส่วนนี้ต้องมีขนาดเท่ากัน ซึ่งที่จริงแล้วขอเพียงว่ามีความสูงหรือขนาดความกว้างด้านใดด้านหนึ่งเท่ากันก็ใช้ได้แล้ว)

การใช้สูตร Array IF แบบเงื่อนไขเดียว

	A	B	C	D	E	F	G	H
1								
2		Num	Id	Name	Amount		Id	Amount
3		1	a001	a	10		a001	???
4		2	a002	b	20			
5		3	a001	c	30			
6		4	a004	d	40			
7		5	a002	e	50			
8								
9		Num	=Sheet1!\$B\$3:\$B\$7					
10		Id	=Sheet1!\$C\$3:\$C\$7					
11		Name	=Sheet1!\$D\$3:\$D\$7					
12		Amount	=Sheet1!\$E\$3:\$E\$7					
13								

โปรดสังเกตว่าใน Range ID มีรหัส a001 และ a002 เป็นรายการที่บันทึกซ้ำ ซึ่งถ้ามีรหัสไม่ซ้ำแล้วต้องการหาค่า Amount ของ a001 จะสามารถหาค่าตอบโดยใช้สูตร Vlookup ได้ทันที แต่เมื่อมีรายการซ้ำ ถ้าใช้สูตร Vlookup ก็จะได้ยอดของ a001 รายการแรกเพียงรายการเดียว หากต้องการหาค่ารวม Amount ของ a001 หรือจะแยกหาค่าของแต่ละรายการที่ซ้ำของ a001 ก็ต้องอาศัยสูตรคำนวณแบบ Array มาช่วย

ถ้าไม่ใช้สูตร Array IF จะหาค่ารวม Amount ของรหัส a001 จากการคำนวณทีละขั้นได้อย่างไร

E8		=SUM(E3:E7)				
A	B	C	D	E	F	G
1						
2		Id	Check	Amount	Amount of a001	Id
3		a001	TRUE	10	10	a001
4		a002	FALSE	20	0	
5		a001	TRUE	30	30	
6		a004	FALSE	40	0	
7		a002	FALSE	50	0	
8					40	
9						
10		C3 : =Id=\$G\$3		E3 : =IF(Check,Amount,0)		
11				E8 : =SUM(E3:E7)		
12						

1. เพื่อให้ตารางแสดงเฉพาะพื้นที่ซึ่งเกี่ยวข้องกับการคำนวณเท่านั้น จึงขอลบพื้นที่ส่วนของ Num และ Name ทิ้งไป แล้ว Insert เพิ่ม column C และ E มาใช้คำนวณที่ละชั้น
2. ตั้งชื่อ Check ให้กับเซลล์ C3:C7 แล้วสร้างสูตร =Id=\$G\$3 จะพบว่าได้คำตอบเป็น TRUE; FALSE; TRUE; FALSE; FALSE ตามลำดับ โดยตำแหน่งที่เป็น True คือ ตำแหน่งของรหัส a001 นั่นเอง
3. เซลล์ E3:E7 สร้างสูตร =IF(Check,Amount,0) จะพบว่าถ้าตำแหน่งใดใน Check มีค่าเท่ากับ True สูตร IF นี้จะเก็บค่า Amount มาใช้ตามเดิม แต่ถ้าไม่ใช่จะใส่ค่า 0 แทน
4. เซลล์ E8 สร้างสูตร =SUM(E3:E7) ได้คำตอบเท่ากับ 40 ซึ่งเป็นยอดรวมแยกประเภทของรหัส a001 ตามต้องการ

แทนที่จะต้องสร้างตารางเพิ่มเติมเพื่อคำนวณที่ละชั้น เราสามารถหาคำตอบยอดรวมแยกประเภทของรหัส a001 ที่ต้องการโดยพิจารณาแยกพื้นที่ตารางเป็น 2 ส่วน

H6		fx		=SUMPRODUCT((Id=\$G\$3)*Amount)							
	A	B	C	D	E	F	G	H	I	J	K
1											
2			Id		Amount		Id	Amount			
3			a001		10		a001	40			
4			a002		20			40			
5			a001		30			40			
6			a004		40			40			
7			a002		50						
8											
9											
10											
11											
12											

H3 : =SUMIF(Id,\$G\$3,Amount)

H4 : {=SUM(IF(Id=\$G\$3,Amount))}

H5 : {=SUM((Id=\$G\$3)*Amount)}

H6 : =SUMPRODUCT((Id=\$G\$3)*Amount)

1. Id เป็นส่วนของ Array ที่ทำหน้าที่คืนค่า True หรือ False จากการนำไปเปรียบเทียบกับรหัสที่ต้องการ
2. Amount เป็นส่วนของ Array ที่เป็นตัวเลขหรือค่าคำตอบเรื่องที่ต้องการ

จากนั้นเราสามารถหาคำตอบยอดรวมแยกประเภทของรหัส a001 ซึ่งบันทึกไว้ในเซลล์ G3 เป็นเซลล์สำหรับใส่รหัสที่ต้องการ แล้วใช้สูตรต่อไปนี้หาผลลัพธ์ที่ต้องการ

1. สูตร **SUMIF** โดยใช้สูตร =SUMIF(Id,\$G\$3,Amount) หรือ
2. สูตร **Array SumIF** โดยใช้สูตร { =SUM(IF(Id=\$G\$3,Amount))} โดยต้องกดปุ่ม Ctrl+Shift+Enter ด้วยจึงจะเกิดวงเล็บปีกกา { } ปิดหัวท้ายสูตร หรือ
3. สูตร **SumArray** โดยใช้สูตร { =SUM((Id=\$G\$3)*Amount)} โดยต้องกดปุ่ม Ctrl+Shift+Enter ด้วยจึงจะเกิดวงเล็บปีกกา { } ปิดหัวท้ายสูตร หรือ
4. สูตร **SumProduct** โดยใช้สูตร =SUMPRODUCT((Id=\$G\$3)*Amount) โดยไม่ต้องกดปุ่ม Ctrl+Shift+Enter ด้วยแต่อย่างใด

สูตรทั้งสี่แบบใช้หลักการคำนวณแบบเดียวกัน คือ

1. Id เป็นส่วนของ Array ที่ทำหน้าที่คืนค่า True หรือ False จากการนำไปเปรียบเทียบกับรหัส a001 ที่ต้องการ จะคืนค่าออกมาเป็น Array {TRUE;FALSE;TRUE;FALSE;FALSE}
2. Amount เป็นส่วนของ Array ที่เป็นตัวเลขหรือค่าคำตอบเรื่องที่ต้องการ {10;20;30;40;50}
3. เมื่อนำ Array {TRUE;FALSE;TRUE;FALSE;FALSE} มาเทียบตำแหน่งกับ Array {10;20;30;40;50} จะได้ Array ตัวเลขเฉพาะตัวเลขคำตอบที่ต้องการออกมาเป็น {10;FALSE;30;FALSE;FALSE}
4. จากนั้นเมื่อนำผล {10;FALSE;30;FALSE;FALSE} ไปหาผลรวมด้วยสูตร Sum จึงหาคำตอบเป็น 40 ตามต้องการ

แม้ว่าสูตรทั้งสี่แบบนี้จะหาผลลัพธ์ที่ต้องการได้เช่นเดียวกันก็ตาม แต่ขอให้พิจารณาเลือกใช้ให้เหมาะสมกับงาน กล่าวคือ

1. สูตร **SUMIF** ให้ใช้กับโจทย์ที่ง่ายไปตลอดเพราะสูตรนี้รับเงื่อนไขได้เพียงเงื่อนไขเดียว และสูตรนี้ใช้เวลาคำนวณเร็วที่สุดในบรรดาสูตรที่กล่าวถึงอยู่นี้

2. **สูตร Array SumIF** สูตรนี้เป็นสูตรพิเศษ เพราะเราสามารถเปลี่ยนสูตร Sum ด้านหน้าไปเป็นสูตร Average, Count, Max, Min, หรือ Small เพื่อหายอดอื่นๆตามสูตรที่ใช้แทนนั้น และถ้ามีหลายเงื่อนไขก็สามารถซ้อน IF เข้าไปได้อีก

{ =SUM(IF(Id=\$G\$3,Amount))} หายอดรวมของ a001 ได้เท่ากับ 40

{ =AVERAGE(IF(Id=\$G\$3,Amount))} หาค่าเฉลี่ยของ a001 ได้เท่ากับ 20

{ =COUNT(IF(Id=\$G\$3,Amount))} นับยอดของ a001 ได้เท่ากับ 2

{ =MAX(IF(Id=\$G\$3,Amount))} หายอดสูงสุดของ a001 ได้เท่ากับ 30

{ =MIN(IF(Id=\$G\$3,Amount))} หายอดต่ำสุดของ a001 ได้เท่ากับ 10

3. **สูตร SumArray** เป็นสูตรที่ผมนิยมใช้เพราะใช้มานานและพบว่าสามารถหาคำตอบที่ต้องการได้เสมอแม้จะลิงค์ข้ามแฟ้มมาจากแฟ้มที่ไม่ได้เปิดก็ยังทำงานได้ และถ้ามีหลายเงื่อนไข การใช้วงเล็บหลายๆชุดช่วยตรวจสอบเงื่อนไขยังง่ายกว่าที่จะใช้สูตร IF ซ้อนกันเข้าไป

4. **สูตร SumProduct** เป็นสูตรยอดนิยมของคนทั่วไปเพราะไม่จำเป็นต้องกดปุ่ม Ctrl+Shift+Enter ด้วยแต่อย่างใด ซึ่งแท้จริงแล้วโครงสร้างภายในวงเล็บของสูตรนี้ก็เหมือนกับของสูตร SumArray นั่นเอง

เรื่องแปลกของ True กับ False

คุณควรทราบลักษณะสำคัญของค่า True หรือ False เพิ่มเติมก่อนที่จะเรียนรู้วิธีใช้สูตร Array IF แบบหลายเงื่อนไข กล่าวคือ Excel ถือว่าค่า True มีค่าเท่ากับ 1 และค่า False มีค่าเท่ากับ 0 ซึ่งเราจะเห็นค่าเหล่านี้ได้ชัด ต่อเมื่อนำค่า True หรือ False มาคำนวณต่อ

แต่ถ้านำค่า True ที่ได้ไปเทียบตรงๆกับเลข 1 ว่าเท่ากันหรือไม่ จะพบว่า True ไม่เท่ากับ 1 และถ้านำค่า False ที่ได้ไปเทียบตรงๆกับเลข 0 ว่าเท่ากันหรือไม่ จะพบว่า False ไม่เท่ากับ 0

C2		=B2=1								
	A	B	C	D	E	F	G	H	I	J
1										
2		TRUE	FALSE	C2 : =B2=1		FALSE	FALSE	H2 : =G2=0		
3			1	C3 : =B2*1			0	H3 : =G2*1		
4			1	C4 : =B2/1			0	H4 : =G2/1		
5			1	C5 : =B2+0			0	H5 : =G2+0		
6			1	C6 : =B2-0			0	H6 : =G2-0		
7			1	C7 : =-B2			0	H7 : =-G2		
8			1	C8 : =B2*TRUE()			0	H8 : =G2*TRUE()		
9			0	C9 : =B2*FALSE()			0	H9 : =G2*FALSE()		
10										

จากภาพตัวอย่างนี้ เซลล์ B2 มีค่า =TRUE() จากนั้นเรามาทดสอบค่าของ B2 กัน จะพบว่าในเซลล์ C2 เมื่อนำมาเทียบค่ากันโดยตรง ซึ่งมีสูตร =B2=1 บอกเราว่า False แสดงว่า True ไม่ได้เท่ากับ 1 แต่เมื่อนำค่าจาก B2 ไปคำนวณต่อในเซลล์ C3:C8 ไม่ว่าจะนำ B2 ไป *1, /1, +0, -0, หรือใส่เครื่องหมายลบลบไว้ข้างหน้า หรือนำไปคูณกับ TRUE() จะกระตุ้นให้แสดงค่าเท่ากับเลข 1 ออกมาให้เห็น

ทำนองเดียวกัน เซลล์ G2 มีค่า =FALSE() จากนั้นเรามาทดสอบค่าของ G2 กัน จะพบว่าในเซลล์ H2 เมื่อนำมาเทียบค่ากันโดยตรง ซึ่งมีสูตร =G2=0 บอกเราว่า False แสดงว่า False ไม่ได้เท่ากับ 0 แต่เมื่อนำค่าจาก G2 ไปคำนวณต่อในเซลล์ H3:H8 ไม่ว่าจะนำ G2 ไป *1, /1, +0, -0, หรือใส่เครื่องหมายลบลบไว้ข้างหน้า หรือนำไปคูณกับ TRUE() จะกระตุ้นให้แสดงค่าเท่ากับเลข 0 ออกมาให้เห็น

ดังนั้นถ้าเรามี Array ซึ่งมีค่า True หรือ False แล้วนำไปบวกหรือคูณกับ Array อื่นที่มีค่าเป็นตัวเลขหรือมีค่าเป็น True หรือ False ก็ตาม ย่อมเหมือนกับนำเลข 1 หรือเลข 0 ไปบวกหรือคูณกับค่าอื่นนั่นเอง โปรดดูข้อพิสูจน์จากรูปต่อไปนี้

	A	B	C	D	E	F	G	H
1								
2		กรณีบวกกัน				กรณีคูณกัน		
4		D4 : =B4+C4				H4 : =F4*G4		
5		TRUE	FALSE	1		TRUE	FALSE	0
6		FALSE	FALSE	0		FALSE	FALSE	0
7		FALSE	TRUE	1		TRUE	TRUE	1
8		FALSE	FALSE	0		FALSE	FALSE	0
9		FALSE	FALSE	0		FALSE	FALSE	0

Array ที่มีเลข 1 และ 0 จากการบวกหรือคูณนี้แหละที่ช่วยทำให้เราหายอดรวมแยกประเภทต่อไปได้

การใช้สูตร Array IF แบบหลายเงื่อนไข

กรณี Range ที่ใช้ตรวจสอบเงื่อนไขมาจากเรื่องเดียวกัน ให้ใช้บวก

G8		=SUMPRODUCT(((Id=G3)+(Id=H3)) *Amount)							
	A	B	C	D	E	F	G	H	I
1									
2		Num	Id	Name	Amount		Id	Id	
3		1	a001	a	10		a001	a002	
4		2	a002	b	20				
5		3	a001	c	30				
6		4	a004	d	40				
7		5	a002	e	50				
8									
9									
10									
11									
12									
13									

Id	Id
a001	a002

Amount
110
110
110

G6 : {=SUM(IF(Id=G3, Amount, IF(Id=H3,Amount,0)))}

G7 : {=SUM(((Id=G3)+(Id=H3)) *Amount)}

G8 : =SUMPRODUCT(((Id=G3)+(Id=H3)) *Amount)

ให้หาผลรวม Amount ของรหัส Id a001 กับ a002 (ทั้งนี้ที่เรียกว่า เป็นเรื่องเดียวกันเพราะเงื่อนไขเป็นเรื่องของรหัสเช่นกันทั้งคู่) โดยบันทึกรหัสที่ต้องการให้ใช้เป็นรหัสที่ค้นหาไว้ที่เซลล์ G3 และ H3 ตามลำดับ ซึ่งสามารถใช้สูตรหาค่าตอบเป็นยอดรวม 110 ได้หลายวิธี ดังนี้

1. ใช้สูตร {=SUM(IF(Id=G3, Amount, IF(Id=H3,Amount,0)))}

โดยต้องกดปุ่ม Ctrl+Shift+Enter เพื่อสร้างเครื่องหมายวงเล็บปีกกา { } ปิดหัวท้ายสูตร ถ้าแกะสูตร IF(Id=G3, Amount, IF(Id=H3,Amount,0)) โดยการคลิกลากหับส่วนของสูตรนี้แล้วกดปุ่ม F9 จะได้ Array {10;20;30;0;50} ซึ่งเปลี่ยนค่า Amount ของรหัสอื่นที่ไม่เท่ากับ a001 หรือ a002 ให้เท่ากับ 0 แทน

2. ใช้สูตร { =SUM(((Id=G3)+(Id=H3)) *Amount) }

โดยต้องกดปุ่ม Ctrl+Shift+Enter เพื่อสร้างเครื่องหมายวงเล็บปีกกา { } ปิดหัวท้ายสูตร ถ้าแกะสูตร ((Id=G3)+(Id=H3)) โดยการคลิกลากหับส่วนของสูตรนี้แล้วกดปุ่ม F9 จะได้ Array {1;1;1;0;1} โดยมีเลข 1 ตรงกับตำแหน่งของรหัสที่ต้องการ ถ้าแกะที่มาของเงื่อนไข (Id=G3) จะได้ Array ของ {TRUE;FALSE;TRUE;FALSE;FALSE} ที่นำมาบวกกับ (Id=H3) ซึ่งเป็น Array ของ {FALSE;TRUE;FALSE;FALSE;TRUE} ทำให้ได้ Array {1;1;1;0;1}

และเมื่อนำ {1;1;1;0;1} คูณกับ Array ของ Amount {10;20;30;40;50}
จะได้ Array {10;20;30;0;50} ซึ่งรวมค่าทั้งหมดได้เท่ากับ 110 ตามต้องการ

3. ใช้สูตร =SUMPRODUCT(((Id=G3)+(Id=H3)) *Amount)

โดยมีหลักการคำนวณแบบเดียวกับวิธีที่สอง แต่ไม่ต้องอาศัย { }

เพื่อช่วยให้มองเห็นลำดับการคำนวณได้ชัดเจนขึ้น ขอให้ดูภาพต่อไปนี้แล้วดูหลักการ
คำนวณทีละขั้นจาก column ซ้ายไปขวา

Id	id a001	id a002	id a001+a002	Amount	Amount
a001	1	0	1	10	10
a002	0	1	1	20	20
a001	1	0	1	30	30
a004	0	0	0	40	0
a002	0	1	1	50	50
					110

- Column ที่ 1 เป็นรหัส Id จะพบว่ามียี่ห้อ a001 กับ a002 อยู่ 4 รายการ
- Column ที่ 2 ตรวจสอบรหัส Id a001 จะพบว่ามียี่ห้อ a001 อยู่ 2 รายการ โดยถ้ามองตำแหน่งที่เป็นรหัสที่ต้องการเป็นเลข 1 และตำแหน่งที่ไม่ใช่เป็นเลข 0
จะได้เลข 1, 0, 1, 0, 0 ตามลำดับจากบนมาล่าง
- Column ที่ 3 ตรวจสอบรหัส Id a002 จะพบว่ามียี่ห้อ a002 อยู่ 2 รายการ โดยถ้ามองตำแหน่งที่เป็นรหัสที่ต้องการเป็นเลข 1 และตำแหน่งที่ไม่ใช่เป็นเลข 0
จะได้เลข 0, 1, 0, 0, 1 ตามลำดับจากบนมาล่าง
- Column ที่ 4 เป็นผลรวมเมื่อนำ Column ที่ 2 มารวมกับ Column ที่ 3
จะได้เลข 1, 1, 1, 0, 1 ตามลำดับจากบนมาล่าง แสดงตำแหน่งของรหัส a001 กับ a002 ณ ตำแหน่งที่ตรงกับเลข 1
- Column ที่ 5 เป็นตัวเลข Amount ทั้งหมด
- Column ที่ 6 เป็นผลคูณเมื่อนำ Column ที่ 4 มารวมกับ Column ที่ 5
จะได้เลข 10, 20, 30, 0, 50 ตามลำดับจากบนมาล่าง ซึ่งรวมเท่ากับ 110

หมายเหตุ สามารถใช้สูตร CountIF มาช่วยแทนการนำเงื่อนไขมาบวกกัน ซึ่งเป็นวิธีการใหม่ที่ผมเพิ่งค้นพบและนำมาสอนในหลักสูตร Excel Dynamic Reports

การใช้สูตร Array IF แบบหลายเงื่อนไข

กรณี Range ที่ใช้ตรวจสอบเงื่อนไขมาจากต่างเรื่องกัน ให้ใช้คุณ

G9		=SUMIFS(Amount,Id,G3,Name,H3)						
	A	B	C	D	E	F	G	H
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								

Num	Id	Name	Amount
1	a001	a	10
2	a002	b	20
3	a001	c	30
4	a004	d	40
5	a002	e	50

Id	Name
a001	c

Amount
30
30
30
30

G6 : {=SUM(IF(Id=G3, IF(Name=H3,Amount,0), 0))}

G7 : {=SUM((Id=G3)*(Name=H3) *Amount)}

G8 : =SUMPRODUCT((Id=G3)*(Name=H3) *Amount)

G9 : =SUMIFS(Amount,Id,G3,Name,H3)

G6 : {=SUM(IF(Id=G3, IF(Name=H3,Amount,0), 0))}

G7 : {=SUM((Id=G3)*(Name=H3) *Amount)}

G8 : =SUMPRODUCT((Id=G3)*(Name=H3) *Amount)

G9 : =SUMIFS(Amount,Id,G3,Name,H3)

ให้หาผลรวม Amount ของรหัส Id a001 เฉพาะรายการที่มี Name เท่ากับ c (ทั้งนี้ที่เรียกว่าเป็นต่างเรื่องกัน เพราะเงื่อนไขรหัสต่างจากเงื่อนไขชื่อ) โดยบันทึกการให้ใช้เป็นการให้ใช้และชื่อที่ค้นหาไว้ที่เซลล์ G3 และ H3 ตามลำดับ ซึ่งสามารถใช้สูตรหาค่าตอบเป็นผลรวม 30 ได้หลายวิธี ดังนี้

1. ใช้สูตร { =SUM(IF(Id=G3, IF(Name=H3,Amount,0), 0))}

โดยต้องกดปุ่ม Ctrl+Shift+Enter เพื่อสร้างเครื่องหมายวงเล็บปีกกา { } ปิดหัวท้ายสูตร ถ้าแกะสูตร IF(Id=G3, IF(Name=H3,Amount,0), 0) โดยการคลิกลากหับส่วนของสูตรนี้แล้วกดปุ่ม F9 จะได้ Array {0;0;30;0;0} ซึ่งเปลี่ยนค่า Amount ของรหัสอื่นที่ไม่เท่ากับ a001 และชื่อไม่ใช่ c ให้เท่ากับ 0 แทน

2. ใช้สูตร { =SUM((Id=G3)*(Name=H3) *Amount) }

โดยต้องกดปุ่ม Ctrl+Shift+Enter เพื่อสร้างเครื่องหมายวงเล็บปีกกา { } ปิดหัวท้ายสูตร ถ้าแกะสูตร (Id=G3)*(Name=H3) โดยการคลิกลากหับส่วนของสูตรนี้แล้วกดปุ่ม F9 จะได้ Array {0;0;1;0;0} โดยมีเลข 1 ตรงกับตำแหน่งรายการที่ต้องการ

ถ้าแถวที่มากของเงื่อนไข (Id=G3) จะได้ Array ของ
 {TRUE;FALSE;TRUE;FALSE;FALSE} นำมาคูณกับ (Name=H3) ซึ่งเป็น Array ของ
 {FALSE;FALSE;TRUE;FALSE;FALSE} ทำให้ได้ Array
 {0;0;1;0;0}

และเมื่อนำ {0;0;1;0;0} คูณกับ Array ของ Amount {10;20;30;40;50}
 จะได้ Array {0;0;30;0;0} ซึ่งรวมค่าทั้งหมดได้เท่ากับ 30 ตามต้องการ

3. ใช้สูตร =SUMPRODUCT((Id=G3)*(Name=H3) *Amount)

โดยมีหลักการคำนวณแบบเดียวกับวิธีที่สอง แต่ไม่ต้องอาศัย { }

4. ใช้สูตร =SUMIFS(Amount,Id,G3,Name,H3) เป็นสูตรที่เกิดขึ้นใน Excel 2007 เป็นต้นมา

เพื่อช่วยทำให้มองเห็นลำดับการคำนวณได้ชัดเจนขึ้น ขอให้ดูภาพต่อไปนี้แล้วดูหลักการ
 คำนวณทีละขั้นจาก column ซ้ายไปขวา

Id	Id a001	Name	Name c	a001 * c	Amount	Amount
a001	1	a	0	0	10	0
a002	0	b	0	0	20	0
a001	1	c	1	1	30	30
a004	0	d	0	0	40	0
a002	0	e	0	0	50	0
						30

- Column ที่ 1 เป็นรหัส Id จะพบว่ามียี่ห้อ a001 อยู่ 2 รายการ
- Column ที่ 2 ตรวจสอบรหัส Id a001 จะพบว่ามียี่ห้อ a001 อยู่ 2 รายการ โดยถ้ามองตำแหน่งที่เป็นรหัสที่ต้องการเป็นเลข 1 และตำแหน่งที่ไม่ใช่เป็นเลข 0 จะได้เลข 1, 0, 1, 0, 0 ตามลำดับจากบนลงมาล่าง
- Column ที่ 3 เป็นชื่อ Name จะพบว่ามียี่ห้อ c อยู่ 1 รายการ
- Column ที่ 4 ตรวจสอบชื่อ Name c จะพบว่ามียี่ห้อ Name c อยู่ 1 รายการ โดยถ้ามองตำแหน่งที่เป็นรหัสที่ต้องการเป็นเลข 1 และตำแหน่งที่ไม่ใช่เป็นเลข 0 จะได้เลข 0, 0, 1, 0, 0 ตามลำดับจากบนลงมาล่าง
- Column ที่ 5 เป็นผลคูณเมื่อนำ Column ที่ 2 มาคูณกับ Column ที่ 4 จะได้เลข 0, 0, 1, 0, 0 ตามลำดับจากบนลงมาล่าง แสดงตำแหน่งของรหัส a001 ที่มีชื่อ Name c ณ ตำแหน่งที่ตรงกับเลข 1

- Column ที่ 6 เป็นตัวเลข Amount ทั้งหมด
- Column ที่ 7 เป็นผลคูณเมื่อนำ Column ที่ 5 มาคูณกับ Column ที่ 6 จะได้เลข 0, 0, 30, 0, 0 ตามลำดับจากบนมาล่าง ซึ่งรวมเท่ากับ 30

วิธีพลิกแพลงสูตร SumProduct

สูตร SumProduct เป็นสูตรที่ทำงานแบบ Array อยู่ในตัว มีหน้าที่นำ Range ของตัวเลขแต่ละตำแหน่งจากตารางที่มีขนาดเท่ากัน มาคูณกันทีละเซลล์ตามตำแหน่งที่ตรงกัน จากนั้นจึงบวกรวมผลคูณที่ได้นั้นเข้าด้วยกัน

ตัวอย่าง เช่น เราต้องการหามูลค่าต้นทุนรวมของสินค้า โดยนำต้นทุนต่อหน่วยมาคูณกับจำนวนหน่วย ซึ่งหากไม่รู้จักสูตร SumProduct ก็ต้องนำ Cost*Quantity ทีละรายการ เพื่อให้ได้ตัวเลข Total ตาม Column D จากนั้นจึงสร้างสูตรในเซลล์ F2 =SUM(D3:D5) จึงจะได้ยอดรวมต้นทุนทั้งหมด

F4		=SUMPRODUCT(Cost, Quantity)					
A	B	C	D	E	F	G	H
1							
2		Cost	Quantity	Total		154	
3		11	1	11		F2 : =SUM(D3:D5)	
4		22	2	44		154	
5		33	3	99		F4 : =SUMPRODUCT(Cost, Quantity)	
6						154	
7						F6 : =SUMPRODUCT(Cost*Quantity)	

ซึ่งแทนที่จะต้องเสียพื้นที่คำนวณหา Total ใน Column D เราสามารถใช้สูตร SumProduct ลัดหาต้นทุนรวมได้โดยใช้สูตรตามนี้

1. **=SUMPRODUCT(Cost, Quantity)** เป็นการใส่สูตรตามโครงสร้างปกติ โดยใช้เครื่องหมาย comma , คั่นแต่ละ Range ในสูตร หรือ
2. **=SUMPRODUCT(Cost*Quantity)** เป็นสูตรที่นำแต่ละ Range มาคูณกันเลย ซึ่งการสร้างแบบคูณกันนี้ จะช่วยให้เราสามารถหลีกเลี่ยงการคำนวณในวงเล็บแล้วกดปุ่ม F9 เพื่อเห็นผลการคูณกันของแต่ละตำแหน่งได้ด้วย

จากนั้น ขอย้อนกลับไปเรื่องโครงสร้างภายในสูตร Array IF ตามที่อธิบายไว้แล้วว่า โครงสร้างของสูตร Array IF ประกอบด้วย 2 ส่วนใหญ่ๆ ซึ่งโดยทั่วไปทั้ง 2 ส่วนต้องมีขนาดความสูงความกว้างของ Array เท่ากัน กล่าวคือ

1. ส่วนของ Array ที่ทำหน้าที่คืนค่า True หรือ False
2. ส่วนของ Array ที่เป็นตัวเลขหรือค่าคำตอบเรื่องที่ต้องการ

ขอนำสูตร **=SUMPRODUCT((Id=G3)*(Name=H3) *Amount)** มาวิเคราะห์แยกโครงสร้างข้างต้น จะพบว่า

1. ส่วนของ Array ที่ทำหน้าที่คืนค่า True หรือ False ได้แก่ ส่วนของ (Id=G3) และ (Name=H3) ซึ่งนำมาคูณกันเพื่อให้เปลี่ยน True เป็นเลข 1 และเปลี่ยน False เป็นเลข 0
2. ส่วนของ Array ที่เป็นตัวเลขหรือค่าคำตอบเรื่องที่ต้องการ ได้แก่ ส่วนของ Amount

ดังนั้นหากต้องการปรับสูตรให้คำนวณนับแทนการหาผลรวมของ Amount โดยให้นับว่ามีค่าตามเงื่อนไขทั้งสี่รายการ ให้ใช้สูตรใดสูตรหนึ่งต่อไปนี้ก็ได้

1. **=SUMPRODUCT((Id=G3)*(Name=H3))** โดยสูตร SumProduct จะทำหน้าที่รวมเลข 1 ที่ได้จากการคูณกันของ Array ภายในวงเล็บว่ามีเลข 1 กี่ตัว นั่นคือนับจำนวนรายการที่มีเงื่อนไขตามต้องการนั่นเอง
2. **=SUMPRODUCT((Id=G3)*1, (Name=H3)*1)**
3. **=SUMPRODUCT((Id=G3)/1, (Name=H3)/1)**
4. **=SUMPRODUCT((Id=G3)+0, (Name=H3)+0)**
5. **=SUMPRODUCT((Id=G3)-0, (Name=H3)-0)**
6. **=SUMPRODUCT(--(Id=G3), --(Name=H3))**

สูตร SumProduct แบบที่ 2-6 เป็นสูตรที่ใช้เครื่องหมาย comma , คั่น จึงจำเป็นต้องกระตุ้นให้เปลี่ยน True เป็นเลข 1 และเปลี่ยน False เป็นเลข 0 โดยนำ True หรือ False ที่ได้จากการตรวจสอบเงื่อนไขไป *1, /1, +0, -0, หรือใส่เครื่องหมายลบลงไว้ข้างหน้าก่อน

ดังนั้นจึงขอแนะนำให้สร้างสูตร SumProduct แบบนำเงื่อนไขมาคำนวณ(คูณหรือบวก)กันเองก่อน เพราะนอกจากจะแกะหาผลจากการคูณหรือบวกกันจากส่วนของเงื่อนไขที่นำมาคำนวณกันได้

แล้ว ยังเป็นสูตรที่สั้นกว่าและมีขั้นตอนการคำนวณน้อยกว่าสูตร SumProduct ที่ใช้แบบเครื่องหมาย comma , คั่นอยู่ภายใน

หมายเหตุ

โดยทั่วไปหากเงื่อนไขที่ใช้ในสูตร Array IF เป็นการตรวจสอบว่าเท่ากันหรือไม่ โดยใช้เครื่องหมาย = ในการเปรียบเทียบว่าเป็น True หรือ False เราสามารถท่องจำได้เลยดังนี้

- ถ้า Range ที่นำมาตรวจสอบเงื่อนไขเป็น **เรื่องเดียวกัน** ให้นำผลลัพธ์จากการตรวจสอบเงื่อนไขมา **บวกกัน** และมีวงเล็บเปิดด้านหน้า 3 ตัว
- ถ้าถ้า Range ที่นำมาตรวจสอบเงื่อนไขเป็น **ต่างเรื่องกัน** ให้นำผลลัพธ์จากการตรวจสอบเงื่อนไขมา **คูณกัน** และมีวงเล็บเปิดด้านหน้า 2 ตัว

แต่ถ้าเป็นการตรวจสอบเงื่อนไขประเภทมากกว่าหรือน้อยกว่าหรือผสมกัน แม้ Range ที่นำมาเป็นเงื่อนไขจะเป็นเรื่องเดียวกัน ก็ไม่จำเป็นว่าต้องนำผลลัพธ์จากการตรวจสอบเงื่อนไขมาบวกกันเสมอไป บางกรณีอาจใช้คูณกันก็ได้ ทั้งนี้ขึ้นกับลักษณะการเปรียบเทียบที่ใช้ด้วย โดยขอให้คิดหาวิธีใดก็ได้ที่จะทำให้เกิดเลข 1 ขึ้นเฉพาะตำแหน่งรายการที่ต้องการให้ได้

ของจริง ของปลอม ของใช้ไม่ได้ใน Excel

ในเรื่องความซื่อสัตย์สุจริต โปรแกรม Excel ถือว่าได้หนึ่งร้อยคะแนนเต็ม ไม่ว่าคุณจะบันทึกค่าสร้างสูตร หรือนำ Excel มาใช้แบบใด อะไรที่คุณทำลงไป Excel ก็จะได้รับผลตามนั้น แต่ถ้าคุณเข้าใจ Excel ได้ไม่ดีพอ ก็อาจจะตีโพยตีพายเมื่อเห็นว่า Excel ทรมาน เพราะมันให้คำตอบที่ผิดพลาดต่างจากคำตอบที่คุณต้องการ ทั้งๆที่แท้จริงแล้วคุณนั้นแหละที่เข้าใจผิด

ของจริงที่ทำให้สูตร IF กลายเป็นของปลอม

เนื้อหาที่จะนำมาอธิบายต่อไปนี้จะชี้เป็นชี้ตายให้กับคนที่ชอบใช้สูตร IF หรือสูตรใดก็ตามที่เกี่ยวข้องกับการตัดสินใจหรือใช้ในการค้นหาข้อมูลว่า แฟ้มงานที่คุณสร้างขึ้นนั้นมีความน่าเชื่อถือหรือไม่ เพื่อพิสูจน์ให้เห็นกับตา ลองดูสูตร IF ต่อไปนี้แล้วคิดว่าจะได้คำตอบออกมาเป็นคำว่า เท่ากัน หรือ ไม่เท่ากัน

=IF(22.3-22.2=0.1, "เท่ากัน", "ไม่เท่ากัน")

เชื่อหรือไม่!!! Excel จะให้คำตอบออกมาว่า 22.3-22.2 นั้นไม่เท่ากับ 0.1 โดยคืนค่าออกมาเป็นคำว่า ไม่เท่ากัน ซึ่งถือเป็นคำตอบที่ถูกต้อง เพราะ 22.3-22.2 ไม่ได้คำนวณออกมาแล้วให้ผลลัพธ์เท่ากับ 0.1 ตามที่เราเข้าใจ

ขอให้ลองสร้างสูตร =22.3-22.2 ลงไปในเซลล์ แล้วกดปุ่ม F2 ตามด้วยปุ่ม F9 จะพบว่าค่าที่แท้จริงที่ Excel คำนวณได้คือ 0.1000000000000001 ซึ่งแน่นอนว่าไม่ใช่ 0.1

บางคนมองค่า 0.1000000000000001 แล้วยอมรับตัวเลขที่ได้นี้ เพราะเห็นว่ามันต่างจาก 0.1 ที่ต้องการเพียงเล็กน้อย ถ้าใครคิดเช่นนี้แสดงว่าเข้าใจ Excel ผิด เพราะ Excel จะไม่ยอมรับว่าค่า 0.1 เท่ากับ 0.1000000000000001 ไม่ว่าค่าที่แตกต่างกันนั้นจะน้อยมากก็ตาม

การที่ Excel คำนวณได้ค่าที่เพี้ยนต่างจากที่ควรนี้ ไม่ได้ถือว่าการคำนวณที่ผิดพลาด และไม่ได้มีแค่กรณีนี้กรณีเดียว สูตรหรือการคำนวณใดๆที่คุณใช้มีโอกาสคืนค่าเพี้ยนไปจากที่ต้องการได้ทั้งนั้น

Excel จะใช้รูปแบบ (Format Number) แสดงค่าในเซลล์ตามลักษณะตัวเลขที่พิมพ์ นี่คือสาเหตุประการแรกที่อธิบายว่า ทำไม Excel จึงไม่แสดงค่า 0.100000000000001 ในเซลล์ออกมาให้เห็นครบทุกหลัก เนื่องจากตัวเลข 22.3-22.2 ที่พิมพ์ลงไปมีทศนิยมเพียงหนึ่งหลัก ดังนั้น Excel จึงแสดงผลลัพธ์ให้เห็นตัวเลขที่มีทศนิยมเพียงหนึ่งหลักตามไปด้วย

ค่าที่เพี้ยนไปไม่ได้ถือว่าการคำนวณที่ผิดพลาด แต่เป็นเพราะ Excel ใช้หลักการคำนวณแบบเลขฐานสองแล้วแปลงกลับมาเป็นเลขฐานสิบ ซึ่งค่า 22.3-22.2 คำนวณแบบเลขฐานสองได้คำตอบเป็นเลขที่ซ้ำไม่รู้จบ (Repeating Binary Number) จากนั้นเมื่อแปลงกลับมาเป็นเลขฐานสิบให้พวกเราเข้าใจ จึงต้องปรับค่าให้เหลือเพียง 15 หลัก (Excel มีความละเอียด Precision โดยรับตัวเลขได้สูงสุด 15 หลัก) แล้วส่งผลให้ได้ผลลัพธ์แตกต่างจากที่ควรบ้างเล็กน้อย และมีได้เกิดขึ้นเฉพาะกรณี 22.3-22.2 เท่านั้น เราไม่มีทางคาดการณ์ได้เลยว่าจะเพี้ยนเมื่อใดและเกิดจากการคำนวณใด

ดังนั้นก่อนที่จะใช้สูตรที่เกี่ยวข้องกับการตัดสินใจหรือสูตรในการค้นหาข้อมูล เช่น IF, SumIF, VLookup, หรือ Match จำเป็นต้องปรับค่าให้เท่ากับค่าที่ต้องการ เพื่อให้เป็นไปตามเงื่อนไขที่ต้องการก่อนเสมอ โดยใช้สูตรอื่นเข้ามาช่วยปรับค่า จากนั้นเมื่อได้ค่าตามที่ต้องการจริงๆแล้วคุณก็จะกำหนดรูปแบบ Format ใดๆให้กับค่านั้นก็ตามใจ (ห้ามใช้แต่เพียงการกำหนดรูปแบบเท่านั้นเพราะมันไม่ได้ช่วยปรับค่าที่แท้จริงแต่อย่างใด)

สูตรสำคัญซึ่งใช้ในการปรับค่าตัวเลขให้เป็นไปตามต้องการ (โปรดศึกษารายละเอียดของสูตรได้จาก Excel Help) ได้แก่

1. สูตร =Round(ตัวเลข, จำนวนหลัก) ใช้สำหรับการปัดค่า

=Round(123.45,0) คื้นค่า 123

=Round(123.45,1) คื้นค่า 123.5

=Round(123.45,-1) คื้นค่า 120

=Round(123.45,-2) คื้นค่า 100

2. สูตร =Trunc(ตัวเลข, จำนวนหลัก) ใช้สำหรับการตัดค่า

=Trunc(123.45,0) คื้นค่า 123

=Trunc(123.45,1) คื้นค่า 123.4 ต่างจาก Round

=Trunc(123.45,-1) คัดค่า 120

=Trunc(123.45,-2) คัดค่า 100

3. สูตร =Int(ตัวเลข) ใช้สำหรับปรับเป็นตัวเลขจำนวนเต็ม(ที่น้อยกว่าค่าเดิม)

=Int(1.23) คัดค่า 1

=Int(-1.23) คัดค่า -2

4. สูตร =Mod(เลขตั้งตั้ง, เลขตัวหาร) ใช้สำหรับหาเศษที่เหลือจากการหาร

=Mod(7,2) คัดค่า 1 เพราะ 7/2 เทียบเท่ากับ 3 1/2

=Mod(7,4) คัดค่า 3 เพราะ 7/4 เทียบเท่ากับ 1 3/4

=Mod(123.45,1) คัดค่าเท่ากับ .45 !!!

วันที่จอมปลอม

	C2		f _x	15/1/2010
	A	B	C	D
1				
2		Jan	Jan	
3		Feb	Feb	
4		Mar	Mar	
5		Apr	Apr	
6		May	May	
7		Jun		

ในหน้ารายงานของทุกคน ต้องมีข้อมูลเกี่ยวกับวันเดือนปีของรายงานนั้นกำกับอยู่ด้วยเสมอ ขอให้เปิดแฟ้มของคุณขึ้นมาดูซิว่าในเซลล์ที่มีข้อมูลแสดงวันเดือนปี ถ้าไม่ได้กำหนดรูปแบบให้ขีดซ้ายหรือขีดขวา ข้อมูลวันเดือนปีของคุณขีดซ้ายของเซลล์ตามรูปข้างบนนี้เหมือนใน Column B หรือขีดขวาของเซลล์เหมือนใน Column C

Column B เป็นชื่อเดือนที่ขีดซ้ายของเซลล์ ไม่ว่าคุณจะบันทึกไว้เป็นเพียงเลขที่วัน หรือเฉพาะเลขที่เดือน หรือชื่อเดือนเท่านั้นตามแบบที่เห็นในภาพนี้ ถ้ามันขีดซ้ายเองโดยคุณไม่ได้

กำหนดให้มันขีดซ้าย แสดงว่าข้อมูลนั้นมีค่าเป็น Text ซึ่งไม่มีประโยชน์ในการนำข้อมูลไปใช้คำนวณต่อแม้แต่น้อย อย่างเช่นเซลล์ B2 ที่มีคำว่า Jan ไม่ได้ให้รายละเอียดว่าเป็นวันที่ใดของเดือนมกราคมหรือเป็นเดือนมกราคมของปีใด คำว่า Jan ที่ขีดซ้ายให้ความหมายแต่เพียงบอกว่า เป็นเดือนมกราคมเท่านั้น ซึ่งหลายคนแก้ปัญหาโดยการเพิ่มเซลล์บันทึกเลขปีลงไปอีก

Column C เป็นการบันทึกแบบที่ถูกต้อง สังเกตว่าชื่อเดือนขีดขวาของเซลล์ และแม่ในเซลล์ C2 แสดงคำว่า Jan ก็ตาม แต่เมื่อมองที่ช่อง Formula Bar ด้านบน จะพบว่าค่าที่แท้จริงเป็น 15/1/2010 ซึ่งหากต้องการให้แสดงชื่อเดือนตั้งแต่เซลล์ C3 ต่อๆกันไป ให้สร้างสูตร =C2+30 ลงไปในเซลล์ C3 แล้ว Copy สูตรนี้ต่อไปในแนวตั้ง จากนั้นให้กำหนด Format Cells ใน Column C เป็น **[\$-409]mmm** หรือ **[\$-en-US]mmm** จะทำให้ค่า 15/1/2010, 14/2/2010, 16/3/2010 แสดงเฉพาะชื่อเดือน Jan, Feb, Mar ที่ขีดขวาของเซลล์

สาเหตุที่ใช้ค่าแรกในเซลล์ C2 เป็นวันที่ 15 นั้นเพื่อช่วยให้เราสามารถบวกเพิ่มต่อไปเซลล์ละ 30 วัน เพื่อให้ได้เดือนต่อไปเพิ่มทีละเดือนได้ง่ายกว่าที่จะเริ่มเซลล์แรกเป็นวันที่สิ้นเดือน ซึ่งต้องบวกเพิ่มด้วยจำนวน 31 วัน 30 วัน 28-29 วันแตกต่างกันไปในแต่ละเดือน

หลักการใช้วันที่และเวลา

1. ในการบันทึก ให้พิมพ์ให้ครบทั้งวันเดือนปี อย่าบันทึกเฉพาะวันที่หรือเฉพาะเดือนหรือเฉพาะปีเท่านั้น โดยจะพิมพ์ส่วนของเวลาต่อท้ายด้วยหรือไม่ก็ได้ เช่น 14/2/2010 12:00
2. ให้บันทึกโครงสร้างของวันที่ลงไปในเซลล์ตามแบบที่กำหนดไว้ใน Regional Setting ของ Windows เช่น ถ้ากำหนดไว้ใน Regional Setting เป็น Thai ก็ต้องบันทึกวันที่ตามแบบของประเทศไทยซึ่งใช้ลำดับตามวันก่อนเดือนก่อนปี แต่ถ้ากำหนดใน Regional Setting เป็น USA ก็ต้องบันทึกตามลำดับเดือนก่อนวันก่อนปี
3. แม้จะใช้ Regional Setting เป็น Thai แต่ในการบันทึกในส่วนของเลขปีต้องใช้ปีค.ศ. ในการบันทึกเท่านั้น จากนั้นให้ใช้ Format ปรับการแสดงผลปีค.ศ. ให้เป็นปีพ.ศ. ในภายหลัง เช่น ให้บันทึก 14/2/2010 (ห้ามใช้ปีพ.ศ. แทนอย่างเด็ดขาด) จากนั้นเมื่อต้องการแสดงเป็น 14/2/2553 ให้ใช้ Format [\$-1070000]d/mm/yyyy
4. ในโครงสร้างของวันที่ให้พิมพ์เครื่องหมาย / ในการแบ่งส่วนของ วัน/เดือน/ปีค.ศ. และในโครงสร้างของเวลาให้พิมพ์เครื่องหมาย : ในการแบ่งส่วนของ ชั่วโมง:นาที:วินาที
5. ควรพิมพ์เลขปีค.ศ. ให้ครบทั้ง 4 หลักเสมอ แต่ถ้าจำเป็นต้องพิมพ์แค่สองหลักท้าย พอกดปุ่ม Enter เพื่อบันทึกค่าลงไป Excel จะเปลี่ยนตัวเลขปี 2 หลักท้าย ดังนี้
 - 5.1. ตั้งแต่เลข 00-29 ให้เป็นช่วงปีค.ศ. 2000-2029
 - 5.2. ตั้งแต่เลข 30-99 ให้ย้อนกลับเป็นช่วงปีค.ศ. 1930-1999
6. เมื่อบันทึกวันที่แล้วต้องขีดขวาของเซลล์เสมอ (สาเหตุที่ขีดขวาเพราะมีค่าเป็นตัวเลข) และขอให้หลีกเลี่ยงการใช้วันที่แบบ Text ซึ่งขีดซ้ายของเซลล์เนื่องจาก Excel จะไม่นำ Regional Setting มาช่วยควบคุมในการแบ่งส่วนของวันเดือนปีที่ใช้แบบ Text
7. ค่าของวันที่และเวลามีชื่อเรียกว่า Date Serial Number หรือเรียกว่า Serial Number (SN) โดย Excel ถือว่า 1/1/1900 0:00:00 มีค่า SN=1 ซึ่งเราสามารถแกะดูค่า SN ได้โดยเปลี่ยน Format เป็น General (โดยการกดปุ่ม Ctrl+Shift+ ~) เช่น 14/2/2010 12:00:00 มีค่า SN เท่ากับ 40223.5
 - 7.1. ในส่วนของตัวเลข SN ส่วนที่เป็นจำนวนเต็ม คือ ค่าของวันที่
 - 7.2. ในส่วนของตัวเลข SN ส่วนที่เป็นเศษทศนิยม คือ ค่าของเวลา
8. เวลาของ Excel เริ่มจาก 0:00:00-23:59:59 และใช้ Format h:mm:ss
9. ระยะเวลา เริ่มจาก 0:00:00 ขึ้นไป โดยไม่สิ้นสุดที่ 23:59:59
 - 9.1. ใช้ Format [h]:mm:ss เพื่อแสดงเลขตั้งแต่ 24 ชั่วโมงขึ้นไปได้

- 9.2. ใช้ Format [mm]:ss เพื่อแสดงเลขตั้งแต่ 60 นาทีขึ้นไปได้
- 9.3. ใช้ Format [ss] เพื่อแสดงเลขตั้งแต่ 60 วินาทีขึ้นไปได้
10. ในการแสดงเวลาหรือระยะเวลา ถ้าค่า SN < 1 จะใช้ Format ของเวลาหรือระยะเวลาก็ได้ แต่ถ้า SN >=1 ต้องเลือกใช้ Format ของระยะเวลาเท่านั้น
11. ถ้าต้องการบันทึกการทำงานข้ามคืน ไม่ควรบันทึกเฉพาะเวลาเข้าออกงาน แต่ให้บันทึกวันเดือนปีกำกับเวลาไว้ด้วย เช่น 14/2/2010 20:00 เพื่อให้ Excel รับรู้ค่าเป็น SN ที่มีครบทั้งส่วนของวันที่และเวลา
12. Format ในการแสดงวันและเวลา
- 12.1. แสดงเลขวันที่ ใช้ d หรือ dd
 - 12.2. แสดงเป็นชื่อวัน แบบย่อใช้ ddd หรือแบบเต็ม dddd
 - 12.3. แสดงเป็นเลขเดือน ใช้ m หรือ mm
 - 12.4. แสดงเป็นชื่อเดือน แบบย่อใช้ mmm หรือแบบเต็ม mmmm
 - 12.5. แสดงเป็นเลขปี ใช้ yy หรือ yyyy
 - 12.6. แสดงชั่วโมง นาที วินาที ใช้ h:mm:ss หรือ hh:mm:ss
 - 12.7. รหัสควบคุม Locale ใช้ [\$-409] หรือ [\$-en-US]mmm สำหรับ USA และ [\$-1070000] หรือ [\$-,107 สำหรับไทย

การคำนวณวันที่และเวลา

ก่อนที่จะนำวันที่และเวลามาคำนวณได้ ต้องเริ่มจากการตรวจสอบว่าข้อมูลเกี่ยวข้องกับวันที่และเวลาซึ่งถูกบันทึกไว้ว่ามีโครงสร้างที่ถูกต้องตรงกับ Regional Setting หรือไม่ หากบันทึกไว้ผิด เช่น ใน Regional Setting กำหนดไว้เป็น USA แต่คนที่บันทึกข้อมูลพิมพ์วันที่ไว้ในแบบประเทศไทย เช่น พิมพ์ 4/2/2001 ซึ่งต้องการบันทึกแบบไทยให้หมายถึงวันที่ 4 เดือน 2 ปี 2001 จะพบว่าเมื่อนำแฟมนี้มาเปิดบนเครื่องที่มี Regional Setting เป็นไทย เลขที่ของวันจะสลับกับเลขที่ของเดือน จะแสดงเป็น 2/4/2001 แทน ทำให้กำหนดเวลาที่บันทึกไว้ผิดทั้งหมด ถ้าไม่เก่งสูตรเกี่ยวข้องกับวันที่และเวลาก็ต้องจัดการพิมพ์ทับใหม่ทั้งหมด

นอกจากนี้ต้องตรวจสอบต่อไปอีกว่า ค่าของวันที่ซึ่งบันทึกไว้เป็นข้อมูลที่มีค่าเป็นตัวเลข (Date Serial Number หรือ SN) หรือไม่ โดยเริ่มจากยกเลิกการจัดชิดซ้ายชิดขวาของเซลล์วันที่ทั้งหมด หากพบว่า ชิดขวาก็ใช้ได้ แต่ถ้าพบว่าชิดซ้ายแสดงว่ามีค่าเป็น Text ซึ่งไม่สามารถนำมาคำนวณต่อได้ในทันที จำเป็นต้องอาศัยสูตร Left, Right, Mid แยกตัวเลขแต่ละส่วนที่เป็นวันเดือนปีออกจากกันแล้วใช้สูตรวันที่และเวลามาช่วยแก้ไขให้มีค่าเป็น SN

หากคุณอยากจะเก่งสูตรคำนวณเรื่องวันที่และเวลา ต้องรู้จักสูตรหา SN หรือสูตรที่สามารถแปลงเลขที่ของวันเดือนปี ที่มนุษย์เข้าใจไปเป็นค่า SN ที่ Excel รู้จัก เช่น สูตร Now(), Today(), Date(Year,Month,Day), Time(Hour,Minute,Second)

จากนั้นต้องสามารถใช้สูตรแปลงค่า SN กลับมาเป็นเลขที่ของวันเดือนปี เช่น สูตร Day(SN), Month(SN), Year(SN), และ WeekDay(SN) หรือหาเลขที่ของเวลา เช่น Hour(SN), Minute(SN), Second(SN)

สูตรแปลงเลขที่ของวันเดือนปีและเวลาที่มนุษย์เข้าใจ ไปเป็นค่า Serial Number

เพื่อช่วยให้เข้าใจสูตรได้ง่ายขึ้น ขอสมมติว่าปัจจุบัน คือ วันที่ 14 เดือนกุมภาพันธ์ ปีค.ศ. 2010 เวลา 12 นาฬิกา 30 นาที 45 วินาที

1. **=NOW()** จะได้วันเดือนปีและเวลาปัจจุบัน เช่น 14/2/2010 12:30:45
2. **=Today()** จะได้เฉพาะวันเดือนปีปัจจุบัน เช่น 14/2/2010
3. **=Date(2010,2,14)** จะได้ 14/2/2010
4. **=Date(Year(Today()), Month(Today())+1, 0)** จะได้วันเดือนปีของวันสุดท้ายของเดือนปัจจุบัน คือ 28/2/2010 (หรือจะใช้สูตร =EOMonth(Today(),0))
5. **=Time(12,30,45)** จะได้เวลา 12:30:45
6. **=Time(12,30,45)+1** จะได้ระยะเวลา 36:30:45 ซึ่งต้องใช้ Format [h]:mm:ss ด้วย

สูตรแปลงค่า Serial Number

กลับมาเป็นเลขที่ของวันเดือนปีและเวลาที่มนุษย์เข้าใจ

สมมติว่าเซลล์ A1 มีสูตร =NOW() ซึ่งแสดงออกมาเป็น 14/2/2010 12:30:45 (ถ้าต้องการแสดงค่าออกมาเป็น SN โดยการเปลี่ยน Format เป็น General จะพบว่า เซลล์ A1 มีค่า SN เท่ากับ 40223.5213541667 ซึ่งเป็นตัวเลขที่ Excel รู้จักแต่เราไม่รู้จักร)

1. **=Day(A1)** จะได้เลขวันที่ 14
2. **=Month(A1)** จะได้เลขเดือน 2
3. **=Year(A1)** จะได้เลขปี 2010

4. **=WeekDay(A1)** จะได้เลขของวันในสัปดาห์ 1=Sunday, 2=Tuesday,...7=Saturday
5. **=Hour(A1)** จะได้เลขชั่วโมง 12
6. **=Minute(A1)** จะได้เลขนาที่ 30
7. **=Second(A1)** จะได้เลขวินาที 45

หลักการกำหนด

Format Cells > Number

1. Excel ใช้เครื่องหมาย # และ 0 แทนตำแหน่งของตัวเลข ซึ่งถ้าใช้เครื่องหมาย # แล้วตัวเลขมีค่าไม่ถึงก็จะไม่แสดงเลขหลักนั้น แต่ถ้าใช้ 0 จะแสดงแทนด้วยเลข 0 ออกมาให้เห็น เช่น ถ้าพิมพ์ตัวเลข 1.2 ลงไป ถ้าใช้ Format ##.## จะแสดง 1.2 แต่ถ้าใช้ Format 00.00 จะแสดง 01.20 ด้วยเหตุนี้ใน Format มาตรฐานที่ Excel เตรียมไว้ จะพบว่าจะอย่างน้อยตัวเลขหลักหน่วยและหลักทศนิยม จึงกำหนดให้ใช้เลข 0 ไว้เสมอ เช่น #,##0.00
2. ด้านหน้าของ Format ที่เป็นเครื่องหมาย # หรือเลข 0 เราสามารถกำหนดสีของ Font ได้โดยพิมพ์ชื่อสีที่ต้องการไว้ในเครื่องหมาย [] เช่น [Red] หรือถ้าจำชื่อสีไม่ได้ ให้ใช้ [Colorn] แทน โดย n คือเลขของสีที่ต้องการ เช่น [Color12]
3. เครื่องหมายวงเล็บ [] ยังใช้ในแบบเงื่อนไขเพื่อควบคุมให้แสดงผลออกมาเฉพาะเมื่อตรงกับเงื่อนไขที่กำหนดไว้ในเครื่องหมาย [] เช่น [>=90]00000.00 จะทำให้ตัวเลขที่พิมพ์ลงไปไปในเซลล์ที่ใช้ Format นี้ เฉพาะเมื่อมีค่ามากกว่าหรือเท่ากับ 90 ให้แสดงตัวเลขในรูปแบบ 00000.00
4. ใน Format สามารถแทรกสัญลักษณ์หรือตัวอักษรไว้ด้านหน้า ด้านหลัง หรือระหว่าง Format ที่เป็นเครื่องหมาย # หรือเลข 0
 - 4.1. กรณีแทรกสัญลักษณ์ ให้พิมพ์แทรกได้โดยตรง เช่น (0 . 00) % มีเครื่องหมายวงเล็บ เครื่องหมายวรรค จุดทศนิยม และ % เป็นสัญลักษณ์
 - 4.2. กรณีแทรกตัวอักษร ต้องพิมพ์ตัวอักษรไว้ระหว่างเครื่องหมายคำพูด เช่น "Total" 0.00 "บาท"
5. ถ้าต้องการปิดตัวเลขให้เป็นหลักพัน ให้เติมเครื่องหมายคอมมาต่อท้ายหนึ่งตัว ถ้าต้องการปิดเป็นหลักล้าน ให้เติมคอมมาต่อท้ายสองตัว เช่น 0.00,, จะแสดงตัวเลข 123456789 ที่บันทึกลงไปออกมาเป็น 123.46 โดยค่าที่แท้จริงยังคงเท่ากับ 123456789 ตามเดิม
6. ให้ใช้เครื่องหมาย ; ได้สูงสุด 3 ตัว เพื่อควบคุม Format ให้เปลี่ยนตามค่าบวก ค่าลบ ค่าศูนย์ และค่าที่เป็นตัวอักษร ในโครงสร้างตามนี้ **ค่าบวก;ค่าลบ;ค่าศูนย์;ค่าที่เป็นตัวอักษร**
 - 6.1. ถ้าไม่ใส่เครื่องหมาย ; เลย แสดงว่าเป็น Format กลางที่ให้ตัวเลขทุกค่าใช้ Format นั้นร่วมกัน
 - 6.2. หลังจากเครื่องหมาย ; ที่เติมต่อท้ายลงไป ถ้าใส่ ; ตัวใด ต้องตามด้วย Format ของค่านั้นๆ แต่ถ้าเราไม่ได้กำหนด Format ของนั้นๆไว้ด้วย **จึงไม่แสดงค่านั้นๆ**

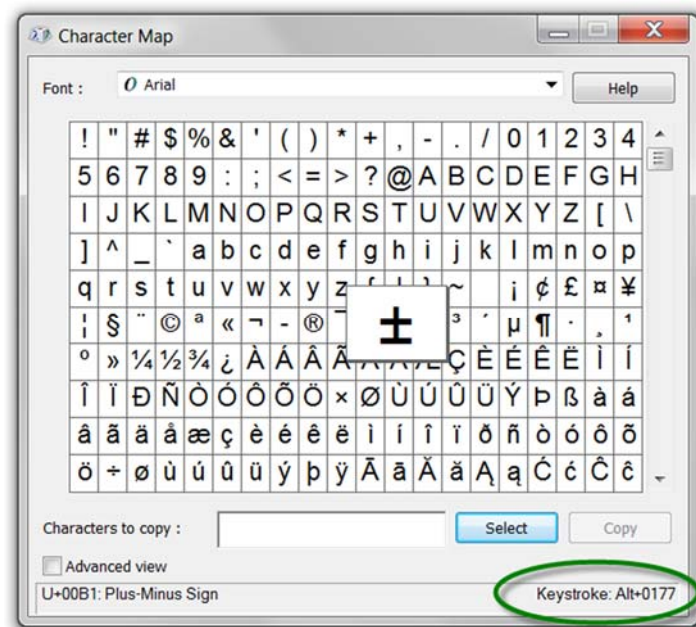
ตัวอย่างต่อไปนี้ สมมติว่าค่าที่พิมพ์ลงไปในเซลล์คือ 123, -123, 0, Hello

- Format ;;; จะไม่แสดงอะไรให้เห็นเลย
- Format 0.00 จะแสดง 123.00, -123.00, 0.00, Hello
- Format 0.00; จะแสดง 123.00, **ค่าลบไม่แสดง**, 0.00, Hello
- Format 0.00;[Red](0.00) จะแสดง 123.00, (123.00) ในสีแดง, 0.00, Hello
- Format 0.00;; จะแสดงเฉพาะค่าบวก 123.00 กับ Hello เท่านั้น **ค่าลบ และ 0 ไม่แสดง**
- Format 0.00;;; จะแสดงเฉพาะค่าบวก 123.00 เท่านั้น
- Format 0.00;(0.00);; จะแสดงเฉพาะค่าบวก 123.00 กับค่าลบ (123.00)
- Format "Yes";"No";"Reject"; จะแสดง Yes, No, Reject
- Format "Total" 0 "Baht". 00 "Satang";; จะแสดง Total 123 Baht. 00 Satang
- Format 0.00_);(0.00);; จะแสดง 123.00 ได้แนวตรงกับค่าลบ (123.00) โดยเครื่องหมาย _) ที่เติมต่อท้าย Format ค่าบวก หมายถึง ให้ห่างจากขอบขวาของเซลล์ เท่ากับความกว้างของเครื่องหมาย)

ตัวอย่าง Format แปลกๆ

- Format [>=90]"A";[>=70]"B";"C" จะเปลี่ยนตัวเลขในเซลล์ตามเงื่อนไขว่า ถ้าค่ามากกว่าหรือเท่ากับ 90 ให้แสดงตัว A แทนตัวเลขนั้น ถ้าค่ามากกว่าหรือเท่ากับ 70 ให้แสดงตัว B แทนตัวเลขนั้น หรือมิฉะนั้นให้แสดงตัว C
- Format [=1]"Yes";[=0]"No";"Reject" จะแสดงคำว่า Yes, No, Reject แทนตัวเลขที่มีค่าเท่ากับ 1, 0, หรือตัวเลขอื่น ตามลำดับ
- Format "4";"4";"4";"4" จะเปลี่ยนค่าให้เป็นเลข 4 แทนไม่ว่าค่าที่แท้จริงจะเป็นเท่าใด
- Format 0000 จะแสดงเลข 123 ออกมาเป็น 0123 (ซึ่งตัวเลขยังคงมีสถานะเป็นตัวเลขตามเดิม ถูกต้องกว่าการพิมพ์ '0123 หรือใช้คำสั่ง Format Cells > Number > Text ซึ่งทำให้ตัวเลขเปลี่ยนสถานะมาเป็น Text ซึ่งดูง่ายกว่าจะขีดซ้ายของเซลล์)
- หากต้องการแทรกสัญลักษณ์แปลกที่ไม่มีบนแป้นพิมพ์ลงไป ใน Format เช่น เครื่องหมายบวกลบ ± ให้กดปุ่ม Alt ค้างไว้แล้วพิมพ์ตัวเลข 0177 ซึ่งสามารถค้นหาตัว

เลขที่ต้องพิมพ์นี้ได้จาก โปรแกรม Character Map ซึ่งสั่งเปิดโปรแกรมนี้ได้ง่ายๆโดยพิมพ์คำว่า charmap ลงไปในช่องของคำสั่ง Start > Run



การใช้คำสั่ง Format Cells > Number จะช่วยปรับการแสดงตัวเลขให้มีตัวอักษรแทรก โดยค่าที่แท้จริงยังคงถือว่าเป็น Number ที่เห็นได้จากการขีดขวาของเซลล์ แต่ถ้าเซลล์มีความกว้างไม่พอจะเห็นเป็นเครื่องหมาย ##### แทน ซึ่งแก้ได้โดยสั่ง **Format Cells > Alignment > กางช่อง Shrink to fit** เพื่อให้ Excel ปรับขนาด Font ย่อลงให้แสดงให้เห็นได้ในเซลล์นั้นเสมอ

ประเด็นเรื่อง Format นี้ขอย้ำว่า การใช้ Format เป็นเพียงการเปลี่ยนสิ่งเห็นเท่านั้น มิได้แก้ไขค่าให้ต่างไปจากเดิมแม้แต่น้อย

หากต้องการนำตัวเลขไปแสดงร่วมกับตัวอักษร ยังมีอีกวิธีหนึ่ง โดยใช้สูตร Text เข้ามาช่วยปรับรูปแบบของตัวเลขแล้วนำไปเชื่อมต่อกับตัวอักษรที่ต้องการโดยใช้เครื่องหมาย & เป็นตัวเชื่อม

สมมติว่า เซลล์ A1 มีค่าเท่ากับ .15 หากต้องการนำเลข .15 ไปแสดงให้เป็นคำว่า Profit 15.0% ทำได้ 2 วิธี

- **วิธีแรก** ใช้คำสั่ง Format Cells > Number > Custom แล้วกำหนด Type เป็น "Profit" 0.0%
- **วิธีที่สอง** สร้างสูตร ="Profit "&Text(A1,"0.0%")

จะพบว่าผลจากวิธีแรกได้ค่าว่า Profit 15.0% ขีดขวาของเซลล์แสดงว่าสามารถนำไปคำนวณต่อได้ แต่วิธีที่สองแม้จะได้ค่าเดียวกันแต่จะขีดซ้ายของเซลล์ซึ่งไม่สามารถนำไปคำนวณต่อ จึงขอแนะนำให้ใช้วิธีที่สองกับผลลัพธ์สุดท้ายที่ไม่ต้องนำไปใช้คำนวณต่อเท่านั้น

นอกจากนี้สูตร Text ยังเปรียบได้กับสูตร Round โดยสูตร Text นี้จะปัดตัวเลขและแสดงตัวเลขตามรูปแบบให้ด้วย เช่น =TEXT(123456789,"0.00,") จะได้คำตอบเป็น 123.46 และมีค่า 123.46 ด้วย

วิธีใช้ Add-In

Add-In เป็นแฟ้มที่มีนามสกุล .xla หรือ .xlam ซึ่งแฟ้มเหล่านี้จะทำให้ Excel ที่ติดตั้งไว้ในแต่ละเครื่องมีสูตรหรือคำสั่งเพิ่มขึ้นมากกว่าเดิม ซึ่งทุกวันนี้เราสามารถนำแฟ้ม Add-in ที่แจกฟรีจากอินเทอร์เน็ตมาใช้งานได้ทันที

1. เริ่มจาก Download แฟ้ม Add-in มาเก็บไว้ในเครื่องคอมพิวเตอร์ของคุณก่อน
2. จากนั้นเปิดโปรแกรม Excel แล้วสั่ง **File > Options > Add-ins** แล้วคลิกที่ปุ่ม Go ด้านล่างของหน้าต่าง จะเปิดให้เห็นรายชื่อ Add-ins Available ที่มาพร้อมกับโปรแกรม Excel
3. กดปุ่ม Browse ค้นหาแฟ้ม Add-in ที่คุณเก็บไว้ในเครื่องตามข้อ 1 ให้พบแล้วกดปุ่ม OK
4. จะพบชื่อ Add-in ที่คุณเลือกปรากฏเพิ่มในช่องรายชื่อ Add-ins Available

ขอให้เลือกกาชื่อ Add-ins เฉพาะที่ต้องการนำมาใช้งานเท่านั้น เพราะการเปิดใช้ Add-in ก็เหมือนกับการเปิดแฟ้มทั่วไป เพียงแต่เมื่อเปิด Excel จะเปิดแฟ้ม Add-in ที่เลือกไว้ต่อให้เอง แต่จะไม่เห็นตัวแฟ้มแต่อย่างใด ซึ่ง Add-in จะทำให้คุณมีสูตรเพิ่มเติมหรือทำให้มีคำสั่งเพิ่มที่จะเห็นได้บนเมนูก็แล้วแต่แฟ้ม Add-in นั้นสร้างขึ้นมาจากจุดประสงค์ใด

ขอแนะนำให้ใช้ **Expert2000.xla** ซึ่งเป็น Add-in ที่แจกให้ใช้ในหลักสูตรสุดยอดเคล็ดลับ โดยจะทำให้คุณได้เมนูชื่อ Expert และมีคำสั่งเพิ่มขึ้น ทำหน้าที่เกี่ยวข้องกับการปรับแต่งหน้าจอเมื่อสร้างงานเสร็จแล้ว และยังทำให้มีสูตรต่อไปนี้เพิ่มเติม

- =Fml(cell) ทำหน้าที่แสดงสูตรในเซลล์
- =Fmt(cell) ทำหน้าที่แสดง Format ในเซลล์

ถ้าเรียกใช้ Add-in ชื่อ **Money.xla** จะได้สูตร =Money(ตัวเลข,"ชื่อหน่วยเงิน","ชื่อหน่วยเศษสตางค์") เช่น =Money(1234.56,"Dollar","Cent") จะอ่านตัวเลขออกมาเป็นคำว่า One Thousand Two Hundred Thirty Four Dollars and Fifty Six Cents

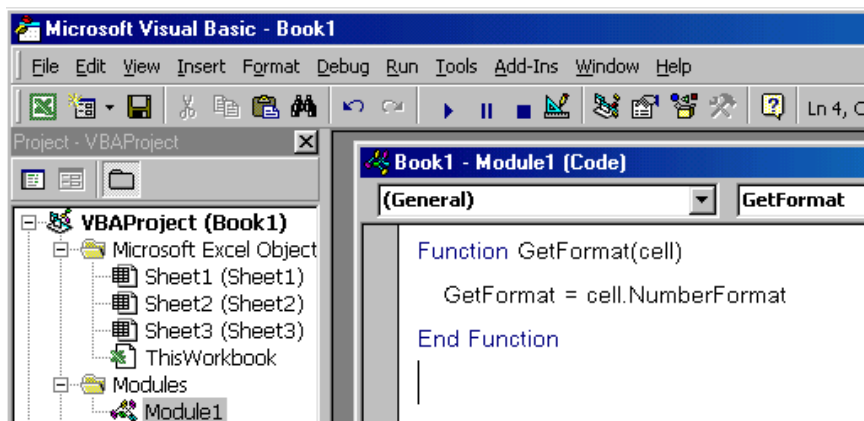
เนื่องจากสูตรเหล่านี้ไม่ใช่สูตรที่มาจากปกติของ Excel ดังนั้นหากเปิดแฟ้มที่ใช้สูตร Fml ขึ้นมา โดยที่ยังไม่ได้เรียกใช้ Expert2000.xla จะทำให้สูตรเหล่านี้กลายเป็น Error ว่า #NAME! ทั้งหมด ซึ่งแก้ไขได้โดยการติดตั้ง Add-in Expert2000.xla ก่อนแล้วจึงตามด้วยการเปิดแฟ้มที่มีสูตรเหล่านี้ต่อ แต่ถ้าพบว่าสูตรยังคง Error อยู่ ให้คลิกที่เซลล์สูตรแล้วกดปุ่ม F2 แล้วกดปุ่ม Enter เพื่อกระตุ้นให้สูตรเริ่มทำงานต่อไปได้ตามปกติ

Function VBA

แทนที่จะต้องสร้างสูตรที่มีขั้นตอนการคำนวณยาวเหยียดซ้ำแล้วซ้ำอีกลงไปตาราง
Function VBA เป็นทางออกที่จะช่วยทำให้เกิดสูตรลัดขึ้นมาใช้งานในแฟ้ม

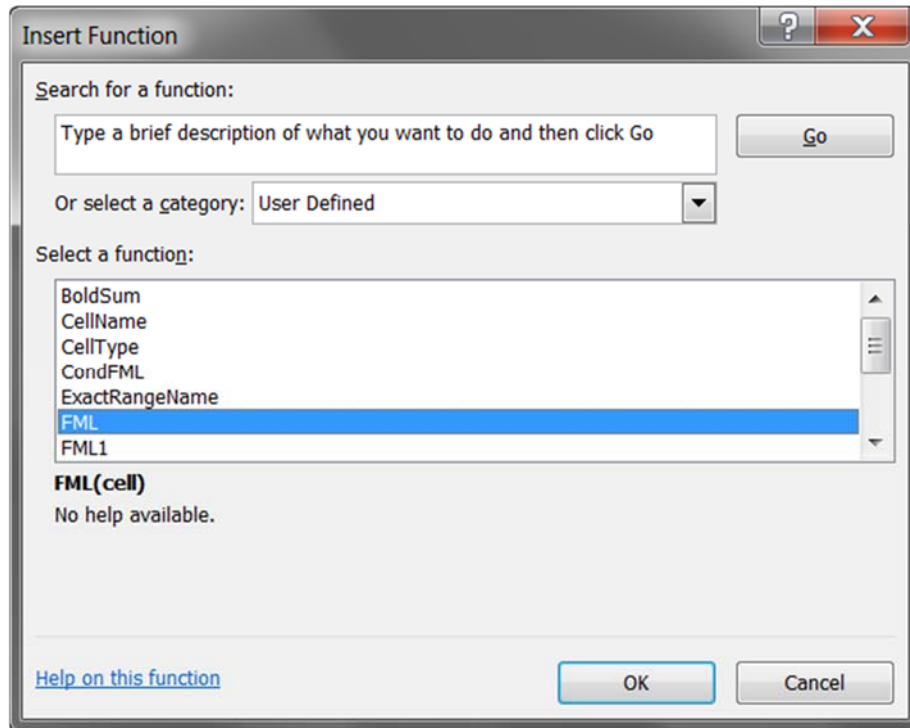
วิธีสร้าง Function VBA

1. เปิด Visual Basic Editor โดยกดปุ่ม Alt+F11
2. ใช้เมนูคำสั่ง Insert > Module เพื่อเปิดพื้นที่จัดเก็บรหัส
3. copy รหัสที่ต้องการ โดยเลือกช่วงตั้งแต่บรรทัด Function ถึงบรรทัด End Function
4. paste รหัสที่ copy ไว้นั้นลงใน module ที่ต้องการ
5. ใน Module หนึ่งๆ จะใช้เก็บ Function ได้ไม่จำกัด ให้เรียง Function ต่อกันไปเรื่อยๆ
6. สั่ง File > Save โดยกำหนดนามสกุลแฟ้ม xlsx



วิธีใช้ Function VBA

ใช้สูตรเช่นเดียวกับสูตรทั่วไปอื่นๆ หากใช้คำสั่ง Formulas > Insert Function หรือคลิกที่ปุ่ม
Fx บน Formula Bar จะพบสูตร VBA Function รวมไว้ในกลุ่ม Function category : User
Defined และแสดงรูปแบบสูตรในจอด้านขวาและด้านล่าง



Function GetFormat(cell)

1 GetFormat = cell.NumberFormat
End Function

แสดงรูปแบบที่ใช้ในเซลล์

Function GetFormula(Cell)

GetFormula = Cell.Formula

End Function

2

Function GetFormulaI(cell)

If VarType(cell) = 8 Then

GetFormulaI = "" & cell.formula

Else

GetFormulaI = cell.formula

End If

If cell.HasArray Then _

GetFormulaI = "{" & cell.formula & "}"

End Function

แสดงสูตรที่ใช้ในเซลล์

Function FormulaText(cell_ref)

'Allow formula to be updated if changes are made on the sheet

Application.Volatile

'Test for reference style in use

If Application.ReferenceStyle = xlA1 Then

3 'Set the return value of the function to the A1 style formula

FormulaText = cell_ref.Formula

Else ' xlR1C1 --Set the return value of the function to the R1C1 style formula

FormulaText = cell_ref.FormulaR1C1

End If

End Function

แสดงสูตรที่ใช้ในเซลล์ ทั้งแบบปกติหรือแบบ R1C1

4 Function FUNCTION_DEF(FCELL As Variant)As Variant

```

'Let's make sure the function
'recalculates when required
    Application.Volatile

'If the argument is not a range
'then return appropriate error.
    If Not TypeName(FCELL) = "Range" Then
        FUNCTION_DEF = CVErr(xlErrRef)
        Exit Function
    End If

'If the argument is not a single
'cell then return appropriate error.
    If Not FCELL.Cells.Count = 1 Then
        FUNCTION_DEF = CVErr(xlErrRef)
        Exit Function
    End If

'If the argument cell contains no
'formula, then just return its value
    If Not FCELL.HasFormula Then
        FUNCTION_DEF = FCELL.Value
        Exit Function
    End If

'Find out the format of the formula
'we need to return...
    Select Case Application.ReferenceStyle
        Case xlA1

```



```

        FUNCTION_DEF = FCELL.Formula
    Case xlR1C1
        FUNCTION_DEF = FCELL.FormulaR1C1
    End Select

    'Finally check whether we are looking at a cell
    'containing an array function.
    If FCELL.HasArray Then _
        FUNCTION_DEF = "{" & FUNCTION_DEF & "}"

End Function

```

แสดงสูตรในเซลล์

Option Explicit

Function DATEDIFF(d1, d2) As Variant

Dim YearDiff As Integer

Dim MonthDiff As Integer

Dim DayDiff As Integer

Dim temp As Date

5

' Swap arguments, if necessary

If d1 > d2 Then

temp = d1

d1 = d2

d2 = temp

End If

```

' Do the year part
YearDiff = Year(d2) - Year(d1)
If DateSerial(Year(d2), Month(d1), Day(d1)) > d2 _
    Then YearDiff = YearDiff - 1

' Do the month part
If Month(d2) > Month(d1) Then
    If Day(d2) >= Day(d1) Then
        MonthDiff = Month(d2) - Month(d1)
    Else
        MonthDiff = Month(d2) - Month(d1) - 1
    End If
Else
    If Day(d2) >= Day(d1) Then
        MonthDiff = Month(d2) - Month(d1) + 12
        If MonthDiff = 12 Then MonthDiff = 0
    Else
        MonthDiff = Month(d2) - Month(d1) + 11
    End If
End If

' Do the day part
If Day(d2) >= Day(d1) Then
    DayDiff = Day(d2) - Day(d1)
Else
    DayDiff = Day(DateSerial(Year(d1), _
        Month(d1) + 1, 1) - 1) - Day(d1) + Day(d2)
End If

' Create the variant array to be returned

```

```
DATEDIFF = Array(YearDiff, MonthDiff, DayDiff)
End Function
```

แสดงระยะเวลาระหว่างช่วงเวลาที่กำหนด

www.j-walk.com/ss/excel/tips/tip55.htm

```
Function Age(TheDate As Double) As String
```

```
Age = _
```

```
    CStr(Evaluate("=DATEDIF(" & TheDate & "," & CDbI(Now) &
    ",""y"")))) & " years " & _
```

```
6    CStr(Evaluate("=DATEDIF(" & TheDate & "," & CDbI(Now) &
    ",""ym"")))) & " months " & _
```

```
    CStr(Evaluate("=DATEDIF(" & TheDate & "," & CDbI(Now) &
    ",""md"")))) & " days"
```

```
End Function
```

แสดงระยะเวลาระหว่างช่วงเวลาที่กำหนดเป็นประโยครวม

```
Function OrdDate(arg)
```

```
    dd = Day(arg)
```

```
    mmmm = Format(arg, "mmmm")    '*Corrected*
```

7

```
    yyyy = Year(arg)
```

```
    Select Case Day(arg)
```

```
        Case 1, 21, 31
```

```
OrdDate = dd & "st " & mmmm & ", " & yyyy
```

```
Case 2, 22
```

```
OrdDate = dd & "nd " & mmmm & ", " & yyyy
```

```
Case 3, 23
```

```
OrdDate = dd & "rd " & mmmm & ", " & yyyy
```

```
Case 4 To 20, 24 To 30
```

```
OrdDate = dd & "th " & mmmm & ", " & yyyy
```

```
End Select
```

```
End Function
```

แปลงวันที่ให้มีคำต่อท้าย st nd rd th

```
Function SheetName() As String
```

```
8   SheetName = Application.Caller.Parent.Name 'in XL97
```

```
End Function
```

แสดงชื่อ sheet

```
Function CellType(c)
```

```
' Returns the cell type of the upper left
```

```
9
```

```
' cell in a range
```

```
Application.Volatile
```

```
Set c = c.Range("A1")
```

```
Select Case True
```

```
Case IsEmpty(c): CellType = "Blank"
```

```
Case Application.IsText(c): CellType = "Text"
```

```
Case Application.IsLogical(c): CellType = "Logical"
```

```
Case Application.IsErr(c): CellType = "Error"
```

```
Case IsDate(c): CellType = "Date"
```

```
Case InStr(1, c.Text, ":") <> 0: CellType = "Time"
```

```
Case IsNumeric(c): CellType = "Value"
```

```
End Select
```

```
End Function
```

แสดงประเภทของค่าในเซลล์

```
Function CellValue(c) As Double
```

```
'John Walkenbach 2001-04-25
```

```
10 ' misc returns number part
```

```
CellValue = Val(c)
```

```
End Function
```

แสดงค่าเป็นตัวเลข

Function HasFormula(cell)

11 HasFormula = cell.HasFormula

End Function

ตรวจสอบว่า มีสูตรหรือไม่

Function UseFormula(cell)

'-- Usage: Not recommended, see notes

UseFormula = Application.Evaluate(cell.formula)

End Function

Function UseFormula2(cell)

'Documented in

<http://www.geocities.com/davemcritchie/excel/formula.htm>

' UseFormula Jul 20, 1998, UseFormula2 Jun 13, 2000

12 'Application.Volatile = True -- DO NOT DO THIS

If Trim(cell.Value) = "" Then

UseFormula2 = ""

Exit Function

ElseIf Left(cell.Value, 1) = "=" Then

UseFormula2 = Application.Evaluate(cell.Formula)

Exit Function

Else

```
UseFormula2 = ""#bad formula"
```

```
End If
```

```
End Function
```

แปลงสูตรที่เป็นตัวอักษร ให้เป็นสูตรที่ใช้งานได้

```
Function fontinfo(cell As Range) As String
```

```
fontinfo = cell.FONT.Name & " -- " & cell.FONT.Size
```

```
If Left(cell.FONT.FontStyle, 7) = "Regular" Then
```

```
13 fontinfo = Trim(fontinfo & Mid(cell.FONT.FontStyle, 8, 100))
```

```
Else
```

```
fontinfo = Trim(fontinfo & " " & cell.FONT.FontStyle)
```

```
End If
```

```
End Function
```

แสดงรายละเอียดของ font ที่ใช้

```
Function SumByColor(InputRange As Range, ColorRange As Range) As  
Double
```

```
14 Dim cl As Range, TempSum As Double, ColorIndex As Integer
```

```
ColorIndex = ColorRange.Cells(1, 1).Interior.ColorIndex
```

```
TempSum = 0
```

```
On Error Resume Next ' ignore cells without values
```

```

For Each cl In InputRange.Cells
    If cl.Interior.ColorIndex = ColorIndex Then TempSum =
TempSum + cl.Value
Next cl
On Error GoTo 0
Set cl = Nothing
SumByColor = TempSum
End Function

```

หายอดรวมตามสีที่ใช้ www.erlandsendata.no

```

Function CountByColor(InputRange As Range, ColorRange as Range) As
Long

Dim cl As Range, TempCount As Long, ColorIndex As Integer

ColorIndex = ColorRange.Cells(1, 1).Interior.ColorIndex

TempCount = 0

For Each cl In InputRange.Cells
15    If cl.Interior.ColorIndex = ColorIndex Then TempCount =
TempCount + 1

Next cl

Set cl = Nothing

CountByColor = TempCount

End Function

```


นับจำนวนตามสัปดาห์

Function WEEKNR(InputDate As Long) As Integer

Dim A As Integer, B As Integer, C As Long, D As Integer

WEEKNR = 0

If InputDate < 1 Then Exit Function

A = Weekday(InputDate, vbSunday)

16

B = Year(InputDate + ((8 - A) Mod 7) - 3)

C = CDate("1.1." & B)

D = (Weekday(C, vbSunday) + 1) Mod 7

WEEKNR = Int((InputDate - C - 3 + D) / 7) + 1

End Function

ค้นค่าเป็นเลขที่สัปดาห์ www.erlandsendata.no

Function UniqueItem(InputRange As Range, ItemNo As Long) As Variant

17 Dim cl As Range, cUnique As New Collection, cValue As Variant

Application.Volatile

On Error Resume Next

```

For Each cl In InputRange
    If cl.Formula <> "" Then
        cUnique.Add cl.Value, CStr(cl.Value)
    End If
Next cl

UniqueItem = ""

If ItemNo = 0 Then
    UniqueItem = cUnique.Count
Else
    If ItemNo <= cUnique.Count Then
        UniqueItem = cUnique(ItemNo)
    End If
End If

On Error GoTo 0

End Function

แสดงค่า unique values

=UniqueItem(A1:A100,2) แสดงค่า the 2nd unique value
=UniqueItem(A1:A100,0) แสดงจำนวน unique values

```

```

Function INDEXN(InputRange As Range, N As Integer) As Variant

' returns every N-th item from InputRange

' select the desired target range for the function and

' enter as an array function with Ctrl+Shift+Enter.

Dim ItemList() As Variant, c As Range, i As Long, iCount As Long

    i = 0

    iCount = 0

    ReDim ItemList(1 To InputRange.Cells.Count \ N)

    For Each c In InputRange
18        i = i + 1

        If i Mod N = 0 Then

            iCount = iCount + 1

            On Error Resume Next

            ItemList(iCount) = c.Value

            On Error GoTo 0

        End If

    Next c

    INDEXN = ItemList

    If InputRange.Rows.Count >= InputRange.Columns.Count Then

```

```
INDEXN = Application.WorksheetFunction.Transpose(INDEXN)
```

```
End If
```

```
Erase ItemList
```

```
End Function
```

แสดงค่าที่อยู่ในลำดับที่ n

```
Sub FindUniqueValues(SourceRange As Range, TargetCell As Range)
```

```
19 SourceRange.AdvancedFilter Action:=xlFilterCopy,  
CopyToRange:=TargetCell, Unique:=True
```

```
End Sub
```

แยก unique items ออกมาสรุป

```
Function TimeInterval(StartTime As Double, EndTime As Double, _  
LowerLimit As Double, UpperLimit As Double) As Double
```

```
' returns EndTime-StartTime limited by LowerLimit and UpperLimit
```

```
20 TimeInterval = 0
```

```
If StartTime > EndTime Then Exit Function
```

```
If StartTime > UpperLimit Then Exit Function
```

```
If EndTime < LowerLimit Then Exit Function
```

If StartTime < LowerLimit Then StartTime = LowerLimit

If EndTime > UpperLimit Then EndTime = UpperLimit

TimeInterval = EndTime - StartTime

End Function

ระยะเวลาระหว่างช่วง

Option Explicit

' Main Function *

Function BahtEng(ByVal MyNumber)

Dim Baht, Satang, Temp

Dim DecimalPlace, Count

21 ReDim Place(9) As String

Place(2) = " Thousand "

Place(3) = " Million "

Place(4) = " Billion "

Place(5) = " Trillion "

' String representation of amount

MyNumber = Trim(Str(MyNumber))

' Position of decimal place 0 if none

DecimalPlace = InStr(MyNumber, ".")

```

'Convert Satang and set MyNumber to Baht amount
If DecimalPlace > 0 Then
    Satang = GetTens(Left(Mid(MyNumber, DecimalPlace + 1) & "00", 2))
    MyNumber = Trim(Left(MyNumber, DecimalPlace - 1))
End If

Count = 1
Do While MyNumber <> ""
    Temp = GetHundreds(Right(MyNumber, 3))
    If Temp <> "" Then Baht = Temp & Place(Count) & Baht
    If Len(MyNumber) > 3 Then
        MyNumber = Left(MyNumber, Len(MyNumber) - 3)
    Else
        MyNumber = ""
    End If
    Count = Count + 1
Loop

Select Case Baht
Case ""
    Baht = "No Baht"
Case "One"
    Baht = "One Baht"
Case Else
    Baht = Baht & " Baht"
End Select

Select Case Satang
Case ""
    Satang = " and No Satang"

```

```

Case "One"
Satang = " and One Satang"
Case Else
Satang = " and " & Satang & " Satang"
End Select

BahtEng = Baht & Satang
End Function

'*****
' Converts a number from 100-999 into text *
'*****

Private Function GetHundreds(ByVal MyNumber)
Dim Result As String

If Val(MyNumber) = 0 Then Exit Function
MyNumber = Right("000" & MyNumber, 3)

'Convert the hundreds place
If Mid(MyNumber, 1, 1) <> "0" Then
Result = GetDigit(Mid(MyNumber, 1, 1)) & " Hundred "
End If

'Convert the tens and ones place
If Mid(MyNumber, 2, 1) <> "0" Then
Result = Result & GetTens(Mid(MyNumber, 2))
Else
Result = Result & GetDigit(Mid(MyNumber, 3))
End If

```

```
GetHundreds = Result
```

```
End Function
```

```
*****
```

```
' Converts a number from 10 to 99 into text. *
```

```
*****
```

```
Private Function GetTens(TensText)
```

```
Dim Result As String
```

```
Result = "" 'null out the temporary function value
```

```
If Val(Left(TensText, 1)) = 1 Then ' If value between 10-19
```

```
Select Case Val(TensText)
```

```
Case 10: Result = "Ten"
```

```
Case 11: Result = "Eleven"
```

```
Case 12: Result = "Twelve"
```

```
Case 13: Result = "Thirteen"
```

```
Case 14: Result = "Fourteen"
```

```
Case 15: Result = "Fifteen"
```

```
Case 16: Result = "Sixteen"
```

```
Case 17: Result = "Seventeen"
```

```
Case 18: Result = "Eighteen"
```

```
Case 19: Result = "Nineteen"
```

```
Case Else
```

```
End Select
```

```
Else ' If value between 20-99
```

```
Select Case Val(Left(TensText, 1))
```

```
Case 2: Result = "Twenty "
```

```
Case 3: Result = "Thirty "
```

```
Case 4: Result = "Forty "
```

```
Case 5: Result = "Fifty "
```



```

Case 6: Result = "Sixty "
Case 7: Result = "Seventy "
Case 8: Result = "Eighty "
Case 9: Result = "Ninety "
Case Else
End Select
Result = Result & GetDigit _
(Right(TensText, 1)) 'Retrieve ones place
End If
GetTens = Result
End Function

'*****
' Converts a number from 1 to 9 into text. *
'*****

Private Function GetDigit(Digit)
Select Case Val(Digit)
Case 1: GetDigit = "One"
Case 2: GetDigit = "Two"
Case 3: GetDigit = "Three"
Case 4: GetDigit = "Four"
Case 5: GetDigit = "Five"
Case 6: GetDigit = "Six"
Case 7: GetDigit = "Seven"
Case 8: GetDigit = "Eight"
Case 9: GetDigit = "Nine"
Case Else: GetDigit = ""
End Select
End Function

```

สูตรแปลงตัวเลขเป็นคำอ่าน

Option Explicit

' Main Function *

Function BahtOnly(ByVal MyNumber)

Dim Baht, Satang, Temp

Dim DecimalPlace, Count

ReDim Place(9) As String

Place(2) = " Thousand "

Place(3) = " Million "

Place(4) = " Billion "

22 Place(5) = " Trillion "

' String representation of amount

MyNumber = Trim(Str(MyNumber))

' Position of decimal place 0 if none

DecimalPlace = InStr(MyNumber, ".")

'Convert Satang and set MyNumber to Baht amount

If DecimalPlace > 0 Then

Satang = GetTens(Left(Mid(MyNumber, DecimalPlace + 1) & "00", 2))

MyNumber = Trim(Left(MyNumber, DecimalPlace - 1))

End If

Count = 1

```

Do While MyNumber <> ""
    Temp = GetHundreds(Right(MyNumber, 3))
    If Temp <> "" Then Baht = Temp & Place(Count) & Baht
    If Len(MyNumber) > 3 Then
        MyNumber = Left(MyNumber, Len(MyNumber) - 3)
    Else
        MyNumber = ""
    End If
    Count = Count + 1
Loop

Select Case Baht
    Case ""
        Baht = "No Baht"
    Case "One"
        Baht = "One Baht"
    Case Else
        Baht = Baht & " Baht"
    End Select

Select Case Satang
    Case ""
        Satang = " Only"
    Case "One"
        Satang = " and One Satang"
    Case Else
        Satang = " and " & Satang & " Satang"
    End Select

BahtOnly = Baht & Satang

```

End Function

' Converts a number from 100-999 into text *

Private Function GetHundreds(ByVal MyNumber)

Dim Result As String

If Val(MyNumber) = 0 Then Exit Function

MyNumber = Right("000" & MyNumber, 3)

'Convert the hundreds place

If Mid(MyNumber, 1, 1) <> "0" Then

Result = GetDigit(Mid(MyNumber, 1, 1)) & " Hundred "

End If

'Convert the tens and ones place

If Mid(MyNumber, 2, 1) <> "0" Then

Result = Result & GetTens(Mid(MyNumber, 2))

Else

Result = Result & GetDigit(Mid(MyNumber, 3))

End If

GetHundreds = Result

End Function

' Converts a number from 10 to 99 into text. *

Private Function GetTens(TensText)

```
Dim Result As String
```

```
Result = "" 'null out the temporary function value
```

```
If Val(Left(TensText, 1)) = 1 Then ' If value between 10-19
```

```
Select Case Val(TensText)
```

```
Case 10: Result = "Ten"
```

```
Case 11: Result = "Eleven"
```

```
Case 12: Result = "Twelve"
```

```
Case 13: Result = "Thirteen"
```

```
Case 14: Result = "Fourteen"
```

```
Case 15: Result = "Fifteen"
```

```
Case 16: Result = "Sixteen"
```

```
Case 17: Result = "Seventeen"
```

```
Case 18: Result = "Eighteen"
```

```
Case 19: Result = "Nineteen"
```

```
Case Else
```

```
End Select
```

```
Else ' If value between 20-99
```

```
Select Case Val(Left(TensText, 1))
```

```
Case 2: Result = "Twenty "
```

```
Case 3: Result = "Thirty "
```

```
Case 4: Result = "Forty "
```

```
Case 5: Result = "Fifty "
```

```
Case 6: Result = "Sixty "
```

```
Case 7: Result = "Seventy "
```

```
Case 8: Result = "Eighty "
```

```
Case 9: Result = "Ninety "
```

```
Case Else
```

```
End Select
```

```
Result = Result & GetDigit _
```

```
(Right(TensText, 1)) 'Retrieve ones place
```

```
End If
```

```
GetTens = Result
```

```
End Function
```

```
*****
```

```
' Converts a number from 1 to 9 into text. *
```

```
*****
```

```
Private Function GetDigit(Digit)
```

```
Select Case Val(Digit)
```

```
Case 1: GetDigit = "One"
```

```
Case 2: GetDigit = "Two"
```

```
Case 3: GetDigit = "Three"
```

```
Case 4: GetDigit = "Four"
```

```
Case 5: GetDigit = "Five"
```

```
Case 6: GetDigit = "Six"
```

```
Case 7: GetDigit = "Seven"
```

```
Case 8: GetDigit = "Eight"
```

```
Case 9: GetDigit = "Nine"
```

```
Case Else: GetDigit = ""
```

```
End Select
```

```
End Function
```

สูตรแปลงตัวเลขเป็นคำอ่าน

```
Option Explicit
```

23

```
*****
```

```
' Main Function *
```

```

*****

Function Money(ByVal MyNumber, UnitName1, UnitName2)
Dim KeyUnit1, KeyUnit2, Temp
Dim DecimalPlace, Count

ReDim Place(9) As String
Place(2) = " Thousand "
Place(3) = " Million "
Place(4) = " Billion "
Place(5) = " Trillion "

' String representation of amount
MyNumber = Trim(Str(MyNumber))

' Position of decimal place 0 if none
DecimalPlace = InStr(MyNumber, ".")
'Convert KeyUnit2 and set MyNumber to KeyUnit1 amount
If DecimalPlace > 0 Then
KeyUnit2 = GetTens(Left(Mid(MyNumber, DecimalPlace + 1) & "00", 2))
MyNumber = Trim(Left(MyNumber, DecimalPlace - 1))
End If

Count = 1
Do While MyNumber <> ""
Temp = GetHundreds(Right(MyNumber, 3))
If Temp <> "" Then KeyUnit1 = Temp & Place(Count) & KeyUnit1
If Len(MyNumber) > 3 Then
MyNumber = Left(MyNumber, Len(MyNumber) - 3)
Else
MyNumber = ""

```

```

End If
Count = Count + 1
Loop

Select Case KeyUnit1
Case ""
KeyUnit1 = "No " & UnitName1
Case "One"
KeyUnit1 = "One " & UnitName1
Case Else
KeyUnit1 = KeyUnit1 & " " & UnitName1 & "s"
End Select

Select Case KeyUnit2
Case ""
KeyUnit2 = " Only"
Case "One"
KeyUnit2 = " and One " & " " & UnitName2
Case Else
KeyUnit2 = " and " & KeyUnit2 & " " & UnitName2 & "s"
End Select

Money = KeyUnit1 & KeyUnit2
End Function

'*****
' Converts a number from 100-999 into text *
'*****

Private Function GetHundreds(ByVal MyNumber)
Dim Result As String

```



```

If Val(MyNumber) = 0 Then Exit Function
MyNumber = Right("000" & MyNumber, 3)

'Convert the hundreds place
If Mid(MyNumber, 1, 1) <> "0" Then
Result = GetDigit(Mid(MyNumber, 1, 1)) & " Hundred "
End If

'Convert the tens and ones place
If Mid(MyNumber, 2, 1) <> "0" Then
Result = Result & GetTens(Mid(MyNumber, 2))
Else
Result = Result & GetDigit(Mid(MyNumber, 3))
End If

GetHundreds = Result
End Function

'*****
' Converts a number from 10 to 99 into text. *
'*****

Private Function GetTens(TensText)
Dim Result As String

Result = "" 'null out the temporary function value
If Val(Left(TensText, 1)) = 1 Then ' If value between 10-19
Select Case Val(TensText)
Case 10: Result = "Ten"
Case 11: Result = "Eleven"

```

```

Case 12: Result = "Twelve"
Case 13: Result = "Thirteen"
Case 14: Result = "Fourteen"
Case 15: Result = "Fifteen"
Case 16: Result = "Sixteen"
Case 17: Result = "Seventeen"
Case 18: Result = "Eighteen"
Case 19: Result = "Nineteen"
Case Else
End Select
Else ' If value between 20-99
Select Case Val(Left(TensText, 1))
Case 2: Result = "Twenty "
Case 3: Result = "Thirty "
Case 4: Result = "Forty "
Case 5: Result = "Fifty "
Case 6: Result = "Sixty "
Case 7: Result = "Seventy "
Case 8: Result = "Eighty "
Case 9: Result = "Ninety "
Case Else
End Select
Result = Result & GetDigit _
(Right(TensText, 1)) 'Retrieve ones place
End If
GetTens = Result
End Function

'*****
' Converts a number from 1 to 9 into text. *
```

```
*****
```

```
Private Function GetDigit(Digit)
```

```
Select Case Val(Digit)
```

```
Case 1: GetDigit = "One"
```

```
Case 2: GetDigit = "Two"
```

```
Case 3: GetDigit = "Three"
```

```
Case 4: GetDigit = "Four"
```

```
Case 5: GetDigit = "Five"
```

```
Case 6: GetDigit = "Six"
```

```
Case 7: GetDigit = "Seven"
```

```
Case 8: GetDigit = "Eight"
```

```
Case 9: GetDigit = "Nine"
```

```
Case Else: GetDigit = ""
```

```
End Select
```

```
End Function
```

สูตรแปลงตัวเลขเป็นคำอ่าน

Option Explicit

```
*****
```

```
' Main Function *
```

```
*****
```

24 Function UnitText(ByVal MyNumber, UnitName1, UnitName2)

```
Dim KeyUnit1, KeyUnit2, Temp
```

```
Dim DecimalPlace, Count
```

```
ReDim Place(9) As String
```

```
Place(2) = " Thousand "
```

```

Place(3) = " Million "
Place(4) = " Billion "
Place(5) = " Trillion "

' String representation of amount
MyNumber = Trim(Str(MyNumber))

' Position of decimal place 0 if none
DecimalPlace = InStr(MyNumber, ".")
'Convert KeyUnit2 and set MyNumber to KeyUnit1 amount
If DecimalPlace > 0 Then
KeyUnit2 = GetTens(Left(Mid(MyNumber, DecimalPlace + 1) & "00", 2))
MyNumber = Trim(Left(MyNumber, DecimalPlace - 1))
End If

Count = 1
Do While MyNumber <> ""
Temp = GetHundreds(Right(MyNumber, 3))
If Temp <> "" Then KeyUnit1 = Temp & Place(Count) & KeyUnit1
If Len(MyNumber) > 3 Then
MyNumber = Left(MyNumber, Len(MyNumber) - 3)
Else
MyNumber = ""
End If
Count = Count + 1
Loop

Select Case KeyUnit1
Case ""
KeyUnit1 = "No " & UnitName1

```

```

Case "One"
KeyUnit1 = "One " & UnitName1
Case Else
KeyUnit1 = KeyUnit1 & " " & UnitName1 & "" 'without s
End Select

Select Case KeyUnit2
Case ""
KeyUnit2 = " Only"
Case "One"
KeyUnit2 = " and One " & " " & UnitName2
Case Else
KeyUnit2 = " and " & KeyUnit2 & " " & UnitName2 & "" 'without s
End Select

UnitText = KeyUnit1 & KeyUnit2
End Function

'*****
' Converts a number from 100-999 into text *
'*****

Private Function GetHundreds(ByVal MyNumber)
Dim Result As String

If Val(MyNumber) = 0 Then Exit Function
MyNumber = Right("000" & MyNumber, 3)

'Convert the hundreds place
If Mid(MyNumber, 1, 1) <> "0" Then
Result = GetDigit(Mid(MyNumber, 1, 1)) & " Hundred "

```

End If

'Convert the tens and ones place

If Mid(MyNumber, 2, 1) <> "0" Then

Result = Result & GetTens(Mid(MyNumber, 2))

Else

Result = Result & GetDigit(Mid(MyNumber, 3))

End If

GetHundreds = Result

End Function

' Converts a number from 10 to 99 into text. *

Private Function GetTens(TensText)

Dim Result As String

Result = "" 'null out the temporary function value

If Val(Left(TensText, 1)) = 1 Then ' If value between 10-19

Select Case Val(TensText)

Case 10: Result = "Ten"

Case 11: Result = "Eleven"

Case 12: Result = "Twelve"

Case 13: Result = "Thirteen"

Case 14: Result = "Fourteen"

Case 15: Result = "Fifteen"

Case 16: Result = "Sixteen"

Case 17: Result = "Seventeen"

Case 18: Result = "Eighteen"

```

Case 19: Result = "Nineteen"
Case Else
End Select

Else ' If value between 20-99
Select Case Val(Left(TensText, 1))
Case 2: Result = "Twenty "
Case 3: Result = "Thirty "
Case 4: Result = "Forty "
Case 5: Result = "Fifty "
Case 6: Result = "Sixty "
Case 7: Result = "Seventy "
Case 8: Result = "Eighty "
Case 9: Result = "Ninety "
Case Else
End Select

Result = Result & GetDigit _
(Right(TensText, 1)) 'Retrieve ones place
End If

GetTens = Result
End Function

'*****
' Converts a number from 1 to 9 into text. *
'*****

Private Function GetDigit(Digit)
Select Case Val(Digit)
Case 1: GetDigit = "One"
Case 2: GetDigit = "Two"
Case 3: GetDigit = "Three"
Case 4: GetDigit = "Four"

```

```

Case 5: GetDigit = "Five"
Case 6: GetDigit = "Six"
Case 7: GetDigit = "Seven"
Case 8: GetDigit = "Eight"
Case 9: GetDigit = "Nine"
Case Else: GetDigit = ""
End Select
End Function

```

สูตรแปลงตัวเลขเป็นคำอ่าน

Option Explicit

Function NetYMD(Day1 As Date, Day2 As Date) As String

Dim years, months, days, m

years = Year(Day2) - Year(Day1)

If Month(Day1) > Month(Day2) Then

years = years - 1

25 End If

If Month(Day2) < Month(Day1) Then

months = 12 - Month(Day1) + Month(Day2)

Else

months = Month(Day2) - Month(Day1)

End If

If Day(Day2) < Day(Day1) Then

months = months - 1


```

If Month(Day2) = Month(Day1) Then
years = years - 1
months = 11
End If
End If

days = Day(Day2) - Day(Day1)
If days < 0 Then
m = CInt(Month(Day2)) - 1
If m = 0 Then m = 12
Select Case m
Case 1, 3, 5, 7, 8, 10, 12
days = 31 + days
Case 4, 6, 9, 11
days = 30 + days
Case 2
If (Year(Day2) Mod 4 = 0 And Year(Day2) _
Mod 100 <> 0) Or Year(Day2) Mod 400 = 0 Then
days = 29 + days
Else
days = 28 + days
End If
End Select
End If
NetYMD = CStr(years) + " years " + CStr(months) _
+ " months " + CStr(days) + " days "
End Function

Function NetYear(Day1 As Date, Day2 As Date) As String
Dim years, months, days, m

```

```

years = Year(Day2) - Year(Day1)
If Month(Day1) > Month(Day2) Then
years = years - 1
End If
NetYear = CStr(years)

End Function

Function NetMonth(Day1 As Date, Day2 As Date) As String
Dim years, months, days, m

If Month(Day2) < Month(Day1) Then
months = 12 - Month(Day1) + Month(Day2)
Else
months = Month(Day2) - Month(Day1)
End If

If Day(Day2) < Day(Day1) Then
months = months - 1
If Month(Day2) = Month(Day1) Then
years = years - 1
months = 11
End If
End If

NetMonth = CStr(months)

End Function

```

```
Function NetDay(Day1 As Date, Day2 As Date) As String
```

```
Dim years, months, days, m
```

```
days = Day(Day2) - Day(Day1)
```

```
If days < 0 Then
```

```
m = CInt(Month(Day2)) - 1
```

```
If m = 0 Then m = 12
```

```
Select Case m
```

```
Case 1, 3, 5, 7, 8, 10, 12
```

```
days = 31 + days
```

```
Case 4, 6, 9, 11
```

```
days = 30 + days
```

```
Case 2
```

```
If (Year(Day2) Mod 4 = 0 And Year(Day2) _
```

```
Mod 100 <> 0) Or Year(Day2) Mod 400 = 0 Then
```

```
days = 29 + days
```

```
Else
```

```
days = 28 + days
```

```
End If
```

```
End Select
```

```
End If
```

```
NetDay = CStr(days)
```

```
End Function
```

สูตรคำนวณระยะเวลา

26 Function PersonalTax(TaxableIncome)

```

Const UpperRange0 = 50000
Const UpperRange1 = 100000
Const UpperRange2 = 500000
Const UpperRange3 = 1000000
Const UpperRange4 = 4000000
Const TaxRate0 = 0
Const TaxRate1 = 0.05
Const TaxRate2 = 0.1
Const TaxRate3 = 0.2
Const TaxRate4 = 0.3
Const TaxRate5 = 0.37
Const TotalTaxPay0 = 0
Const TotalTaxPay1 = 2500
Const TotalTaxPay2 = 42500
Const TotalTaxPay3 = 142500
Const TotalTaxPay4 = 1042500
Select Case TaxableIncome
Case Is >= UpperRange4
PersonalTax = TotalTaxPay4 + _
(TaxRate5 * (TaxableIncome - UpperRange4))
Case Is >= UpperRange3
PersonalTax = TotalTaxPay3 + _
(TaxRate4 * (TaxableIncome - UpperRange3))
Case Is >= UpperRange2
PersonalTax = TotalTaxPay2 + _
(TaxRate3 * (TaxableIncome - UpperRange2))
Case Is >= UpperRange1
PersonalTax = TotalTaxPay1 + _
(TaxRate2 * (TaxableIncome - UpperRange1))
Case Is >= UpperRange0

```

```

PersonalTax = TotalTaxPay0 + _
(TaxRate1 * (TaxableIncome - UpperRange0))
Case Is < UpperRange0
PersonalTax = TaxRate0 * TaxableIncome
End Select

End Function

```

สูตรคำนวณภาษีหัก ณ ที่จ่าย

```

Public Function SigFig(X As Variant, SigFigs As Integer) As Variant

```

'Rounds X to Sigfigs significant figures

'Many thanks to John N. of Locum Destination Consulting for sharing his
SigFig() function for rounding to significant figures.

'http://www.pcqna.com/Excel_Rounding.htm

```

Dim Powers As Double, Sign As Long

```

```

On Error GoTo ErrHandler

```

```

27 If SigFigs < 1 Then GoTo ErrHandler

```

```

Sign = Sgn(X)

```

```

X = Abs(X)

```

```

Powers = 10 ^ (Int(Log(X) / Log(10#)) + 1)

```

'Application.Round() in next line is an Excel function; adjust as needed

```

SigFig = Sign * Application.Round(X / Powers, SigFigs) * Powers

```

```

Exit Function

```

```

ErrHandler:

```

```

SigFig = CVErr(xlErrValue)

```

```

End Function

```

'Further developed by Somkiat Foongkiat

'Excel Expert Training

Public Function SigFigDown(X As Variant, SigFigs As Integer) As Variant

Dim Powers As Double, Sign As Long

On Error GoTo ErrHandler

If SigFigs < 1 Then GoTo ErrHandler

Sign = Sgn(X)

X = Abs(X)

Powers = 10 ^ (Int(Log(X) / Log(10#)) + 1)

SigFigDown = Sign * Application.RoundDown(X / Powers, SigFigs) *

Powers

Exit Function

ErrHandler:

SigFigDown = CVErr(xlErrValue)

End Function

Public Function SigFigUp(X As Variant, SigFigs As Integer) As Variant

Dim Powers As Double, Sign As Long

On Error GoTo ErrHandler

If SigFigs < 1 Then GoTo ErrHandler

Sign = Sgn(X)

X = Abs(X)

Powers = 10 ^ (Int(Log(X) / Log(10#)) + 1)

SigFigUp = Sign * Application.RoundUp(X / Powers, SigFigs) * Powers

Exit Function

ErrHandler:

SigFigUp = CVErr(xlErrValue)

End Function

Public Function PowerNum(X As Variant) As Double

X = Abs(X)

PowerNum = (Int(Log(X) / Log(10#)) + 1)

End Function

Public Function RoundSP(Num As Variant, NumDigits As Integer) As Variant

Dim PowersX As Double, PowersZ As Double, Sign As Long

Dim X1 As Variant, X2 As Variant, X3 As Variant

Dim Y1 As Variant, Y2 As Variant

Dim Z1 As Variant, Z2 As Variant, Z3 As Variant

On Error GoTo ZeroHandler

Sign = Sgn(Num)

Num = Abs(Num)

X1 = Application.RoundDown(Num, NumDigits + 1)

PowersX = 10 ^ (Int(Log(X1) / Log(10#)) + 1)

X2 = X1 / PowersX

X3 = Right(X2, 1) 'หาตัวเลขหลักถัดไป

Y1 = Application.RoundDown(Num, NumDigits + 1)

Y2 = Num - Y1 'หาว่าถัดจากนั้นมีค่าอะไรอยู่อีก

Z1 = Application.RoundDown(Num, NumDigits)

PowersZ = 10 ^ (Int(Log(Z1) / Log(10#)) + 1)

Z2 = Z1 / PowersZ

Z3 = Right(Z2, 1) 'หาตัวเลข ณ หลักนั้น

If (X3 > 5) _

Or ((X3 = 5) And (Y2 <> 0)) _

Or ((X3 = 5) And (Z3 - Application.Odd(Z3)) = 0) _

Then

RoundSP = Sign * Application.Round(Num, NumDigits)

Else

RoundSP = Sign * Application.RoundDown(Num, NumDigits)

End If

Exit Function

ZeroHandler:

RoundSP = Sign * Application.Round(Num, NumDigits)

End Function

Public Function SigFigSP(X As Variant, SigFigs As Integer) As Variant

Dim Powers As Double, Sign As Long

On Error GoTo ErrHandler

If SigFigs < 1 Then GoTo ErrHandler

Sign = Sgn(X)

X = Abs(X)

Powers = 10 ^ (Int(Log(X) / Log(10#)) + 1)

SigFigSP = Sign * RoundSP(X / Powers, SigFigs) * Powers

Exit Function

ErrHandler:

SigFigSP = CVErr(xlErrValue)

End Function

Public Function CountDecimal(X As Variant) As Integer


```

X = Abs(X)
If X < 1 Then
CountDecimal = Len(X) - Len(Int(X))
Else
CountDecimal = Application.Max(0, Len(X) - Len(Int(X)) - 1)
End If
End Function

```

```

Public Function CountInteger(X As Variant) As Integer
X = Abs(X)
If Int(X) = 0 Then
CountInteger = 0
Else
CountInteger = Len(Int(X))
End If
End Function

```

```

Public Function CountSF(X As Variant) As Integer
Dim Powers As Double, X1 As Variant
X = Abs(X)
Powers = 10 ^ (Int(Log(X) / Log(10#)) + 1)
X1 = X / Powers
CountSF = Len(X1) - 1
End Function

```

Significant Figure

General Functions

1 =SUM(Sheet1:Sheet10!A1:A100)

รวมค่าในเซลล์ A1:A100 ของทุก sheet ตั้งแต่ sheet 1 ถึง 10

2 =COUNTA(Sheet1:Sheet10!A1:A100)

นับเซลล์ A1:A100 ที่มีค่าของทุก sheet ตั้งแต่ sheet 1 ถึง 10

3 =COUNTIF(data,"<0")

นับจำนวนเซลล์ที่มีค่าน้อยกว่า 0

4 =COUNTIF(data,"yes")

นับจำนวนเซลล์ที่มีค่า "yes"

5 =COUNTIF(data,"*")

นับจำนวนเซลล์ที่มีค่าใดๆ

6 =COUNTIF(data,"s*")

นับจำนวนเซลล์ที่มีค่านำด้วยตัวอักษร "s"

7 =COUNTIF(data,"*s*")

นับจำนวนเซลล์ที่มีตัวอักษร "s"

8 =COUNTIF(data,"yes")+COUNTIF(data,"no")

นับจำนวนเซลล์ที่มีค่า "yes" หรือ "no"

9 =COUNTIF(data,"???")

นับจำนวนเซลล์ที่มีค่า 3 ตัวอักษร

10 =COUNTIF(data,">=1")-COUNTIF(data,">10")

นับจำนวนเซลล์ที่มีค่าระหว่าง 1 ถึง 10

11 =SUMIF(A2:A10,"Jan",C2:C10)

รวมค่าของ C2:C10 ในตำแหน่งซึ่ง A2:A10 = "Jan"

12 =COUNTIF(A2:A10,"Jan",C2:C10)

นับค่าของ C2:C10 ในตำแหน่งซึ่ง A2:A10 = "Jan"

13 =SUMIF(A2:A10,"<>Jan",C2:C10)

รวมค่าของ C2:C10 ในตำแหน่งซึ่ง A2:A10 <> "Jan"

14 =SUMIF(Month,"Jan",Sales)+SUMIF(Month,"Feb",Sales)

รวมค่าของ Sales ในตำแหน่งซึ่ง Month="Jan" หรือ "Feb"

15 =SUMIF(A1:A6,"Part#1",B1:B6)
=SUMPRODUCT((A1:A6="Part#1")*(B1:B6))

รวมค่าของ Part#1

16 =SUM(IF(FREQUENCY(data,data)>0,1,0))

นับจำนวนเซลล์ที่มีค่า unique numeric values

17 =IF(COUNTIF(A\$2:A2,A2)>1,COUNTIF(A\$2:A2,A2),"")
=IF(COUNTIF(A\$2:A2,A2)>1,"Dup","")

แสดงตำแหน่งเซลล์ที่มีค่าซ้ำ

18 =IF(A1>=1,INT(A1)&" " & TEXT(MOD(A1,1)*12,"-
0"&IF(ABS(MOD(A1,1)*12-ROUND(MOD(A1,1)*12,0))>1/32,"
0/"&CHOOSE(ROUND(MOD(MOD(A1,1)*12,1)*16,0),16,8,16,4,16,
8,16,2,16,8,16,4,16,8,16),"")) &"")

ปัดเลขความยาวจาก feet และ decimal feet ไปเป็น feet, inches
แบบปัดค่า 1/16 inch fractions.

19 =INT(A1/12)&" " & TEXT(MOD(A1,12),"0"&IF(ABS(MOD(A1,12)-
ROUND(MOD(A1,12),0))>1/32,"
0/"&CHOOSE(ROUND(MOD(MOD(A1,12),1)*16,0),
16,8,16,4,16,8,16,2,16,8,16,4,16,8, 16),"")) &"")

ปัดเลขความยาวจาก inches และ decimal fractions ไปเป็น feet
และ inches แบบปัดค่า 1/16th inch fractions

20 =TEXT(A1,"0"&IF(ABS(A1-ROUND(A1,0))>1/32," 0/"& CHOOSE(
ROUND(MOD(A1,1)*16,0),16,8,16,
4,16,8,16,2,16,8,16,4,16,8,16),"")) &"")

ปัดเลขความยาวจาก inches และ decimal fractions ไปเป็น inches
only with rounded 1/16th inch fractions

21 =VALUE(LEFT(A1,FIND("","",A1)-1)) +
VALUE(MID(A1,FIND("","",A1) +1,FIND("","",A1)-FIND("","",A1)-
1))/12

เปลี่ยนเลขความยาว จาก Feet and Inches ไปเป็น feet

22 =VALUE(LEFT(A1,FIND("","A1")-1))*12 +
VALUE(MID(A1,FIND("","A1") +1,FIND("","A1")-FIND("","A1")-1))

เปลี่ยนเลขความยาว จาก Feet and Inches back ไปเป็น inches

สูตร	คืนค่า
=SUM(E4:E23)	รวมค่าใน E4:E23
=SUBTOTAL(9;G4:G26)	รวมค่าใน G4:G26 เฉพาะ visible cells
=SUMPRODUCT(E4:E13;F4:F13)	รวมผลคูณระหว่างแถว E4:E13 คูณกับ F4:F13
{=SUM(E4:E13*F4:F13)}	เหมือนสูตรข้างต้นแต่ใช้ array formula
=SUMSQ(E4:E13)	$S(x^2)$
=SUMX2MY2(E4:E13;E14:E23)	$S(x^2 - y^2)$
=SUMX2PY2(E4:E13;E14:E23)	$S(x^2 + y^2)$
=SUMXMY2(E4:E13;E14:E23)	$S(x - y)^2$

24 =COUNTBLANK(E2:E23)

นับเซลล์ว่าง

25 =CELL("filename")

แสดง path- และ filenames

26 =RAND()*100

แสดงเลขสุ่มเป็นเลขทศนิยมระหว่าง 0 ถึง 100

27 =RAND()*(100-50)+50

แสดงเลขสุ่มเป็นเลขทศนิยมระหว่าง 50 ถึง 100

28 =ROUND(RAND()*100,0)

แสดงเลขสุ่มเป็นเลขจำนวนเต็มระหว่าง 0 ถึง 100

29

A

สูตรสะสม

1	100	=SUM(\$A\$1:A1)
2	200	=SUM(\$A\$1:A2)
3	300	=SUM(\$A\$1:A3)
4	400	=SUM(\$A\$1:A4)
5	500	=SUM(\$A\$1:A5

สูตรยอดรวมสะสม

30 =LEN(B1)-LEN(SUBSTITUTE(SUBSTITUTE(B1,"B",""),"b",""))

count the number of "B"s, both upper and lower case, in the string in B1

31 =OFFSET(Sheet1!\$A\$2,0,0,COUNTA(\$A\$2:\$A\$20),1)

Dynamic Ranges

32 =RIGHT(A2,LEN(A2)-FIND(" ",SUBSTITUTE(A2," ","*",LEN(A2)-LEN(SUBSTITUTE(A2," ","")))))

To return the last name. Suppose cell A2 contains the name
"John A Smith"

33 =LEFT(A2,FIND("*",SUBSTITUTE(A2," ","*",LEN(A2)-
LEN(SUBSTITUTE(A2," ",""))))-1)

To return the first name, including the middle name (if
present),

34 =LEFT(B2,FIND(" ",B2,1))

To return the first name, without the middle name (if
present),

35 =LEFT(A1,FIND(" ",A1,1))

Returning First Word In A String

36 =RIGHT(A1,LEN(A1)-FIND(" ",A1,1))

Returning All But First Word In A String

"Grades" which refers to = {0,"F";60,"D";70,"C";80,"B";90,"A"}

37

=VLOOKUP(A1,Grades,2)

to convert the number to the grade:

38 `=IF(ROW()-ROW(TopRng)+1>TopN,"",LARGE(RankRng,ROW()-ROW(TopRng)+1))`

Ranking Numbers : to return the N highest or lowest values from a range of data

Suppose we have a range of numeric data called RankRng. Create a range next to RankRng (starting in the same row, with the same number of rows) called TopRng. Also, create a named cell called TopN, and enter into it the number of values you want to return (e.g., 5 for the top 5 values in RankRng). Enter the following formula in the first cell in TopRng, and use Fill Down to fill out the range

39 `=IF(ROW()-ROW(TopRng)+1>TopN,"",SMALL(RankRng,ROW()-ROW(TopRng)+1))`

To return the TopN smallest values of RankRng

40 `=CELL("filename",A1)`

To return the full sheet name (including the file path)

41 = MID(CELL("filename",A1),FIND("]",CELL("filename",A1))+1,
 LEN(CELL("filename",A1))-FIND("]",CELL("filename",A1)))

To return the sheet name, without the path

42 = MID(CELL("filename",A1),FIND("[",CELL("filename",A1))+1,FIND("]",
 CELL("filename",A1))-FIND("[",CELL("filename",A1))-1)

To return the file name without the path

43 = LEFT(CELL("filename",A1),FIND("]",CELL("filename",A1))) Or
 = SUBSTITUTE(SUBSTITUTE(LEFT(CELL("filename",A1),FIND("]",
 CELL("filename",A1))),"[",""),"]", "")

To return the file name with the path

44 = RadiusEarth*ACOS(COS(RADIANS(90-
 (Lat1*24)))*COS(RADIANS(90-(Lat2*24)))+
 SIN(RADIANS(90-(Lat1*24)))*SIN(RADIANS(90-(Lat2*24)))*
 COS(RADIANS(24*(Long1-Long2))))

Great Circle Distances

Lat1 is the latitude of point 1, entered as DD:MM:SS.

Long1 is the longitude of point 1, entered as DD:MM:SS.

Lat2 is the latitude of point 2, entered as DD:MM:SS.

Long2 is the longitude of point 2, entered as DD:MM:SS.
 RadiusEarth is the radius of the earth (3,963 miles or 6,377 kilometers).

45 =RadiusEarth*ACOS(COS(RADIANS(90-Lat1))*COS(RADIANS(90-Lat2))+
 SIN(RADIANS(90-Lat1))*SIN(RADIANS(90-Lat2))*COS(RADIANS(Long1-Long2)))

mixing hemispheres, enter Northern and Western coordinates as positive, and Southern and Eastern coordinates as negative

46 =LEFT(A2,IF(ISERROR(FIND(",",A2,1)),LEN(A2),FIND(",",A2,1)-1))

To return the last name of the full name in A2

47 =TRIM(IF(ISERROR(FIND(",",A2,1)),A2,MID(A2,FIND(",",A2,1)+1,IF(ISERROR(FIND(" ",A2,FIND(",",A2,1)+2)),LEN(A2),FIND(" ",A2,FIND(",",A2,1)+2))-FIND(",",A2,1))))

To return the first name

48 =TRIM(RIGHT(A2,LEN(A2)-IF(ISERROR(FIND(" ",A2,
FIND(" ",A2,FIND(" ",A2,1)+2))),LEN(A2),
FIND(" ",A2,FIND(" ",A2,FIND(" ",A2,1)+2))-1)))

To return the middle name

49 =MAX(A1:A10,B1)

the highest score ever reached

the Tolls->Options dialog, click on the Calculation tab, and
check the Iterations check box

50 =SUM(INDIRECT("A1:A10"))

=SUM(\$A\$1:\$A\$10),

51 =A1&IF(AND(A1>=10,A1<=14),"th",
CHOOSE(MOD(A1,10)+1,"th","st","nd","rd","th","th","th","th","th","th"))

return the number in A1 with the suffix appended

52 =RANK(C7,C\$7:C\$16)+COUNTIF(C\$7:C7,C7)-1

Unique Ranks

53 =COUNT(C\$7:C\$16)-
 (RANK(C7,C\$7:C\$16)+COUNTIF(C\$7:C7,C7))+2

Reverse Unique Ranks

54 =CELL("filename",A1)

D:\driveM\excel\TAXES\[1996FEDT.XLS]Sheet1

55 =LEFT(CELL("filename",A1),FIND("[",CELL("filename",A1),1)-1)
 =INFO("directory")

D:\driveM\excel\TAXES\

56 =SUBSTITUTE(LEFT(CELL("filename",A1),FIND("]",CELL("filename",A1),1)-
 1),"[",",")

D:\driveM\excel\TAXES\1996FEDT.XLS

57 =SUBSTITUTE(SUBSTITUTE(CELL("filename",A1),"[",","),"]",", " [")
 & "]"

D:\driveM\excel\TAXES\1996FEDT.XLS [Sheet1]

58 =SUBSTITUTE(LEFT(LOWER(CELL("filename",A1)),FIND("]",
CELL("filename",A1))-1),"[",")" & " [" &
MID(CELL("filename",A1),FIND("]",CELL("filename",A1))+1,28) &
"]"

d:\drivem\excel\taxes\1996fedt.xls [Sheet1]

59 =MID(CELL("filename",A1),FIND("[",CELL("filename",A1),1)+1,FIND("]",
CELL("filename",A1),1)-FIND("[",CELL("filename",A1),1)-1)

1996FEDT.XLS

60 =RIGHT(CELL("filename",A1),LEN(CELL("filename",A1))-
FIND("]", CELL("filename",A1)))
=MID(CELL("filename",A1),FIND("]",CELL("filename",A1))+1,99)
=SheetName()

Sheet1

61 =INFO("directory")
directory, memavail, memused, numfile, origin, osversion, recalc,
release, system, totmem

C:\WINNT\Profiles\Administrator\Personal\

62 =UPPER(LEFT(A37,1)&". "&MID(A37,FIND(" ",A37&" ")+1,1)&".
 ")

Extract the first two initials

63 =IF(LEN(A11)=0,"",IF(ISERR(FIND(" ",A11)),A11,LEFT(A11,FIND(" ",A11)-1)))

Extract the first word

 =len(a1)-len(substitute(a1,"a","")) — lettercase must match
 64 =len(a1)-len(substitute(upper(a1),"A","")) — lettercase does
 not matter

Find number of occurrences of a character in a cell

65 =LEN(A1)-LEN(SUBSTITUTE(A1,"",""))+(LEN(TRIM(A1))>0)

Find the number of items separated by commas

66 =LEN(SUBSTITUTE(TRIM(A1),CHAR(32),CHAR(32)&CHAR(32))
)-LEN(TRIM(A1))+1

Find the number of words in a string

67 =LEN(A1)-LEN(SUBSTITUTE(A1,"aa",""))/LEN("aa")

Find the number of occurrences of a string in a single cell

68 =countif(A1:J1,"aa")

Find the number of cells in a range that have string as their value

=IF(ISERR(FIND("/",E22)),E22,MID(E22,FIND("/",E22,1)+1,99))
69 =IF(ISERR(FIND("/",E22)), "", MID(E22,FIND("/",E22,1)+1,99))

Return string after first "/" character

=NOT(ISNUMBER(VALUE(LEFT(D7,1))))
70 =IF(LEFT(TRIM(D7))="", "", NOT(ISNUMBER(VALUE(LEFT(D7,1)))))

First Character checked as alphabetic character

71 =SUBSTITUTE(A1," ","")

Remove All Spaces

72 =IF(MOD(INT(16*(+B2-INT(B2)+0.0312)),16)=0,TEXT(B2,"#"),IF(MOD(INT(16*(+B2-INT(B2)+0.0312)),8)=0,TEXT(B2,"# 0/2"),IF(MOD(INT(16*(+B2-INT(B2)+0.0312)),4)=0,TEXT(B2,"# 0/4"),IF(MOD(INT(16*(+B2-INT(B2)+0.0312)),2)=0,TEXT(B2,"# 0/8"),TEXT(B2,"# 0/16")))))

to reduce 8/16's to 1/2, 10/16 to 5/8

73 =SUMPRODUCT(B2:B6,C2:C6)/SUM(C2:C6)

Weighted Average

74 =OFFSET(MySheet!\$A\$1,0,0,COUNTA(MySheet!\$A:\$A),1)

Dynamic Named Ranges

75 =POWER((SUM(IF(values0,values*(POWER(1+rRate,(MAX(dates)-dates)/daybase)),0)))/(SUM(IF(values<0,values/(POWER(1+iRate,(MAX(dates)-dates)/daybase)),0))*-1,1/((MAX(dates)-MIN(dates))/daybase))-1

combine the functionality of the XIRR and MIRR functions

values is the row or column range of cashflows

dates is the row or column range of corresponding dates

iRate is the interest rate you pay on the money used in the

cash flows

rRate is the interest rate you receive on the cash flows as you
reinvest them

daybase is days-in-year basis to use (usually 360 or 365).

7 $=\text{MROUND}(\text{Num}, \text{IF}(\text{VALUE}(\text{RIGHT}(\text{Num}/10^{(\text{INT}(\text{LOG}(\text{ABS}(\text{Num}))) - \text{Plc} + 1}), 2)) = 0.5, 2, 1) * \text{SIGN}(\text{Num}) * 10^{(\text{INT}(\text{LOG}(\text{ABS}(\text{Num}))) - \text{Plc} + 1)})$

6 $=\text{MROUND}(\text{Num}, \text{IF}(\text{VALUE}(\text{RIGHT}(\text{Num}/\text{Fact}, 2)) = 0.5, 2, 1) * \text{SIGN}(\text{Num}) * \text{Fact})$

'bankers rounding' for a number (Num) to a given number (Plc) of
significant digits

define 'Fact' as $=10^{(\text{INT}(\text{LOG}(\text{ABS}(\text{Num}))) - \text{Plc} + 1)}$,

77 $=\text{INDEX}(\text{FREQUENCY}((\text{A1}, \text{A3}, \text{A5}), 20), 2)$

นับจำนวนเซลล์ซึ่งไม่ได้ติดต่อกัน และมีค่ามากกว่า 20

78 $=\text{LEN}(\text{A1}) - \text{LEN}(\text{SUBSTITUTE}(\text{A1}, "b", ""))$

$=\text{LEN}(\text{A1}) - \text{LEN}(\text{SUBSTITUTE}(\text{SUBSTITUTE}(\text{A1}, "b", ""), "b", ""))$

นับตัวอักษร b ในเซลล์

79 $=\text{SUM}(\text{LEN}(\text{A1}) - \text{LEN}(\text{SUBSTITUTE}(\text{A1}, \text{B1}, ""))) / \text{LEN}(\text{B1})$

=SUM(LEN(A1)-
LEN(SUBSTITUTE(UPPER(A1),UPPER(B1),"")))/LEN(B1)

นับจำนวนคำ B1 ในประโยค A1

=A1&IF(OR(VALUE(RIGHT(A1,2))={11,12,13}),
80 "th",IF(OR(VALUE(RIGHT(A1))={1,2,3}),CHOOSE(RIGHT(A1),
"st","nd","rd"), "th"))

Ordinal Number

=IF(AND(A1>0,A1<257),IF(A1>26,CHAR(CEILING(A1/26,1)+63),"")
81 &CHAR(IF(MOD(A1,26)=0,26,MOD(A1,26))+64),"")

แปลงเลขที่ column เป็นตัวอักษร column

=MID(A1,FIND(";",SUBSTITUTE(A1,"\\",";",LEN(A1)-
82 LEN(SUBSTITUTE(A1,"\\",))))+1,LEN(A1))

แยกชื่อ file จาก c:\MainDir\ชื่อ file

=IF(ISERR(LEFT(A1,FIND(" ",A1)-1)),A1,LEFT(A1,FIND(" ",A1)-
83 1))

แยกคำแรกออกจากประโยคภาษาอังกฤษ

=IF(LEN(A4)-LEN(SUBSTITUTE(A4," ",""))=0, A4,
 84 RIGHT(A4,LEN(A4)-FIND(" ",SUBSTITUTE(A4," ","*", LEN(A4)-
 LEN(SUBSTITUTE(A4," ",""))))))

แยกคำสุดท้ายออกจากประโยคภาษาอังกฤษ

85 =RIGHT(A1,LEN(A1)-FIND(" ",A1,1))

แยกคำที่เหลือซึ่งไม่ใช่คำแรก ออกจากประโยคภาษาอังกฤษ

=LEFT(A1,FIND(" ",A1)-1)
 =IF(ISERR(MID(A1,FIND(" ",A1)+1,IF(ISERR(FIND(" ",A1,FIND(" ",A1)+1)),FIND(" ",A1),FIND(" ",A1,FIND(" ",A1)+1))-FIND(" ",A1)-1)),",",MID(A1,FIND(" ",A1)+1,IF(ISERR(FIND(" ",A1,FIND(" ",A1)+1)),FIND(" ",A1), FIND(" ",A1,FIND(" ",A1)+1))-FIND(" ",A1)-1))
 86
 =RIGHT(A1,LEN(A1)-FIND(" ",SUBSTITUTE(A1," ","*",LEN(A1)-LEN(SUBSTITUTE(A1," ",""))))))

แยก first name, middle names, last names

87 =IF(OR(LEFT(A1,2)="Mr",LEFT(A1,3)="Mrs",LEFT(A1,2)="Ms"),
 RIGHT(A1,LEN(A1)-FIND(" ",A1)),A1)

ตัดคำนำหน้าชื่อออกจากชื่อ

88 =LEN(SUBSTITUTE(TRIM(A1),CHAR(32),CHAR(32)&CHAR(32)))-
LEN(TRIM(A1))+1

นับจำนวนคำในเซลล์

89 =LEFT(TRIM(A1),FIND(" ",TRIM(A1),1)-1)&"
&RIGHT(TRIM(A1),LEN(TRIM(A1))-IF(ISERROR(FIND("
",TRIM(A1),FIND(" ",TRIM(A1),1)+1)),FIND("
",TRIM(A1),1),FIND(" ",TRIM(A1),FIND(" ",TRIM(A1),1)+1)))

ตัด middle names ออกจากชื่อนามสกุล

90 =SUBTOTAL(9, INDIRECT(ADDRESS(ROW(C3),COLUMN(C3)
)&":"& ADDRESS(ROW(C3),
ROW(INDIRECT(COLUMN(C3)&":"&COLUMN(G3))))))

Cumulative Sum ค่าในเซลล์ C3:G3

91 =MAX(0, MIN(เวลาสิ้นสุดกะ, กำหนดสิ้นสุด) - MAX(เวลาเริ่มกะ,
กำหนดเริ่ม))

ระยะเวลาที่ใช้ในกะนั้น

92 =IF(Date>=Start, Amount, 0)

แสดง Amount เมื่อถึงวันที่กำหนด

93 =IF(Date<=Stop, Amount, 0)

แสดง Amount เมื่อยังไม่เลยวันที่กำหนด

94 =IF(AND(Date>=Start, Date<=Stop), Amount, 0)

แสดง Amount ในช่วงวันที่กำหนด

95 =IF(AND(Date>=Start, Date<=Stop, OR(MOD(Date-
Start+1,Cycle) = 1,Cycle = 1)), Amount, 0)

แสดง Amount ในช่วงวันที่กำหนด และกำหนดเป็นช่วงๆตาม Cycle

Day and Time Formulas

1 =DATEDIF(Date1,Date2,Interval)

คำนวณระยะเวลาระหว่างช่วงที่กำหนด	
Interval Code	ความหมาย
"m"	จำนวนเดือนสะสม
"d"	จำนวนวันสะสม
"y"	จำนวนปีสะสม
"ym"	จำนวนเดือนที่เหลือ เมื่อนับปีชนปี
"yd"	จำนวนวันที่เหลือ เมื่อนับปีชนปี
"md"	จำนวนวันที่เหลือ เมื่อนับเดือนชนเดือน แต่ไม่ควรใช้เพราะอาจคืนค่าแปลกๆ

2 =DATEDIF(วันเกิด,NOW(),"y") & " years, " & DATEDIF(วันเกิด,NOW(),"ym") & " months, " & DATEDIF(วันเกิด,NOW(),"md") & " days"

คำนวณอายุ

3 =AND((TDate>=MIN(TDate1,TDate2)),TDate<=MAX(TDate1,TDate2))

ตรวจสอบว่า TDate อยู่ระหว่าง TDate1 ถึง TDate2

=IF(OR(Date2<VDate1,VDate2<Date1),Date2-
Date1+1,IF(OR(AND(Date1<=VDate1,Date2>=VDate2),
AND(Date1>=VDate1,Date2<=VDate2)),MAX(0,(Date2-Date1)-(VDate2-
4 VDate1)),
IF(OR(AND(Date1<=VDate1,Date2<=VDate2),AND(Date1>=VDate1,Date2
>VDate2)),
MAX(0,(VDate1-Date1))+MAX(0,Date2-VDate2),NA()))))

นับจำนวนวันในระหว่างช่วงวันที่กำหนด

ชื่อ	ความหมาย
Date1	วันที่เริ่มงาน
Date2	วันที่เสร็จงาน
VDate1	วันเริ่ม ซึ่งไม่ต้องการนับเป็นวันทำงาน
VDate2	วันสุดท้าย ซึ่งไม่ต้องการนับเป็นวันทำงาน

NWRRange

ตารางวันหยุด

```
=IF(OR(Date2<VDate1,VDate2<Date1),NETWORKDAYS(Date1,Date2,NWR
ange),
IF(OR(AND(Date1<=VDate1,Date2>=VDate2),AND(Date1>=VDate1,Date2
<=VDate2)),
MAX(0,NETWORKDAYS(Date1,Date2,NWRRange)-
5 NETWORKDAYS(VDate1,VDate2,NWRRange)),
IF(OR(AND(Date1<=VDate1,Date2<=VDate2),AND(Date1>=VDate1,Date2
>VDate2)),
IF((Date1>=VDate1),0,NETWORKDAYS(Date1,VDate1-1,NWRRange))+
IF((Date2<=VDate2),0,NETWORKDAYS(VDate2+1,Date2,NWRRange)),NA()
))
```

นับวันเฉพาะวันธรรมดา ไม่นับวันหยุดสุดสัปดาห์หรือวันหยุดอื่น

```
6 =IF(OR(IDate2<RDate1,IDate1>RDate2),0,(MIN(IDate2,RDate2)-
MAX(IDate1,RDate1)+1))
```

นับจำนวนวันที่อยู่ในทั้งสองช่วงเวลาเหมือนกัน

ชื่อ	ความหมาย
IDate1	วันเริ่มต้นของช่วงแรก
IDate2	วันสุดท้ายของช่วงแรก

RDate1	วันเริ่มต้นของช่วงที่สอง
RDate1	วันสุดท้ายของช่วงที่สอง
NWRange	ตารางวันหยุด

=IF(OR(IDate2<RDate1,IDate1>RDate2),0,
7 ABS(NETWORKDAYS(MIN(IDate2,RDate2),MAX(IDate1,RDate1),NWRange)
))

นับวันเฉพาะวันธรรมดา ไม่นับวันหยุดสุดสัปดาห์หรือวันหยุดอื่น

8 =MAX(0,NETWORKDAYS(MAX(D\$5,\$B6),MIN(DATE(YEAR(D\$5),MONTH(D\$
5)+1,0),\$C6)))

สรุปจำนวนวันทำงานแต่ละเดือน

	A	B	C	D	E	F
3						
4		Working Days		Months		
5		Start Date	End Date	Jan	Feb	Mar
6		15-Jan-00	12-Jun-00	11	21	23
7		16-Jan-00	2-Sep-00	11	21	23
8						

9 =MAX(0,(MIN(\$C17,DATE(YEAR(D\$16),MONTH(D\$16)+1,0))-
MAX(\$B17,D\$16)+1))

สรุปจำนวนวันแต่ละเดือน

	A	B	C	D	E	F
14						
15		Days		Months		
16		Start Date	End Date	Jan	Feb	Mar
17		15-Jan-00	12-Jun-00	17	29	31
18		21-Jan-00	2-Sep-00	11	29	31
19						

1 =MAX(0,NETWORKDAYS(MAX(DATE(D\$27,1,1),\$B28),MIN(DATE(D\$27,12,1),

0 31),\$C28)))

สรุปจำนวนวันทำงานแต่ละปี

	A	B	C	D	E	F
25						
26		Network days		Years		
27		Start Date	End Date	2000	2001	2002
28		10-Jan-00	15-Jan-02	255	261	11
29		15-Jun-03	28-Jul-09	0	0	0
30						

11 =MAX(0,(MIN(\$C39,DATE(D\$38,12,31))-

MAX(\$B39,DATE(D\$38,1,1))+1))

สรุปจำนวนวันแต่ละปี

	A	B	C	D	E	F
36						
37		Days		Years		
38		Start Date	End Date	2000	2001	2002
39		10-Jan-00	15-Jan-02	357	365	15
40		15-Jun-03	28-Jul-09	0	0	0
41						

12 =FLOOR("5/"&DAY(MINUTE(B2/38)/2+56)&"/"&B2,7)-34
 =FLOOR(DAY(MINUTE(B2/38)/2+56)&"/5/"&B2,7)-34

วัน Easter

B2 เป็นปีที่ต้องการหาวัน Easter

13 =TRUNC(((B6-DATE(YEAR(B6),1,0))+6)/7)

เลขที่สัปดาห์

14 =INT((A1-SUM(MOD(DATE(YEAR(A1-MOD(A1-2,7)+3),1,2),
 {1E+99,7})*{1,-1}+5)/7)
 =1+INT((A1-DATE(YEAR(A1+4-WEEKDAY(A1+6)),1,5)+
 WEEKDAY(DATE(YEAR(A1+4-WEEKDAY(A1+6)),1,3)))/7)

เลขที่สัปดาห์ตาม ISO

15 =RIGHT(YEAR(A1),2)&TEXT(A1-DATE(YEAR(A1),1,0),"000")

เปลี่ยนวันที่ปกติ เป็น Julian Date

16 =DATE(IF(0+(LEFT(A1,2))<30,2000,1900)+LEFT(A1,2),1,RIGHT(A1,3))

เปลี่ยน Julian date เป็นวันที่ปกติ

=DATE(IF(0+(LEFT(A1,2))<30,2000,1900)+LEFT(A1,2),1,RIGHT(A1,3))
 17 -
 DATE(IF(0+(LEFT(A2,2))<30,2000,1900)+LEFT(A2,2),1,RIGHT(A2,3))

ระยะเวลาระหว่างสอง Julian Dates

1 =DATE(IF(0+(LEFT(A1,2))<30,2000,1900)+LEFT(A1,2),1,RIGHT(A1,3)+A
 8 2)

เพิ่มจำนวนวันนับจาก Julian Date

19 =IF(ROW(A1),CALL("Xlcall32","Excel4","2JRJ",74))
 =IF(ROW(A1:A4),CALL("Xlcall32","Excel4","2JRJ",74))

วันปรับปรุงค่าครั้งสุดท้าย

20 =IF(INT(StartDT)=INT(EndDT),"0 days " & ROUND(24*(EndDT-
 StartDT),2)&"hours",MAX(NETWORKDAYS(StartDT+1,EndDT-
 1,HolidayList),0)+INT(24*(((EndDT-INT(EndDT))-(StartDT-
 INT(StartDT)))+(DayEnd-DayStart))/(24*(DayEnd-DayStart)))&"
 days "&MOD(ROUND(((24*(EndDT-INT(EndDT)))-
 24*DayStart)+(24*DayEnd-(24*(StartDT-INT(StartDT))))),2),
 ROUND((24*(DayEnd-DayStart)),2))&" hours ")

ระยะเวลาเป็นวันและชั่วโมง ระหว่างช่วงเวลาที่กำหนด

ชื่อ	ความหมาย	ตัวอย่าง
StartDT	วันที่และเวลา เริ่มต้น	25-Oct-99 13:00
EndDT	วันที่และเวลา สิ้นสุด	28-Oct-99 15:00
DayStart	เวลาเริ่มงาน ประจำวัน	9:00
DayEnd	เวลาเลิกงาน ประจำวัน	17:00
HolidayList	ตารางวันหยุด	H1:H10

=IF(INT(StartDT)=INT(EndDT),ROUND(24*(EndDT-StartDT),2),
 (24*(DayEnd-DayStart)*(MAX(NETWORKDAYS(StartDT+1,EndDT-
 1,HolidayList),0)+INT(24*(((EndDT-INT(EndDT))-
 21 (StartDT-INT(StartDT)))+(DayEnd-DayStart)))/(24*(DayEnd-
 DayStart))))+MOD(ROUND(((24*(EndDT-INT(EndDT)))-
 24*DayStart)+(24*DayEnd-(24*(StartDT-INT(StartDT)))),2),
 ROUND((24*(DayEnd-DayStart),2))))

นับจำนวนชั่วโมงทำงาน

22 =IF(OR(WEEKDAY(A4+1)=1,WEEKDAY(A4+1)=7),A4+3,A4+1)

ลำดับวันที่เฉพาะวันทำงานปกติ นับตั้งแต่วันแรกในเซลล์ A4

23 =DAY(DATE(YEAR(A1),MONTH(A1)+1,0))

เลขที่จำนวนวันในแต่ละเดือน หรือเลขวันที่สุดท้ายของเดือน

24 =DATE(YEAR(A1),MONTH(A1)+1,0)

วันที่สุดท้ายของเดือน

25 =DATE(YEAR(A1),MONTH(A1),0)

วันที่สุดท้ายของเดือนก่อน

26 =DATE(Yr,Mon,1+((Nth-(DoW>=WEEKDAY(DATE(Yr,Mon,1))))*7)+
(DoW-WEEKDAY(DATE(Yr,Mon,1))))

return the date of Nth day-of-week for a given month and year.
For example, it will return 26-March-98 for the 4th Thursday of
March, 1998. Days-of-week range from 1 to 7, with Sunday =
1 and Saturday = 7.

Where Yr, Mon, Nth, and DoW are cell references or values
indicating Year, Month, Nth, and Day-Of-Week.

27 `=DATE(Yr,1,1+(Nth-(Dow>=WEEKDAY(DATE(Yr,1,1))))*7)+
Dow-WEEKDAY(DATE(Yr,1,1))`

return the date of Nth day-of-week for a given year

Where Yr,Nth, and DoW are cell references or values indicating
Year, Month, Nth, and Day-Of-Week

28 `=TRUNC((B1-A1)/7)&" Weeks "&MOD(B1-A1,7)&" Days"`

จำนวนสัปดาห์ระหว่างช่วงเวลาที่กำหนด

29 `=IF(ISERROR(VLOOKUP(WEEKDAY(A1),WorkDays,1,0)),FALSE,TRUE)`

ตรวจสอบว่า เป็นวันทำงานปกติหรือไม่

30 `=ROUNDUP(MONTH(A1)/3,0)`

เลขที่ไตรมาส

31 `=TRUNC(((A1-DATE(YEAR(A1),1,1))/7))+1+
IF(WEEKDAY(DATE(YEAR(A1),1,1))>WEEKDAY(A1),1,0)`

เลขที่สัปดาห์ของวันที่กำหนด

32 =DATE(YEAR(A1)+1,MONTH(A1)+6, DAY(A1)+10)

วันที่นับถัดไปอีก 1 ปี 6 เดือน 10 วัน

33 =IF(A1>B1,B1+1-A1,B1-A1)

ระยะเวลาระหว่างช่วงเวลา

34 =TIME(HOUR(A1),MROUND(MINUTE(A1),B1),0)

ปรับเวลา ในเซลล์ A1 เป็นเลขชั่วโมง ครึ่งชั่วโมง หรือสิบห้านาที ที่ใกล้ที่สุด

B1 เป็นเลขนาทีกี่ ซึ่งต้องการปัดเข้าสู่

=TIME(HOUR(A1),FLOOR(MINUTE(A1),B1),0)

35

=TIME(HOUR(A1),CEILING(MINUTE(A1),B1),0)

ปัดเวลาขึ้นลงสู่ช่วงที่ใกล้ที่สุด

36 =SUMPRODUCT(N(MONTH(A1:A20)=12))

นับจำนวนเซลล์ที่มีวันที่ซึ่งอยู่ในเดือนธันวาคม

37 =SUMPRODUCT((MONTH(A1:A20)=12)*(YEAR(A1:A20)=2000))

นับจำนวนเซลล์ที่มีวันที่ซึ่งอยู่ในเดือนธันวาคม ปี 2000

3 =SUMPRODUCT(N(A1:A20>DATE(2000,11,15))*N(A1:A20<=DATE(2000,12,15)))

8 =countif(A1:A20,">=16/11/2000")-countif(A1:A20,">15/12/2000")

นับจำนวนเซลล์ที่มีวันที่ซึ่งอยู่ระหว่างวันที่ 16 พฤศจิกายน ถึงวันที่ 15 เดือนธันวาคม

39 =(A1>B1)+B1-A1

หาระยะเวลาระหว่างเวลาที่กำหนด

40		A	B	C
	1	TIME	Time	สูตรที่ใช้
	2	1	0:01:00	=ROUNDDOWN(A2,-2) / 2400 + MOD(A2,100) / 1440
	3	2	0:02:00	=ROUNDDOWN(A3,-2) / 2400 + MOD(A3,100) / 1440
	4	3	0:03:00	=ROUNDDOWN(A4,-2) / 2400 + MOD(A4,100) / 1440

5	1200	12:00:00	=ROUNDDOWN(A5,-2) / 2400 + MOD(A5,100) / 1440
6	1201	12:01:00	=ROUNDDOWN(A6,-2) / 2400 + MOD(A6,100) / 1440
7	2400	0:00:00	=ROUNDDOWN(A7,-2) / 2400 + MOD(A7,100) / 1440
8	3359	9:59:00	=ROUNDDOWN(A8,-2) / 2400 + MOD(A8,100) / 1440

เปลี่ยนเวลาแบบทหาร เป็น เวลาแบบมาตรฐาน

=INT((A1-(DATE(YEAR(A1+(MOD(8-WEEKDAY(A1),7)-3)),1,1))-3+
 41 MOD(WEEKDAY(DATE(YEAR(A1+(MOD(8-WEEKDAY(A1),7)-3)),1,1))+1,7))/7)+1

คืนค่าเป็นเลขที่สเปดาห์

42 =NOW()+12/24

เวลา 12 ชั่วโมงถัดไปจากเวลาปัจจุบัน (24 ชม / วัน)

43 =NOW()+10/1440

เวลา 10 นาทีถัดไปจากเวลาปัจจุบัน (1440 นาที / วัน)

44 =NOW()+30/86400

เวลา 30 วินาทีถัดไปจากเวลาปัจจุบัน (86400 วินาที / วัน)

A2: =TODAY()

45 B2: =A2-MOD(A2-2,7)

C2: =B2

Monday week starting dates

=A2&IF(INT(MOD(A2,100)/10)=1, "th", IF(MOD(A2,10)=1, "st",
IF(MOD(A2,10)=2,"nd", IF(MOD(A2,10)=3, "rd","th"))))

46 =DAY(A2)&IF(INT(MOD(DAY(A2),100)/10)=1, "th",
IF(MOD(DAY(A2),10)=1, "st", IF(MOD(DAY(A2),10)=2,"nd",
IF(MOD(DAY(A2),10)=3, "rd","th"))))& " " & TEXT(A2,"mmmm,
yyyy")

ตัวเลขและวันที่แบบนับมีคำว่า st nd rd th ต่อท้าย

47 =MROUND(A22,1/(24*4))

Rounding to nearest quarter hour

48 =FLOOR(A22,1/(24*4))

Rounding down to nearest quarter hour

49 =CEILING(A22,1/(24*4))

Rounding Up to nearest quarter hour

50 =TEXT(NOW(),"dd mmmm yyyy")

เปลี่ยนเวลาเป็นรูปแบบตัวอักษร จะได้จัดชิดซ้ายและยืดข้ามขอบขวา
ออกไปได้

(วันที่แสดงในแบบของ US/Canada mm/dd/yyyy)

5 1		04/14/200 1	Sat	36995
	First Day of Week	04/08/200 1	Sun	=B1-WEEKDAY(B1)+1
	Last Day of Week	04/14/200 1	Sat	=B1-WEEKDAY(B1)+7

First Day of Month	04/01/2001	Sun	=DATE(YEAR(B1),MONTH(B1),1)
Last Day of Month	04/30/2001	Mon	=DATE(YEAR(B1),MONTH(B1)+1,0)
First Day of Year	04/01/2001	Sun	=DATE(YEAR(B1),MONTH(B1),1)
Last Day of Year	12/31/2001	Mon	=DATE(YEAR(B1)+1,1,0)
Closest Monday	04/16/2001	Mon	=DATE(YEAR(B1),MONTH(B1),DAY(B1)+CHOOSE(WEEKDAY(B1),1,0,-1,-2,-3,3,2,1))
Next Monday	04/16/2001	Mon	=DATE(YEAR(B1),MONTH(B1),DAY(B1)+CHOOSE(WEEKDAY(B1),1,7,6,5,4,3,2))
Next Monday	04/16/2001	Mon	=A1-WEEKDAY(A1,2)+8
1st Monday of Month	04/02/2001	Mon	=DATE(YEAR(B1),MONTH(B1),CHOOSE(WEEKDAY(DATE(YEAR(B1),MONTH(B1),1)),1,7,6,5,4,3,2))

			1), MONTH(B1,1)),2,1,7,6,5,4,3))
2nd Monday of Month	04/09/2001	Monday	=DATE(YEAR(B1),MONTH(B1),7 +CHOOSE(WEEKDAY(DATE(YEAR(B1), MONTH(B1,1)),2,1,7,6,5,4,3))
3rd Monday of Month	04/16/2001	Monday	=DATE(YEAR(B1),MONTH(B1),14 +CHOOSE(WEEKDAY(DATE(YEAR(B1), MONTH(B1,1)),2,1,7,6,5,4,3))
4th Monday of Month	04/23/2001	Monday	=DATE(YEAR(B1),MONTH(B1),21 +CHOOSE(WEEKDAY(DATE(YEAR(B1), MONTH(B1,1)),2,1,7,6,5,4,3))
5th Monday of Month	04/30/2001	Monday	=IF(MONTH(DATE(YEAR(B1), MONTH(B1),28+ CHOOSE(WEEKDAY(DATE(YEAR(B1), MONTH(B1,1)),2,1,7,6,5,4,3)))= MONTH(B1),DATE(YEAR(B1), MONTH(B1),28 +CHOOSE(WEEKDAY(DATE(YEAR(B1), MONTH(B1,1)),2,1,7,6,5,4,3))), "none")

Last Monday of Month	04/30/2001	Mon	=DATE(YEAR(\$B\$1), MONTH(\$B\$1)+1,1)-WEEKDAY(DATE(YEAR(\$B\$1), MONTH(\$B\$1)+1,6))
<p>For a different day of the week rotate the 2nd to last parameters in CHOOSE.</p> <p>i.e. 7,6,5,4,3,2,1 for Wednesday instead of 2,1,7,6,5,4,3 for Monday as used in some of the formulae.</p>			
Formula in C1 & D1, downward			=IF(ISNUMBER(B1),WEEKDAY(B1), "")

5 2	A	B	E
	Dates	format mm/dd/yyyy	สูตรที่ใช้
	First ddd of Month	03/01/1999	=DATE(YEAR(B18),MONTH(B18),1)

Last ddd of Month	03/31/1999	=DATE(YEAR(\$B\$18),MONTH(\$B\$18)+1,1)-1
First Sunday of Mo.	03/07/1999	=DATE(YEAR(\$B\$18),MONTH(\$B\$18),1)+ MOD(1+7- WEEKDAY(DATE(YEAR(\$B\$18),MONTH(\$B\$18),1)),7)
First Monday	03/01/1999	=DATE(YEAR(\$B\$18),MONTH(\$B\$18),1)+ MOD(2+7- WEEKDAY(DATE(YEAR(\$B\$18),MONTH(\$B\$18),1)),7)
First Tuesday	03/02/1999	=DATE(YEAR(\$B\$18),MONTH(\$B\$18),1)+ MOD(3+7- WEEKDAY(DATE(YEAR(\$B\$18),MONTH(\$B\$18),1)),7)
First Wednesda y	03/03/1999	=DATE(YEAR(\$B\$18),MONTH(\$B\$18),1)+ MOD(4+7- WEEKDAY(DATE(YEAR(\$B\$18),MONTH(\$B\$18),1)),7)
First Thursday	03/04/1999	=DATE(YEAR(\$B\$18),MONTH(\$B\$18),1)+ MOD(5+7- WEEKDAY(DATE(YEAR(\$B\$18),MONTH(\$B\$18),1)),7)

First Friday	03/05/1999	=DATE(YEAR(\$B\$18),MONTH(\$B\$18),1)+ MOD(6+7- WEEKDAY(DATE(YEAR(\$B\$18),MONTH(\$B\$18),1)),7)
First Saturday	03/06/1999	=DATE(YEAR(\$B\$18),MONTH(\$B\$18),1)+ MOD(7+7- WEEKDAY(DATE(YEAR(\$B\$18),MONTH(\$B\$18),1)),7)
Last Sunday of Mo.	03/28/1999	=DATE(YEAR(\$B\$18),MONTH(\$B\$18)+1,1)- WEEKDAY(DATE(YEAR(\$B\$18),MONTH(\$B\$18)+1,7))
Last Monday	03/29/1999	=DATE(YEAR(\$B\$18),MONTH(\$B\$18)+1,1)- WEEKDAY(DATE(YEAR(\$B\$18),MONTH(\$B\$18)+1,6))
Last Tuesday	03/30/1999	=DATE(YEAR(\$B\$18),MONTH(\$B\$18)+1,1)- WEEKDAY(DATE(YEAR(\$B\$18),MONTH(\$B\$18)+1,5))
Last Wednesday	03/31/1999	=DATE(YEAR(\$B\$18),MONTH(\$B\$18)+1,1)- WEEKDAY(DATE(YEAR(\$B\$18),MONTH(\$B\$18)+1,4))

Last Thursday	03/25/1999	=DATE(YEAR(\$B\$18),MONTH(\$B\$18)+1,1)- WEEKDAY(DATE(YEAR(\$B\$18),MONTH(\$B\$18)+ 1,3))
Last Friday	03/26/1999	=DATE(YEAR(\$B\$18),MONTH(\$B\$18)+1,1)- WEEKDAY(DATE(YEAR(\$B\$18),MONTH(\$B\$18)+ 1,2))
Last Saturday	03/27/1999	=DATE(YEAR(\$B\$18),MONTH(\$B\$18)+1,1)- WEEKDAY(DATE(YEAR(\$B\$18),MONTH(\$B\$18)+ 1,8))
Last weeknum of yr	53	=WEEKNUM(DATE(YEAR(B18),12,31),1)

	Description	สูตร
	The date for Monday in the previous week	=TODAY()-WEEKDAY(TODAY(),2)-6
	The date for Monday in the current week	=TODAY()-WEEKDAY(TODAY(),2)+1
53	The date for Monday in the next week	=TODAY()-WEEKDAY(TODAY(),2)+8
	Create a valid date	=DATE(2002,12,24)
	Create a valid date	=DATEVALUE("1.1.1980")
	Count of days in a month	=DAY(DATE(YEAR(A1),MONTH(A1)+1,0))
	Last date in a month	=DATE(YEAR(A1),MONTH(A1)+1,0)

Determine which
quarter a date
belongs to

=CHOOSE(MONTH(A1),1,1,1,2,2,2,3,3,3,4,4,4)

54

=DAY(A1)&IF(INT(MOD(DAY(A1),100)/10)=1, "th",
IF(MOD(DAY(A1),10)=1, "st",IF(MOD(DAY(A1),10)=2,"nd",
IF(MOD(DAY(A1),10)=3, "rd", "th"))))& " " &TEXT(A1,"mmmm,
yyyy")

Ordinal Date

List & Database Formulas

1 =INDEX(Rng,MATCH(MAX(COUNTIF(Rng,Rng)),COUNTIF(Rng,Rng),0))

แสดงค่าที่ใช้บ่อยที่สุด

2 =INDEX(Rng,MATCH(MIN(COUNTIF(Rng,Rng)),COUNTIF(Rng,Rng),0))

แสดงค่าที่ใช้บ่อยครั้งที่สุด

3 =INDIRECT(ADDRESS(ROW(Rng)+MATCH(C1,Rng,0)-1,COLUMN(Rng)-
ColsToLeft)) Or
=INDIRECT(ADDRESS(ROW(Rng)+MATCH(C1,Rng,0)-1,COLUMN(A:A)
)

Left Lookups

4 =IF(MAX(COUNTIF(Range1,Range1))>1,"Duplicates","No Duplicates")

ตรวจสอบค่าซ้ำ

5 =IF(COUNTIF(Range1, A5)>1,TRUE,FALSE)

ตรวจสอบค่าซ้ำ

6 =IF(COUNTIF(Range1,???)>1,"Duplicate","")

ตรวจสอบค่าซ้ำ

7 =COUNTIF(\$A:\$A,\$A1)>1

ตรวจสอบค่าซ้ำในแถวตั้ง

=SUM(N(OFFSET(E1,ROW(1:5)-1,-1*ROW(1:5)+1)))

=SUMPRODUCT(N(OFFSET(E1,ROW(INDIRECT("1:5"))-1,-

8 1*ROW(INDIRECT("1:5"))+1)))

=SUM(N(OFFSET(E1,ROW(INDIRECT("1:5"))-1,-

1*ROW(INDIRECT("1:5"))+1)))

sum A5:E1 ตามแนวทะแยงมุม

=INDEX(Range2,MATCH(Value,Range1,0))

9 =LOOKUP(Value,Range1,Range2)

=VLOOKUP(Value,Range12,2,FALSE)

สูตรค้นหา

Conditional Formatting & Data Validation Formulas

Conditional Formatting

1 $\text{=MOD(ROW()-Rw,N*2)+1} \leq \text{N}$

กำหนดรูปแบบให้เป็นแถบแบบเลขคี่
Rw เป็นเลขที่บรรทัดแรก
N เป็นจำนวนบรรทัดในแต่ละแถบ

2 $\text{=MOD(ROW()-Rw,N*2)+1} > \text{N}$

กำหนดรูปแบบให้เป็นแถบแบบเลขคู่
Rw เป็นเลขที่บรรทัดแรก
N เป็นจำนวนบรรทัดในแต่ละแถบ

3 $\text{=ISERROR(cell_reference)}$

กำหนดรูปแบบให้กับเซลล์ที่เป็น Error Values

4 =ISTEXT(A1)

เซลล์ซึ่งไม่ใช่ตัวเลข

5 =A1>AVERAGE(\$A\$1:\$D\$12)

เซลล์ซึ่งมีค่าเกินกว่าค่าเฉลี่ย

6 =MONTH(A1)=6

เซลล์ซึ่งเป็นเดือนมิถุนายน

7 =A1=TODAY()

เซลล์ซึ่งเป็นวันนี้

8 =IF(ISBLANK(A1),"",OR(WEEKDAY(A1)=7,WEEKDAY(A1)=1))

เซลล์ซึ่งเป็นวันเสาร์อาทิตย์

9 =A1=MAX(\$A\$1:\$A\$30)

เซลล์ซึ่งมีค่าสูงสุด

10 =OR(A1=LARGE(\$A\$1:\$A\$30,1), A1=LARGE(\$A\$1:\$A\$30,2),
A1=LARGE(\$A\$1:\$A\$30,3))

เซลล์ซึ่งมีค่าสูงสุด 3 ลำดับแรก

11 =MOD(ROW(),2)=0

row เลขคู่

12 =MOD(INT((ROW()-1)/4)+1,2)

row ชุดละ 4 row

13 =MOD(ROW(),2)=MOD(COLUMN(),2)

row เลขคู่ และเป็น column เลขคู่

14 =IF(COUNTIF(\$A\$1:\$D\$12,A1)>1,TRUE,FALSE)

เซลล์ซึ่งมีค่าซ้ำ

15 =A2<A1

เซลล์ซึ่งไม่ได้เรียงลำดับ

16 =LEN(SUBSTITUTE(TRIM(A1), CHAR(32), CHAR(32)&CHAR(32)))-
LEN(TRIM(A1))+1>1

เซลล์ซึ่งมีคำมากกว่า 1 คำ

17 =LEN(A3)-LEN(SUBSTITUTE(SUBSTITUTE(A3,"A",""),"a",""))>0

เซลล์ซึ่งมีตัวอักษรตัว a

18 =AND(\$B\$1:\$B\$4<>"")

เมื่อเซลล์อื่นมีค่าครบแล้ว

1 =MATCH(\$A1&\$B1&\$C1&\$D1,\$F\$6:\$F\$15&\$G\$6:\$G\$15&\$H\$6:\$H\$15&\$I\$
9 6:\$I\$15,0)>=1

เซลล์ซึ่งมีค่าตรงกับค่าในตารางทุก column \$F\$6:\$F\$15 และ \$G\$6:\$G\$15
และ \$H\$6:\$H\$15 และ \$I\$6:\$I\$15

Data Validation

1 =COUNTIF(\$A\$1:\$A\$50,A1)=1

ป้องกันการบันทึกข้อมูลซ้ำ

2 =ISNA(VLOOKUP(A9,A\$1:A8,1,FALSE))

ป้องกันการบันทึกข้อมูลซ้ำกับข้อมูลบรรทัดเหนือขึ้นไป

3 =ISTEXT(A5)

รับเฉพาะตัวอักษร

4 =A2>A1

รับค่าซึ่งมากขึ้น

5 =LEFT(A2)="a"

รับค่าซึ่งนำด้วยตัว a

6 =COUNTIF(A3,"A????")=1

รับค่าซึ่งนำด้วยตัว a และมี 5 ตัวอักษร

Array Formulas

สูตร Array ต่อไปนี้ต้องสร้างขึ้นโดยกดปุ่ม **Ctrl+Shift+Enter** จะเกิดเครื่องหมายวงเล็บปีกกาปิดหัวท้ายสูตร ซึ่งสามารถใช้กับ Excel ได้ทุกรุ่น

=SUM(IF(List>0,List,FALSE))

=AVERAGE(IF(List>0,List,FALSE))

1

=MIN(IF(List>0,List,FALSE))

=MAX(IF(List>0,List,FALSE))

หาผลลัพธ์เฉพาะค่าบวกเท่านั้น สามารถละไม่ต้องใส่
คำว่า FALSE ในสูตร

=SUM(IF(List<>0,List,FALSE))

=AVERAGE(IF(List<>0,List,FALSE))

2

=MIN(IF(List<>0,List,FALSE))

=MAX(IF(List<>0,List,FALSE))

หาผลลัพธ์เฉพาะค่าที่ไม่เท่ากับ 0 เท่านั้น สามารถละ
ไม่ต้องใส่คำว่า FALSE ในสูตร

=SUM(IF((List>=LLim)*(List<=ULim),List,FALSE))

=AVERAGE(IF((List>=LLim)*(List<=ULim),List,FALSE))

3

=MIN(IF((List>=LLim)*(List<=ULim),List,FALSE))

=MAX(IF((List>=LLim)*(List<=ULim),List,FALSE))

หาผลลัพธ์เฉพาะค่าที่อยู่ระหว่าง LLim ถึง ULim สามารถ
ละไม่ต้องใส่คำว่า FALSE ในสูตร

LLim เป็นขีดจำกัดล่าง, ULim เป็นขีดจำกัดบน

=IF(A1=B1,1,0)

=SUM(IF(A1:A10=B1:B10,1,0))

=SUM((A2:A10="Phone")*(B2:B10="Smith")*C2:C10)

4

=SUM(IF((A2:A10="Fax")+(B2:B10="Jones"),1,0))

=SUM(IF(MOD((A2:A10="Fax")+(B2:B10="Jones")),2),1,0))

=SUM(IF((A2:A10="Fax")+(B2:B10="Jones")<>2,1,0))

นับจำนวนเซลล์ตามเงื่อนไข โดยใช้สูตร SUM ตัวเลข 1 โดย
ไม่ต้องใช้สูตร COUNT

สังเกตการใช้เครื่องหมายบวก หรือ เครื่องหมายคูณ

5 =SUM((Month="Jan")*(Region="North")*Sales)

รวมค่าของ Sales ในตำแหน่งซึ่ง Month="Jan" AND
Region="North"

6 =SUM((Month="Jan")*(Region<>"North")*Sales)

รวมค่าของ Sales ในตำแหน่งซึ่ง Month="Jan" AND
Region<>"North"

7 =SUM((Month="Jan")*(Region="North"))

นับค่าของ Sales ในตำแหน่งซึ่ง Month="Jan" และ
Region="North"

8 =SUM((Month="Jan")*(Sales>=200)*(Sales))

รวมค่าของ Sales ในตำแหน่งซึ่ง Month="Jan" และ
Sales>= 200

9 =SUM((Sales>=300)*(Sales<=400)*(Sales))

รวมค่าของ Sales ระหว่าง 300 ถึง 400

10 =SUM((Sales>=300)*(Sales<=400))

นับค่าของ Sales ระหว่าง 300 ถึง 400

11 =SUM((A1:A10>=5)*(A1:A10<=10)*A1:A10)

รวมค่าระหว่าง 5 ถึง 10

12 =SUM((A1:A10>=5)*(A1:A10<=10))

นับจำนวนเซลล์ที่มีค่าระหว่าง 5 ถึง 10

สูตรนับ	คืนค่า
=SUM((F4:F23="yes")+(F4:F23="no"))	นับจำนวนเซลล์ที่มีคำว่า yes หรือ no
=SUM(MOD(G4:G23,2))	นับจำนวนเซลล์ที่มีเลขคี่
=SUM(IF(MOD(G4:G23,2)=0,1,0))	นับจำนวนเซลล์ที่มีเลขคู่
=SUM(IF((F4:F23="yes")*(G4:G23>10),1,0))	นับจำนวนเซลล์แบบและ
=SUM((F5:F24="yes")*(G5:G24>10))	นับจำนวนเซลล์แบบและ

=SUM((F4:F23="yes")+(F4:F23="no"))

นับจำนวนเซลล์
แบบหรือ

สังเกตว่าคำนวณแบบหรือ ใช้เครื่องหมายบวก ส่วนแบบและ ใช้
เครื่องหมายคูณ

=SUM(IF(MOD(ROW(\$A\$1:\$A\$20),N)=0,\$A\$1:\$A\$20,0))

=SUM(IF(MOD(ROW(\$B\$3:\$B\$22)-
14 ROW(\$B\$3)+1,N)=0,\$B\$3:B\$20,0))

=SUM(IF(MOD(ROW(\$B\$3:\$B\$22)-
ROW(\$B\$3),N)=0,\$B\$3:B\$22,0))

รวมค่าทุกๆตำแหน่งที่ N ตรงกัน

15 =IF(COUNTIF(\$A\$1:A1,A1)=1,A1,"")

แยกข้อมูลที่ไม่ซ้ำออกมาแสดง

16 =IF(COUNTIF(\$A\$1:\$A\$10,B1)>0,B1,"")

แยกข้อมูลที่ซ้ำกันจากตารางต่างกันออกมาแสดง

17 =IF(COUNTIF(\$A\$1:\$A\$10,B1)=0,B1,"")

แยกข้อมูลที่อยู่ในตารางหนึ่ง ซึ่งไม่ซ้ำกับข้อมูลในอีก
ตารางออกมาแสดง

18 =AVERAGE(IF((A1:A60>=Low)*(A1:A60<=High),A1:A60))

เฉลี่ยค่าที่อยู่ระหว่าง Low ถึง High

19
=AVERAGE(LARGE(A1:A60,ROW(1:10)))
=AVERAGE(LARGE(A1:A60,ROW(INDIRECT("1:10")))))

เฉลี่ยค่าที่มากที่สุด 10 อันดับแรก

ให้เปลี่ยน "1:10" เป็น "1:N" เพื่อกำหนด N ลำดับแรกที่
ต้องการ

20 =AVERAGE(SMALL(A1:A60,ROW(INDIRECT("1:10")))))

เฉลี่ยค่าที่น้อยที่สุด 10 อันดับแรก

ให้เปลี่ยน "1:10" เป็น "1:N" เพื่อกำหนด N ลำดับแรกที่
ต้องการ

21 =SUM(IF(ISERR(data),1,0))

นับจำนวนเซลล์ที่มีค่า error

สูตร	คืนค่า
=SUM(E4:E13*F4,F13)	รวมผลคูณของตาราง E4:E13 คูณด้วย F4:F13
=SUM((E4:E23)*(MOD(ROW(E4:E23)-ROW(E4),2)=0))	รวมค่าแบบบรรทัดเว้นบรรทัด
=SUM((E4:E23)*(MOD(ROW(E4:E23)-ROW(E4),5)=0))	รวมค่าทุกบรรทัดที่ 5 ของแต่ละชุด
22 =SUM(IF(MOD(I4:I23;2)=0,I4:I23,0))	รวมค่าเฉพาะค่าที่เป็นเลขคู่
=SUM(IF(MOD(I4:I23,2)<>0,I4:I23,0))	รวมค่าเฉพาะค่าที่เป็นเลขคี่
=SUM(IF((E4:E23>50)*(H4:H23="North"),E4:E23,0))	รวมค่าเฉพาะค่าแบบและ
=SUM((E5:E24>50)*(H5:H24="North")*E5:E24,0)	รวมค่าเฉพาะค่าแบบและ
=SUM(((H4:H23="North")*E4:E23+(H4:H23="South")*E4:E23)	รวมค่าเฉพาะค่าแบบหรือ

23 =OFFSET(List,MAX(ROW(RevList))-ROW(),0)

กลับลำดับค่าในแนวดิ่ง

24 =OFFSET(RowList,0,MAX(COLUMN(RevRowList))-
COLUMN())

กลับลำดับค่าในแนวนอน

RowList เป็นตารางข้อมูลเดิมตามแนวนอน,
RevRowList เป็นตารางเก็บค่าที่จัดลำดับกลับข้าง

=OFFSET(CList,COLUMN()-MIN(COLUMN(RList)),0)
25 =OFFSET(CList,MAX(COLUMN(RevRList))-
COLUMN(),0)

ตะแคงตารางแนวดิ่ง CList เป็นแนวนอน RList

RevRList เป็นตารางแนวดิ่งที่สลับค่า

=OFFSET(RowList,0,ROW()-MIN(ROW(TXList)))
26 =OFFSET(RowList,0,MAX(ROW(TList))-ROW())

ตะแคงตารางแนวนอน RowList เป็นแนวดัง TXList

TList เป็นตารางแนวนอนที่สลับค่า

```
2 =ADDRESS(MIN(IF(NumRange=MIN(NumRange),ROW(NumRange))
7 ), COLUMN(NumRange),4)
```

Address ของค่าที่น้อยที่สุดค่าแรก

```
2 =ADDRESS(MAX(IF(NumRange=MIN(NumRange),ROW(NumRange)
8 * (NumRange<>"")),
COLUMN(NumRange),4)
```

Address ของค่าที่น้อยที่สุดค่าสุดท้าย

```
2 =ADDRESS(MIN(IF(NumRange=MAX(NumRange),ROW(NumRange)
9 )), COLUMN(NumRange),4)
```

Address ของค่าที่มากที่สุดค่าแรก

```
3 =ADDRESS(MAX(IF(NumRange=MAX(NumRange),ROW(NumRange)
0 * (NumRange<>"")),COLUMN(NumRange),4)
```

Address ของค่าที่มากที่สุดค่าสุดท้าย

3 =INDEX(Rng,MATCH(MAX(COUNTIF(Rng,Rng)),COUNTIF(Rng,Rng),
1 0))

ค่าที่ใช้ซ้ำมากที่สุด

32 =(MAX((DataRange2<>"")*ROW(DataRange2)))-
ROW(DataRange2)+1

นับจำนวนบรรทัดที่มีข้อมูล

3 =INDIRECT(ADDRESS(MAX((DataRange2<>"")*ROW(DataRange2))
3 '
3 COLUMN(DataRange2),4))

คืนค่าสุดท้ายล่างสุดในแถวตั้งแถวแรก

3 =INDIRECT(ADDRESS(MAX((DataRange2<>"")*ROW(DataRange2))
4 '
4 COLUMN(DataRange2)+1,4))

คืนค่าสุดท้ายล่างสุดในแถวตั้งแถวที่สอง

3 =SUM(IF(FREQUENCY(IF(LEN(Range1)>0,MATCH(Range1,Range1,0
5),""),
IF(LEN(Range1)>0,MATCH(Range1,Range1,0),""))>0,1))

นับจำนวนค่าที่เป็น Unique

36 =SUM(1/COUNTIF(A1:A10,A1:A10))

นับจำนวนค่าที่เป็น Unique กรณีไม่มี blank

37 =SUM(N(FREQUENCY(A1:A10,A1:A10)>0))

นับจำนวนค่าที่เป็น Unique ใช้กับตารางตัวเลขหรือ
เซลล์ว่าง

=ADDRESS(ROW(DataRange2),COLUMN(DataRange2),4)&":"&
38 ADDRESS(MAX((DataRange2<>"")*ROW(DataRange2)),
COLUMN(DataRange2)+COLUMNS(DataRange2)-1,4)

แสดง range ที่มีข้อมูล data เช่น DataRange2 เป็นตารางข้อมูล
H7:I25 โดยเซลล์ H7:I17 มีตัวเลขอยู่ นอกจากนั้นเป็นเซลล์ว่าง

=RIGHT(A1,LEN(A1)-
39 MAX(ROW(INDIRECT("1:"&LEN(A1))))
*(MID(A1,ROW(INDIRECT("1:"&LEN(A1))),1)="")
"))

แสดงค่าสุดท้ายขวาสุด

=MID(A10,SMALL(IF(MID(" "&A10,ROW(INDIRECT("1:"&LEN(A10)+1)),1)="",ROW(INDIRECT("1:"&LEN(A10)+1))),
 40 B10),SUM(SMALL(IF(MID(" "&A10&" ",ROW(INDIRECT("1:"&LEN(A10)+2)),1)="",ROW(INDIRECT("1:"&LEN(A10)+2))),
 B10+{0,1})*{-1,1}-1)

แยกคำออกจากประโยค ซึ่งมีเครื่องหมายวรรคแยกแต่ละคำ

=MID(A10,SMALL(IF(MID(" "&A10,ROW(INDIRECT("1:"&LEN(A10)+1)),1)="",ROW(INDIRECT("1:"&LEN(A10)+1))),
 41 B10),SUM(SMALL(IF(MID(" "&A10&" ",ROW(INDIRECT("1:"&LEN(A10)+2)),1)="",ROW(INDIRECT("1:"&LEN(A10)+2))),
 B10+C10*{0,1})*{-1,1}-1)

แยกคำลำดับที่ต้องการออกจากประโยค

A10 เป็นประโยคที่มีคำอยู่

B10 เป็นเลขที่คำเริ่มต้นที่ต้องการ

C10 เป็นจำนวนคำที่ต้องการ เริ่มต้นจาก B10

42 =SUM(IF(WEEKDAY(A2-1+ROW(INDIRECT("1:"&TRUNC(B2-A2)+1)))=C2,1,0))

นับจำนวนวันจันทร์หรือวันอื่นๆ จากช่วงเวลาที่กำหนด

A2 เป็นวันเริ่มต้น

B2 เป็นวันสิ้นสุด

C2 เป็นเลขที่ของวันในสัปดาห์ (1=Sunday,
2=Monday,...,7=Saturday)

43 =IF(ROW()-ROW(NoBlanksRange)+1>ROWS(BlanksRange)-
COUNTBLANK(BlanksRange),"",INDIRECT(ADDRESS(SMALL(
(IF(BlanksRange<>"",ROW(BlanksRange),ROW()+
ROWS(BlanksRange))),
ROW()-ROW(NoBlanksRange)+1),COLUMN(BlanksRange),4)))

ตัดเซลล์ว่างออกจากตาราง BlanksRange ให้เป็นตารางใหม่
NoBlanksRange เหลือเฉพาะเซลล์ที่มีค่าเท่านั้น

44 =SUM(IF(IF(B\$5>=F6:F10,F6:F10-F5:F9,\$B\$5-F5:F9+1)>0,
IF(B\$5>=F6:F10,F6:F10-F5:F9,\$B\$5-F5:F9+1),0)*G5:G9)

Progressive Pricing คำนวณจำนวนเงินทั้งหมด ใช้หลัก FIFO

สูตรนี้มีประโยชน์อย่างมาก ใช้กับปัญหาอื่นๆ เช่น อัตราภาษีแบบ
ก้าวหน้า การตัด Stock แบบ FIFO

	E	F	G	H
4		Quantity	UnitPrice	
5		1	5	

6		10	4	
7		30	3	
8		60	2	
9		100	1	
10		9999999		
11				

	A	B	C	D	E
4	1 Part#	ALPHA portio n	NUMERI C portion		
5	2 ab2	ab	2	{=LEFT(A2, MIN(IF(ISNUMBER(MID(A2,{1,2,3,4,5,6},1) *1),{1,2,3,4,5,6},7))- 1))}	=RIGHT(A2 , LEN(A2) - LEN(B2))*1 1)}

3	OP30	OP	30	{=LEFT(A6, MIN(IF(ISNUMBER(MID(A3,{1,2,3,4,5,6},1) *1),{1,2,3,4,5,6},7)))- 1)}	=RIGHT(A3 , LEN(A6) - LEN(B6))*1
4	OP122 1	OP	1221	{=LEFT(A8, MIN(IF(ISNUMBER(MID(A4,{1,2,3,4,5,6},1) *1),{1,2,3,4,5,6},7)))- 1)}	=RIGHT(A4 , LEN(A8) - LEN(B8))*1

แยกค่าที่เป็นตัวอักษรออกจากตัวเลข

46

=INDIRECT(ADDRESS(ROW(MyRange) +INT((ROW() -
ROW(CxRV))/COLUMNS(MyRange)),
COLUMN(MyRange)+MOD(ROW() -ROW(CxRV),
COLUMNS(MyRange)),4,,"Sheet1"))

=INDIRECT(ADDRESS(ROW(MyRange) +MOD(ROW() -
ROW(CxCV),ROWS(MyRange)),
COLUMN(MyRange)+INT((ROW() -
ROW(CxCV))/ROWS(MyRange)), 4,,"Sheet1"))

=INDIRECT(ADDRESS(ROW(MyRange) +INT((COLUMN() -
COLUMN(RxRV))/COLUMNS(MyRange)),
COLUMN(MyRange)+MOD(COLUMN() -COLUMN(RxRV),
COLUMNS(MyRange)), 4,,"Sheet1"))

```
=INDIRECT(ADDRESS( ROW(MyRange) +MOD(COLUMN() -  
COLUMN(RxCV),ROWS(MyRange)),  
COLUMN(MyRange)+INT((COLUMN() -  
COLUMN(RxCV))/ROWS(MyRange)),4,,"Sheet1"))
```

ตะแคงตารางเป็นแนวตั้ง

MyRange เป็นตารางข้อมูล มีขนาด M rows และ N
columns

CxRV เป็นตารางผลลัพธ์

```
=OFFSET($B$5,MIN(IF(COUNTIF($G$11:G11,$B$5:$GR$10)=0,  
ROW($B$5:$GR$10)-  
ROW($B$5))),MOD(MIN(IF(COUNTIF($G$11:G11,$B$5:$GR$10)=0,  
ROW($B$5:$GR$10)- ROW($B$5) +(COLUMN($B$5:$GR$10)-  
COLUMN($B$5))/1000)),1)*1000)
```

47

```
=INDEX(tbl,MIN(IF(COUNTIF($A$1:A1,tbl)=0,ROW(tbl)-  
MIN(ROW(tbl))+1)),  
MATCH(0,COUNTIF($A$1:A1,INDEX(tbl,MIN(IF(  
COUNTIF($A$1:A1,tbl)=0,ROW(tbl) -MIN(ROW(tbl))+1))),1)),0),1)
```

list of unique entries in an n * m array

48 =AND(NOT(ISNA(MATCH(Range1,Range2,0))))

ตรวจสอบว่าข้อมูลใน Range1 แสดงใน Range2

49

=MMULT(N(ROW(INDIRECT("1:"&ROWS(A)))=TRANSPOSE(
ROWS(A)+1-ROW(INDIRECT("1:"&ROWS(A))))),A)

=MMULT(A,N(ROW(INDIRECT("1:"&COLUMNS(A)))=TRANSPOSE
(COLUMNS(A)+1-ROW(INDIRECT("1:"&COLUMNS(A))))))

กลับข้างตาราง

50

=OR(A1=2,A1=3,ISNA(MATCH(TRUE,A1/ROW(INDIRECT(
"2:"&INT(SQRT(A1))))=INT(A1/ROW(INDIRECT("2:"&INT(SQRT(A1))))),0)))

ตรวจสอบว่าเป็นตัวเลข prime number

51 =MATCH(TRUE,EXACT("A",MyRange),0)

case-sensitive MATCH

52

$$17 = 5x + 3y + 2z$$

$$13 = 2x + 4y + z$$

$$22 = 3x + 2y + 5z$$

Put the coefficients of the unknowns in, e.g., A1:C3 (i.e., 5,3,2 in A1:C1, 2,4,1 in A2:C2, etc.);

Put the constants (17, 13, 22) in, e.g., D1:D3;

Highlight, e.g., E1:E3 and array enter (i.e., enter with Ctrl+Shift+Enter instead of just Enter)

=MMULT(MINVERSE(A1:C3,D1:D3)

and the solution vector (1,2,3) will appear in E1:E3; i.e., x=1, y=2, z=3 For a set of equations that does not have a solution, the #VALUE error will appear in E1:E3.

solve a set of simultaneous equations

=SUM(N(MATCH(MID(A1,ROW(INDIRECT("1:"&LEN(A1))),1),
53 MID(A1,ROW(INDIRECT
("1:"&LEN(A1))),1),0)=ROW(INDIRECT("1:"&LEN(A1))))=LEN(A1)

ตรวจสอบว่าตัวอักษรในเซลล์มีซ้ำกันหรือไม่

54 =SUM(VALUE(MID(A1,ROW(INDIRECT("1:"&LEN(A1))),1))*10
^(ROW(INDIRECT ("1:"&LEN(A1)))-1))

กลับข้างหลักตัวเลข

55 A1:D1 = {1,2,3,4}
A5:D5 = {5,6,7,5}
=SUM({1,2,3,4}*{5,7,6,5}) = 57
=SUM(A1:D1*INDEX(A5:D5,5-COLUMN(A5:D5)))

สูตรเขียนโดยไม่ทราบสาเหตุ เมื่อสร้างสูตรสุดท้ายลงในเซลล์เดียวจะ
คำนวณผิด ต้องสร้างสูตรลงในเซลล์ 2 เซลล์

56 =ROWS(iRng)-
ROW(OFFSET(\$A\$1,,,ROWS(iRng),)).

กลับตำแหน่งเซลล์

57 =IF(T(OFFSET(iRng,ROWS(iRng)-
ROW(OFFSET(\$A\$1,,,ROWS(iRng),)),,,,))="" ,N
(OFFSET(iRng,ROWS(iRng)-
ROW(OFFSET(\$A\$1,,,ROWS(iRng),)),,,,)),T(OFFSET
(iRng,ROWS(iRng)-ROW(OFFSET(\$A\$1,,,ROWS(iRng),)),,,,)))

กลับตารางแนวตั้ง

58 =LARGE(IF(MATCH(Rng,Rng,0)=ROW(Rng)-
MIN(ROW(Rng))+1,Rng,""),N)

Nth largest unique value in a column range

59 =SUM(N(FREQUENCY(IF(lookupRange="specifStr",
MATCH(colRange,colRange,0)), MATCH(colRange,colRange,0))0))

นับจำนวนข้อมูล unique ที่มีตัวอักษรตามกำหนด

60 =SMALL(IF(MATCH(TheList,TheList,0)=ROW(TheList),
ROW(TheList),""),
ROW(INDIRECT("1:"&SUM(N(MATCH(TheList,TheList,0)=
ROW(TheList))))))-1

แยกข้อมูล Unique item ออกมา

61 =IF(SUBTOTAL(3,OFFSET(Rge,ROW(Rge)-
MIN(ROW(Rge)),,1)),Rge,"")

create an array of filtered items in a column list.

62 =IF(MONTH(DATE(y,m,1))<>MONTH(DATE(y,m,1)-
(WEEKDAY(DATE(y,m,1))-1)+{0;1;2;3;4;5}*7+{1,2,3,4,5,6,7}-1),
"", DATE(y,m,1)-(WEEKDAY(DATE(y,m,1))-
1)+{0;1;2;3;4;5}*7+{1,2,3,4,5,6,7}-1)

สร้างปฏิทินของเดือน

63 =SUM((WEEKDAY(DATE(\$B\$2,\$A3,ROW(INDIRECT("1:"&
DAY(DATE(\$B\$2,\$A3+1,0))))))=C\$1)*1)

รวมจำนวนวันของสัปดาห์ C1 ในเดือน A3 และปี B2 ที่
กำหนด

64 =SUM(IF(ISNONTEXT(data),1))

จำนวนเซลล์ที่ไม่เป็นตัวอักษร

65 =SUM(IF(ISTEXT(data),1))

จำนวนเซลล์ที่เป็นตัวอักษร

66 =SUM(IF(ISLOGICAL(data),1))

จำนวนเซลล์ที่เป็น logical

67 =SUM(IF(EXACT(data,text),1))

นับจำนวนเซลล์ ซึ่งมี text ตรงกัน

68 =SUM(IF(LEN(data)-
LEN(SUBSTITUTE(data,text,""))>0,1))

นับจำนวนเซลล์ ซึ่งมี text ตรงกัน แม้บางส่วน

69 =(SUM(LEN(data))-
SUM(LEN(SUBSTITUTE(UPPER(data),UPPER(text),""))))/LEN(text)
=(SUM(LEN(data))-SUM(LEN(SUBSTITUTE(data,text,""))))/LEN(text)

นับจำนวนเซลล์ ซึ่งมี text ตรงกัน ทั้งหมดหรือแม้บางส่วน

70 =INDIRECT(ADDRESS(ROW(Data)+MATCH(MIN(ABS(Target-Data)),ABS(Target-Data),0)-1,COLUMN(Data)+ColOffset))

Closest Match

71 =SUM((MATCH(Grades,{"F","D","C","B","A"},0)-1)*CreditHours)/SUM(CreditHours)

GPA

72 =ADDRESS(ROW(Data)+MATCH(Target,Data,0)-1,COLUMN(Data))

ตำแหน่งเซลล์ที่มีค่าอยู่ในตารางแนวดิ่ง

73 =ADDRESS(ROW(Data),COLUMN(Data)+MATCH(Target,Data,0)-1)

ตำแหน่งเซลล์ที่มีค่าอยู่ในตารางแนวนอน

74 =INDEX(Code,
MATCH(Make&Model,Range1&Range2,0))

2 value lookup

```
=INDEX(Data, SMALL(IF(Data>0,
ROW(INDIRECT("1:"&ROWS(Data)))),
ROW(INDIRECT("1:"&ROWS(Data)))))
```

75 IF(ISERR(SMALL(IF(Data>0,
ROW(INDIRECT("1:"&ROWS(Data))),
ROW(INDIRECT("1:"&ROWS(Data))))) , "" , INDEX(Data,
SMALL(IF(Data>0, ROW(INDIRECT("1:"&ROWS(Data))),
ROW(INDIRECT("1:"&ROWS(Data)))))

สรุปค่าบวก

7 =IF(ISERR(SMALL(IF(Data<>"",ROW(INDIRECT("1:"&ROWS(Data))
6)), ROW(INDIRECT("1:"&ROWS(Data))))) ,
"",INDEX(Data,SMALL(IF(Data<>"",
ROW(INDIRECT("1:"&ROWS(Data))),
ROW(INDIRECT("1:"&ROWS(Data)))))

สรุปค่าที่ไม่ใช่เซลล์ว่าง

77 =IF(INDEX(Data,ROWS(Data) -
ROW(INDIRECT("1:"&ROWS(Data)))+1)="", "",
INDEX(Data,ROWS(Data) -
ROW(INDIRECT("1:"&ROWS(Data)))+1))

กลับตำแหน่งค่าในตารางแนวนั่ง

=LARGE(Data,ROW(INDIRECT("1:"&ROWS(Data))))

78 =IF(ISERR(LARGE(Data, ROW(INDIRECT("1:"&ROWS(Data))))),
"", LARGE(Data, ROW(INDIRECT("1:"&ROWS(Data)))))

เรียงลำดับตัวเลข จากมากไปน้อย

=INDEX(data, SMALL(IF(MATCH(data,data,0) =
79 ROW(INDIRECT("1:"&ROWS(data))), MATCH(data,data,0), ""),
ROW(INDIRECT("1:"&ROWS(data)))))

Unique items in List

80 =SUM(LARGE(Data,ROW(INDIRECT("1:"&n))))

รวมค่าที่มากที่สุด n ลำดับแรก

81 =IF(OR(TheName=NameList),"Found","Not
Found")

ตรวจสอบว่า มีค่า TheName อยู่ใน NameList

82 =SUM(IF(MyData=YourData,0,1))

นับจำนวนเซลล์ที่มีค่าไม่เท่ากัน เมื่อเทียบ 2 ตาราง

83 =ADDRESS(MIN(IF(Data=MAX(Data),ROW(Data),
 "")),COLUMN(Data))

ตำแหน่งเซลล์ซึ่งมีค่าสูงสุด

84 =MIN(IF(Data=MAX(Data),ROW(Data), ""))

เลขที่ row ของเซลล์ซึ่งมีค่าสูงสุด

85 =SMALL(IF(Data=Value,ROW(Data), ""),n)

เลขที่ row ของเซลล์ซึ่งมีค่าซ้ำกันครบ n ครั้ง

86 =INDEX(Data,MATCH(MAX(LEN(Data)),LEN(Data),FALSE),1)

ค่ายาวที่สุดในตาราง

87 =ISNA(MATCH(TRUE,ISNA(MATCH(MyList,Master,0)),0))

ตรวจสอบว่า ค่าใน MyList มีอยู่ใน Master

88 =SUM(1*ISNA(MATCH(MyList,Master,0)))

จำนวนเซลล์ของ MyList ซึ่งไม่มีอยู่ใน Master

89 =INDEX(MyList,MATCH(TRUE,ISNA(MATCH(MyList,Master,0)),0))

ค่าแรกของ MyList ซึ่งไม่มีอยู่ใน Master

90 =SUM(VALUE(MID(ABS(A1),ROW(INDIRECT
("1:"&LEN(ABS(A1)))),1)))

รวมตัวเลขทุกหลักในค่าจำนวนเต็ม

91 =MID(A4,MATCH(0,(ISERROR(MID(A4,
ROW(INDIRECT("1:"&LEN(A4))),1)*1)*1),0),LEN(A4)-
SUM((ISERROR(MID(A4, ROW(INDIRECT(
"1:"&LEN(A4))),1)*1)*1)))

แยกเฉพาะส่วนตัวเลขออกจากตัวอักษร

92 =INDEX(Data,MATCH(SMALL(ABS(Target-
Data),1),ABS(Target-Data),0))

ค่าที่ใกล้เคียงที่สุดกับค่าที่ต้องการ

93 =INDIRECT(ADDRESS(MAX((ROW(4:503)*
(A4:A503<>""))),COLUMN(A:A)))

ค่าล่างสุดใน column A แม้จะไม่ได้อยู่ติดกัน

94 =INDIRECT(ADDRESS(6,(MAX((TRANPOSE(
ROW(1:256))*(6:6<>""))))))

ค่าขวาสุดใน row 6 แม้จะไม่ได้ยู่ติดกัน

95 =SUM(1*(B6<=Sales))-(SUM(1*(B6=Sales))-1)/2

ลำดับที่ของค่า โดยใช้ลำดับแบบเฉลี่ยกรณีซ้ำกัน ต่าง
จากการใช้สูตร =RANK(B6,Sales)

96 =SUM(IF(\$E6&F\$5=Date&Category,Amount))

Dynamic crosstabulation

97 =LEN(A1)+1-MATCH(" ",MID(A1,LEN(A1)+1-
ROW(INDIRECT("1:"&LEN(A1))),1),0)

ตำแหน่ง space ขวาสุด

98 =SUM((SUBTOTAL(9, INDIRECT(ADDRESS(ROW(E3),COLUMN(E3)
)&":"& ADDRESS(ROW(E3),
ROW(INDIRECT(COLUMN(E3)&":"&COLUMN(AI3))))))<=B3)*1)+1

จำนวนเซลล์ ซึ่งยอดรวม Cumulative Sum ค่าในเซลล์ C3:G3 มีค่า
มากกว่าหรือเท่ากับ ค่าในเซลล์ B3

99 =INDEX(ค่าทั้งหมด,SMALL(IF(ค่าทั้งหมด<0,ลำดับตัวเลข),ลำดับ
ตัวเลข),1)

ค่าติดลบ

100 =INDEX(Product, SMALL(IF((Type=YourType) *
(Account=YourAcct) * (Date>=YourDate) * ROW(INDIRECT("1:"&
ROWS(Product)))<>0, (Type=YourType) * (Account=YourAcct) *
(Date>=YourDate) * ROW(INDIRECT("1:"&ROWS(Product)))),
ROW(INDIRECT("1:"&ROWS(Product)))),1)

แยกข้อมูลซึ่งมีประเภท เลขที่บัญชี วันที่ ตามต้องการ

101 =INDEX(Type, LARGE(IF(NOT(ISBLANK(Type)),
MATCH(Type,Type,0)) * (IF(NOT(ISBLANK(Type)),
MATCH(Type,Type,0))=ROW(INDIRECT("1:"&ROWS(Type)))),
ROW(INDIRECT("1:"&SUM(IF(NOT(ISBLANK(Type)),
1/COUNTIF(Type,Type))))))))

Unique List

102 =INDEX(Product, MATCH(MAX(COUNTIF(Product,Product)),
COUNTIF(Product,Product), 0),1)

ค่าซึ่งใช้บ่อยที่สุด

การกำหนดขอบเขตตารางแบบยืดหยุ่น

งานที่สร้างขึ้นด้วย Excel ไม่ได้ใช้ตำแหน่งอ้างอิง (Reference) เพื่อใช้ในสูตรคำนวณแต่เพียงอย่างเดียว ตำแหน่งอ้างอิงยังถูกเรียกใช้ในคำสั่ง Conditional Formatting, Data Validation, Advanced Filter, Pivot Table, หรือในคำสั่งอื่นๆ แม้แต่ในรหัส Visual Basic for Applications (VBA) ซึ่งโดยทั่วไปเมื่อมีการ Insert หรือ Delete row หรือ column หรือโยกย้ายตำแหน่งตาราง จะส่งผลให้ตำแหน่งอ้างอิงที่ใช้เดิมปรับเปลี่ยนขนาดตามทันที และหากรู้จักใช้ชื่อในการกำหนดตำแหน่งอ้างอิงจะช่วยทำให้กำหนดตำแหน่งอ้างอิงในสูตรได้ง่าย และยังทำให้ตำแหน่งอ้างอิงที่ใช้ในกราฟหรือ VBA ปรับขนาดตามได้อีกด้วย ช่วยทำให้ไม่ต้องเสียเวลาสร้างกราฟใหม่หรือต้องย้อนกลับมาแก้ไขรหัส VBA อีกแม้แต่บ่อย

ขอบเขตตารางแบบยืดหยุ่น หมายถึง ขอบเขตตารางที่สามารถยืดได้หดได้ตามจำนวนรายการหรือปริมาณข้อมูลที่มีเพิ่มลด ทำให้ไม่ต้องใช้วิธีค่อยๆ Insert เพิ่ม row เพื่อขยายพื้นที่ตารางให้มีเซลล์ว่างเตรียมไว้สำหรับข้อมูลในอนาคต หรือไม่ต้องกำหนดขอบเขตตารางแบบฟุ่มเฟือยโดยสร้างสูตรอ้างอิงกับทั้ง column เช่น

=VLookup(IDCode, A:D, 3)

ซึ่งการกำหนดตำแหน่งอ้างอิง A:D เช่นนี้จะทำให้สูตรทำงานช้าลงอย่างมากและยังทำให้แฟ้มมีขนาดใหญ่ขึ้นโดยไม่จำเป็นอีกด้วย

ในบทความนี้ขอแนะนำวิธีการกำหนดขอบเขตตารางแบบยืดหยุ่น ตั้งแต่วิธี Dynamic Cell ไปจนถึงวิธี Dynamic Range ซึ่งใช้สูตร Offset, Index, Indirect, Address ช่วยในการกำหนดขอบเขตตารางแบบอัตโนมัติ

วิธีการหนดตำแหน่งเซลล์แบบยืดหยุ่น (Dynamic Cell)

คนทั่วไปมักมองข้ามเรื่องพื้นฐานที่นึกว่าไม่สำคัญหรือไม่เคยคิดว่า Excel ทำได้ไปใช้สูตร Offset กำหนดขอบเขตตารางกันเลย โดยหารู้ไม่ว่าเมื่อนำตำแหน่งเซลล์ของเซลล์เพียงเซลล์เดียวหรือเซลล์ในแนวเดียวกันมาใช้ให้เป็น จะทำให้เกิดขอบเขตตารางแบบยืดหยุ่นได้เช่นกัน แถมเป็นวิธีที่ง่ายกว่าเสียอีกเพราะไม่จำเป็นต้องใช้สูตร Offset ให้ยุ่งยาก

ตัวอย่าง

	A	B	C	D	E	F	G
1							
2			2	3	4	5	
3		10	20	30	40	50	#VALUE!
4		20	40	60	80	100	#VALUE!
5		30	60	90	120	150	#VALUE!
6			#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!
7			#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!
8							
9		Top	=\$C\$2:\$F\$2				
10		Left	=\$B\$3:\$B\$5				

ตัวอย่างนี้เป็นตารางสูตรคูณซึ่งนำตัวเลข 2, 3, 4, 5 บนหัวตารางมาคูณกับตัวเลข 10, 20, 30 ด้านข้างซ้ายของตาราง โดยใช้วิธีตั้งชื่อเซลล์ C2:F2 บนหัวตารางว่า Top และตั้งชื่อเซลล์ B3:B5 ด้านซ้ายของตารางว่า Left

ในเซลล์ C3 สร้างสูตร =Top*Left แล้วเมื่อ copy เซลล์ C3 ไป paste ลงไปในเซลล์ตั้งแต่ C3:G7 ย่อมพบว่าสูตร =Top*Left สามารถคำนวณหาผลคูณได้ถูกต้องเฉพาะพื้นที่ C3:F5 ซึ่งเป็นตารางที่อยู่ในขอบเขตแนวคู่ขนานตัดกันของ Top และ Left เท่านั้น ส่วนสูตรในเซลล์นอกแนวคู่ขนานของ Top กับ Left ให้คำตอบเป็น error ว่า #VALUE!

แทนที่จะดัดแปลง Top กับ Left ให้เป็น Dynamic Range ด้วยสูตร Offset เพื่อทำให้ Top กับ Left ขยายขอบเขตให้มีขนาดตามค่าตัวเลขบนหัวตารางและตัวเลขข้างซ้ายของตารางที่อาจจะเพิ่มขึ้น ให้ดัดแปลงตำแหน่งอ้างอิงของ Top กับ Left ให้เป็น Dynamic Cell โดยมีขั้นตอนดังนี้

1. คลิกเลือกเซลล์ C3
2. เลือกคำสั่ง Formulas > Name Manger
3. แก่ตำแหน่งอ้างอิงของ Top =C\$2 และ Left =\$B3

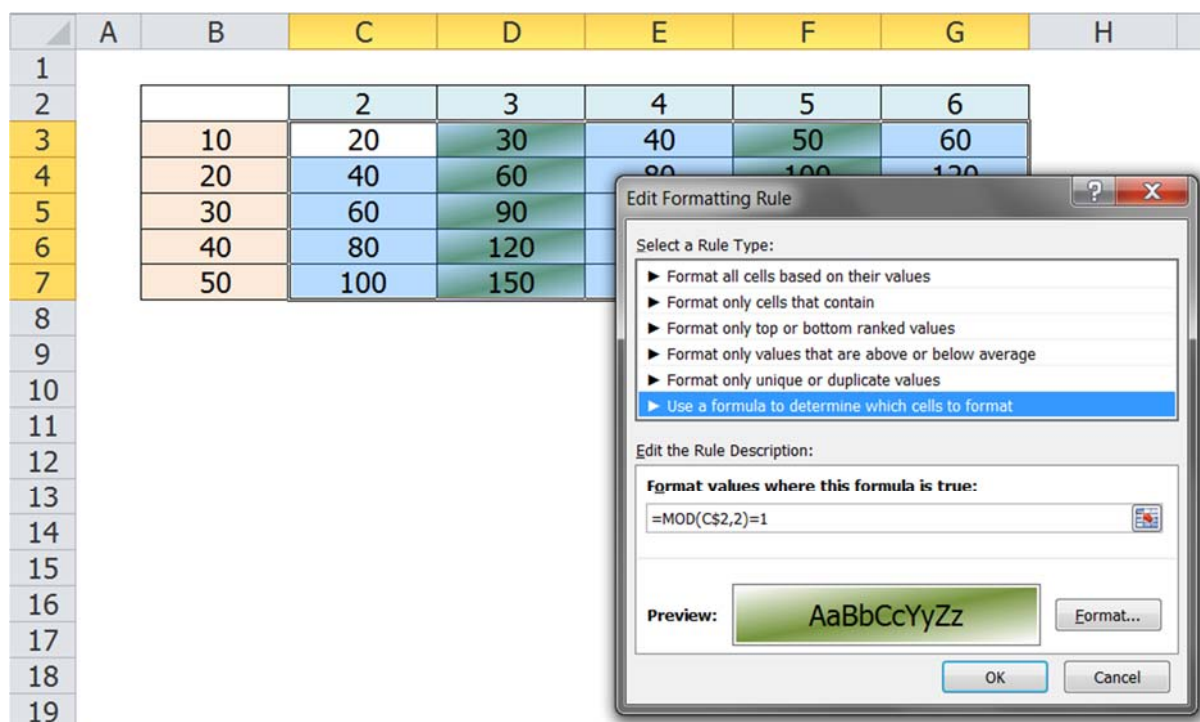
C3		=Top*Left					
	A	B	C	D	E	F	G
1							
2			2	3	4	5	6
3		10	20	30	40	50	60
4		20	40	60	80	100	120
5		30	60	90	120	150	180
6		40	80	120	160	200	240
7		50	100	150	200	250	300
8							
9		Top	=C\$2				
10		Left	=\$B3				

เมื่อ Top =C\$2 จะทำให้ Excel ถือว่าทุกเซลล์ที่อยู่ใน row 2 มีชื่อว่า Top และเมื่อ Left =\$B3 ย่อมหมายถึงทุกเซลล์ใน column B มีชื่อว่า Left ทำให้สามารถใช้สูตร Top*Left ได้ทุกที่ในชีทโดยไม่ต้องกำหนดขอบเขตตารางของ Top กับ Left ตามขนาดข้อมูลอีกต่อไป

หากต้องการย้ายตารางสูตรคุณนี้ไปยัง row อื่น ไม่สามารถใช้วิธี Cut เฉพาะพื้นที่ตารางสูตรคุณ B2:G7 แต่ต้องเลือก Cut row 2:7 ไปวางที่ row อื่น หรือใช้วิธี Insert row เพื่อขยับตำแหน่งอ้างอิงของ Top ทั้งแนว row ที่ 2 ไปพร้อมกัน เพราะ Top มีได้หมายถึงแค่พื้นที่บางส่วนของ หากครอบคลุมพื้นที่ทั้ง row ที่ 2 ทั้งหมด

ตัวอย่าง

ในกรณีที่ต้องการเปลี่ยนสี column ในตารางสูตรคุณให้มีสีเฉพาะแนวของ column ที่อยู่ได้ ตัวเลขบนหัวตารางที่เป็นเลขคี่



ให้เลือกพื้นที่ตัวเลขผลคูณตั้งแต่ C3:G7 แล้วใช้คำสั่ง Home > Conditional Formatting
แบบ Use a formula to determine which cells to format แล้วใช้สูตรต่อไปนี้

$$=MOD(C\$2,2)=1$$

ขอให้สังเกตตำแหน่งเซลล์ C\$2 ในสูตรนี้ว่าเป็นเซลล์แรกบนหัวตารางด้านบนซึ่งอยู่ในแนว
คู่ขนานกับพื้นที่ C3:G7 ที่เลือกไว้ และต้องกำหนดให้ใส่เครื่องหมาย \$ ไว้หน้าเลข 2 แต่ไม่ได้
ใส่ \$ หน้าตัว C เพื่อควบคุมให้ Excel นำตำแหน่งเซลล์ถัดไปใน row ที่ 2 มาใช้ในการคำนวณ
ซึ่งได้แก่เซลล์ C2, D2, E2, F2, และ G2 นั่นคือตัวเลข 2, 3, 4, 5, 6 ตามลำดับ ซึ่งเมื่อนำมา
หารด้วยเลข 2 แล้ว หากเหลือเศษเท่ากับ 1 ย่อมแสดงว่าเลขในเซลล์บนหัวตารางด้านบน
เซลล์นั้นเป็นเลขคี่ ทำให้ Conditional Formatting เปลี่ยนสีเซลล์ผลคูณที่อยู่ใต้เลข 3 และ 5
ซึ่งเป็นเลขคี่

ประโยชน์ของวิธีทำให้เซลล์ C\$2 เป็น Dynamic Cell จะช่วยให้ใช้คำสั่ง Conditional
Formatting กำหนดเงื่อนไขให้กับทั้งตารางได้พร้อมกันในทีเดียว โดยไม่จำเป็นต้องเสียเวลา
ใช้คำสั่งนี้กับทีละเซลล์หรือทีละ column และยังสามารถนำหลักการนี้ไปใช้กับคำสั่ง Data
Validation ที่กำหนด Allow แบบ Custom โดยใช้สูตรตรวจสอบการบันทึกค่าได้เช่นกัน

วิธีใช้ Table กำหนดขอบเขตตารางโดยอัตโนมัติ

ตามปกติเมื่อสร้างสูตรอ้างอิงกับตำแหน่งเซลล์หรือตำแหน่งตาราง จากนั้นเมื่อมีการย้ายหรือ Insert/Delete ทำให้เซลล์หรือตารางนั้นเปลี่ยนตำแหน่งหรือมีขนาดพื้นที่ต่างไปจากเดิม จะพบว่าตำแหน่งอ้างอิงที่กำหนดไว้ในสูตรมีการปรับตำแหน่งตามให้เอง แต่ถ้ามีการเปลี่ยนแปลงเฉพาะจำนวนรายการของข้อมูลในตารางนั้นให้มีจำนวนรายการเพิ่มลด จะพบว่าตำแหน่งอ้างอิงที่กำหนดไว้ในสูตร มิได้ขยายหรือลดขอบเขตตารางตามปริมาณข้อมูลที่มีอยู่แต่อย่างใด ผู้ใช้ Excel ทั่วไปจึงมักแก้ไขโดยสร้างสูตรที่มีขอบเขตตารางเพื่อไว้ บางคนถึงกับใช้ตำแหน่งอ้างอิงทั้ง column เพื่อไว้ ส่งผลให้แฟ้มนั้นมีขนาดใหญ่และคำนวณช้าลงผิดปกติ

ใน Excel 2003 และรุ่นก่อนนั้น มีคำสั่ง Data > List ไว้ใช้ช่วยแก้ปัญหาดังกล่าว ส่วนใน Excel 2010 ได้ตั้งชื่อคำสั่งใหม่เป็น Table (เป็นคำสั่งที่ต่างจาก Data Table) โดยสั่ง Insert > Table หรือ Home > Format as Table ทำให้ Excel ปรับขอบเขตตารางตามปริมาณข้อมูลที่เพิ่มให้โดยอัตโนมัติ ช่วยทำให้ตำแหน่งอ้างอิงที่กำหนดไว้ในสูตร, Format, Conditional Formatting, Data Validation, หรือแม้แต่ใน Pivot Table ปรับขนาดตามให้ทันที

ตารางฐานข้อมูล				ตารางค้นหาข้อมูล		
Id	Name	Amount	Cost	Id	Name	Amount
a001	a	<10.00>	1.43	a001	a	10

Table2	Formula
MyData	=B\$3:\$E\$3
Amount	=B\$3:\$D\$3
Name	=D\$3:\$D\$4
Id	=C\$3:\$C\$4

ก่อนที่จะใช้คำสั่ง Table ควรตั้งชื่อ Range Name ให้กับตารางฐานข้อมูลทางด้านซ้ายมือของภาพนี้ไว้ก่อน โปรดสังเกตว่า Range Name แต่ละชื่อต้องมีขนาดอย่างน้อย 2 เซลล์ขึ้นไป (ตามภาพนี้คือ MyData, Id, Name, Amount และสาเหตุที่ไม่ตั้งชื่อให้กับเซลล์ของ Cost เพื่อพิสูจน์ว่าไม่จำเป็นต้องตั้งชื่อไว้ก่อนก็ได้) จึงจะทำให้ Table ขยายขอบเขตของชื่อตารางตามปริมาณข้อมูลที่เพิ่มขึ้น

เมื่อตั้งชื่อ Range Name ไว้ก่อนเรียบร้อยแล้ว ให้คลิกลงไปตารางฐานข้อมูลแล้วสั่ง Insert > Table โดย Excel จะเลือก =\$B\$2:\$E\$3 เป็นขอบเขตตารางให้เองพร้อมกับกาช่อง My table has headers ไว้ด้วยแล้ว เมื่อกดปุ่ม OK จะเกิด Range Name ชื่อ Table2 มีตำแหน่ง

อ้างอิง =\$B\$3:\$E\$3 สร้างขึ้นโดยอัตโนมัติ (ชื่อ Table2 นี้ จะมีเลขต่อท้ายเพิ่มขึ้นทีละ 1 ตามจำนวนครั้งที่ใส่คำสั่ง Table)

เมื่อบันทึกข้อมูลเพิ่มขึ้น จะพบว่าชื่อ Range Name ทั้งหมดมีขอบเขตของตำแหน่งอ้างอิงขยายตามให้อย่างอัตโนมัติ ช่วยทำให้ตำแหน่งอ้างอิงที่กำหนดไว้ในสูตร, Format, Conditional Formatting, Data Validation, หรือแม้แต่ใน Pivot Table ปรับขนาดตามให้ทันที

	A	B	C	D	E	F	G	H	I	J	K
1		ตารางฐานข้อมูล						ตารางค้นหาข้อมูล			
2		Id	Name	Amount	Cost			Id	Name	Amount	
3		a001	a	<10.00>	1.43			a003	c	30	
4		a002	b	<20.00>	2.86						
5		a003	c	<30.00>	4.29						
6		a004	d	<40.00>	5.71						
7		a005	e	<50.00>	7.14						
8											
9											
10											
11											
12											

Table2	=\$B\$3:\$E\$7
MyData	=\$B\$3:\$D\$7
Amount	=\$D\$3:\$D\$8
Name	=\$C\$3:\$C\$8
Id	=\$B\$3:\$B\$8

แม้คำสั่ง Table จะมีประโยชน์อย่างมาก แต่ก็เหมาะกับตารางเก็บข้อมูลซึ่งบันทึกไว้เป็นฐานข้อมูลที่ดีเท่านั้น และผู้ใช้งานควรทราบเพิ่มเติมอีกว่า คำสั่งนี้จะทำให้เกิดตำแหน่งอ้างอิงแบบพิเศษที่เรียกว่า Structured Reference มาใช้แทนตำแหน่งอ้างอิงแบบปกติ เช่น =SUM(Table2[Cost]) จะคืนค่าเป็นยอดรวมของ Cost หากต่อมามีการสังยกเลิก Table จะทำให้สูตรที่เคยใช้ Structured Reference แก่กลับมาใช้ตำแหน่งอ้างอิงตามปกติให้เอง ทำให้สูตร =SUM(Table2[Cost]) แก่กลับมาเป็น =SUM(Sheet1!\$E\$3:\$E\$7)

การยกเลิก Table ทำได้ง่ายๆโดยคลิกขวาเข้าไปในตารางที่เป็น Table แล้วสั่ง Table > Convert to range แต่ถ้าเปลี่ยนใจสั่ง Insert > Table ขึ้นมาใหม่ จะพบว่าสูตร =SUM(Sheet1!\$E\$3:\$E\$7) ไม่เปลี่ยนกลับไปเป็น =SUM(Table2[Cost]) แต่อย่างใด ดังนั้นจึงขอแนะนำให้ใช้ Range Name ที่ตั้งชื่อขึ้นมาเองไว้ก่อน จะยืดหยุ่นและถาวรกว่าการใช้ Structured Reference

นอกจากนี้โปรดสังเกตว่า ในตัวอย่างนี้ก่อนการใช้คำสั่ง Table มีข้อมูลบันทึกไว้เพียงรายการเดียว จึงจำเป็นต้องตั้งชื่อ Range Name ชื่อ Id, Name, และ Amount ให้กับเซลล์ B3:B4, C3:C4, และ D3:D4 ซึ่งเผื่อไว้ถึง row 4 ที่เป็นเซลล์ว่าง ทำให้สูตรที่ใช้ตำแหน่งอ้างอิงรวมถึงเซลล์ว่างนี้อาจทำงานผิดพลาดก็ได้ ดังนั้นเพื่อป้องกันปัญหานี้ ควรใช้คำสั่ง Table กับตารางที่บันทึกข้อมูลไว้ตั้งแต่ 2 รายการขึ้นไป และตั้งชื่อ Range Name ให้กำหนดขอบเขตไว้เท่ากับปริมาณข้อมูลที่มีจริงโดยไม่ต้องเผื่อเซลล์ว่างไว้ด้วย

อย่างไรก็ตามหากมีการลบเฉพาะตัวข้อมูลทิ้งโดยไม่ได้สั่ง Delete Row จะพบว่าขอบเขตตารางยังคงมีขนาดคงเดิม ทั้ง Range Name และตำแหน่งอ้างอิงที่ใช้ในสูตร ไม่ได้ลดขนาดตามจำนวนรายการที่ลดลงแต่อย่างใด จึงเป็นข้อแม้สำคัญของการใช้ Table ที่จะช่วยขยายขอบเขตตารางตามปริมาณข้อมูลที่เพิ่มโดยอัตโนมัติ แต่ไม่ได้ลดขนาดตามปริมาณข้อมูลที่ลดลง จึงขอแนะนำให้ใช้ Table เมื่อจะเพิ่มรายการ จากนั้นให้คืนสภาพเป็นพื้นที่ธรรมดา

วิธีใช้สูตร IF หรือ Choose กำหนดขอบเขตตาราง

ตามปกติในสูตร VLookup, Match, Offset, หรือสูตรด้านฐานข้อมูลใดๆ สามารถรับตำแหน่งตารางได้เพียงตารางเดียวในวงเล็บของสูตร เช่น สูตร =VLOOKUP(\$H\$3,MyData,2,0) ใช้ Range Name ชื่อ MyData เป็นตำแหน่งตารางฐานข้อมูลที่ต้องการค้นหาคำตอบ ถ้าต้องการทำให้สูตร VLookup หรือสูตรฐานข้อมูลสูตรใดสูตรเดียวสามารถค้นหาข้อมูลจากหลากหลายตาราง ให้ใช้สูตร IF หรือ Choose ซ้อนเข้าไปในส่วนของ MyData

สมมติว่า มีตารางฐานข้อมูลของลูกค้าหลายๆกลุ่มตามเกรดของลูกค้า ซึ่งลักษณะของตารางฐานข้อมูลของแต่ละกลุ่มมีหน้าตาตารางเหมือนกัน เพียงแต่จัดเก็บแยกตารางไว้ต่างพื้นที่ (หรือต่างชีทกัน) แล้วตั้งชื่อตารางว่า ClassA, ClassB, และ ClassC

C14 =VLOOKUP(\$B\$14, IF(Class="A", ClassA, IF(Class="B", ClassB, ClassC)), 2, 0)											
A	B	C	D	E	F	G	H	I	J	K	L
1	ClassA				ClassB				ClassC		
2	Id	Name	Amount		Id	Name	Amount		Id	Name	Amount
3	a001	a	10		b001	f	11		c001	j	111
4	a002	b	20		b002	g	22		c002	k	222
5	a003	c	30		b003	h	33		c003	l	333
6	a004	d	40		b004	i	44		c004	m	444
7	a005	e	50						c005	n	555
8									c006	o	666
9	Class										
10	B								Class	=B\$10	
11									ClassA	=B\$3:D\$7	
12	ตารางคำตอบ								ClassB	=F\$3:H\$6	
13	Id	Name	Amount						ClassC	=J\$3:L\$8	
14	b002	g	22								
15											
16	C14 : '=VLOOKUP(\$B\$14, IF(Class="A", ClassA, IF(Class="B", ClassB, ClassC)), 2, 0)										
17	D14 : '=VLOOKUP(\$B\$14, IF(Class="A", ClassA, IF(Class="B", ClassB, ClassC)), 3, 0)										

เมื่อต้องการค้นหาข้อมูล ให้พิมพ์ชื่อ Class A, B, หรือ C ลงไปในเซลล์ B10 ซึ่งตั้งชื่อว่า Class แล้วพิมพ์รหัสลูกค้าลงไปในเซลล์ B14 จะสามารถค้นหาข้อมูลของ Name และ Amount โดยใช้สูตร IF(Class="A", ClassA, IF(Class="B", ClassB, ClassC)) ซ้อนลงไปในสูตร

VLookup เพื่อให้สามารถเลือกได้ว่าหาข้อมูลจากตารางใดตามเงื่อนไขชื่อ Class กลายเป็นสูตรดังนี้

ในเซลล์ C14

=VLOOKUP(\$B\$14, IF(Class="A", ClassA, IF(Class="B", ClassB, ClassC)), 2, 0)

ในเซลล์ D14

=VLOOKUP(\$B\$14, IF(Class="A", ClassA, IF(Class="B", ClassB, ClassC)), 3, 0)

หากไม่ต้องการเสียเวลาพิมพ์ชื่อ Class เพื่อเพิ่มความสะดวกให้ใช้รหัสในเซลล์ B14 เลือกค้นหาข้อมูลได้ทันที ให้ใช้สูตร =Left(C14) สร้างลงไปในเซลล์ B10 ซึ่งตั้งชื่อว่า Class เพื่อตั้งตัวอักษรที่อยู่ท้ายสุดของรหัสมาใช้เป็นชื่อ Class ดังนั้นการตั้งชื่อรหัสให้มีความหมายจะทำให้เกิดประโยชน์ช่วยลดขั้นตอนในการทำงานอย่างมาก

การใช้สูตร IF เพื่อเลือกพื้นที่ตารางหลายๆแห่งมาใช้ จะกลายเป็นสูตร IF ซ้อนกัน ทำให้สร้างขึ้นมาได้ยากและกลายเป็นสูตรยาวขึ้นเรื่อยๆ จึงแนะนำให้ใช้สูตร Choose แทน IF แต่ต้องใช้ตัวเลขที่ตาราง 1, 2, หรือ 3 มาใช้เป็นเงื่อนไขแทนชื่อตาราง A, B, หรือ C

C14 =VLOOKUP(\$B\$14, CHOOSE(Class,Class1,Class2,Class3), 2, 0)										
A	B	C	D	E	F	G	H	I	J	K
1	Class1				Class2				Class3	
2	Id	Name	Amount		Id	Name	Amount		Id	Name
3	a001	a	10		b001	f	11		c001	j
4	a002	b	20		b002	g	22		c002	k
5	a003	c	30		b003	h	33		c003	l
6	a004	d	40		b004	i	44		c004	m
7	a005	e	50						c005	n
8									c006	o
9	Class									
10	2								Class	=\$B\$10
11									Class1	=\$B\$3:\$D\$7
12	ตารางคำตอบ								Class2	=\$F\$3:\$H\$6
13	Id	Name	Amount						Class3	=\$J\$3:\$L\$8
14	b002	g	22							
15										
16	C14 : '=VLOOKUP(\$B\$14, CHOOSE(Class,Class1,Class2,Class3), 2, 0)									
17	D14 : '=VLOOKUP(\$B\$14, CHOOSE(Class,Class1,Class2,Class3), 3, 0)									

เมื่อต้องการค้นหาข้อมูล ให้พิมพ์เลขที่ Class 1, 2, หรือ 3 ลงไปในเซลล์ B10 ซึ่งตั้งชื่อว่า Class แล้วพิมพ์รหัสลูกคาลงไปในเซลล์ B14 จะสามารถค้นหาข้อมูลของ Name และ Amount โดยใช้สูตร CHOOSE(Class,Class1,Class2,Class3) ซ้อนลงไปในสูตร VLookup เพื่อให้สามารถเลือกได้ว่าหาข้อมูลจากตารางใดตามเงื่อนไขเลขที่ Class กลายเป็นสูตรดังนี้

ในเซลล์ C14

=VLOOKUP(\$B\$14, CHOOSE(Class,Class1,Class2,Class3), 2, 0)

ในเซลล์ D14

=VLOOKUP(\$B\$14, CHOOSE(Class,Class1,Class2,Class3), 3, 0)

หากไม่ต้องการเสียเวลาพิมพ์เลขที่ Class เพื่อเพิ่มความสะดวกให้ใช้รหัสในเซลล์ B14 เลือกค้นหาข้อมูลได้ทันที ต้องสร้างตารางสรุปความสัมพันธ์ระหว่างรหัสกับเลขที่ Class ขึ้นมาใช้งานตามภาพต่อไปนี้คือเซลล์ F10:G12 โดยตั้งชื่อว่า ClassTable ซึ่งมี column ซ้ายสุดเรียงลำดับเลขรหัส Id เฉพาะรหัสตัวแรกของแต่ละกลุ่ม แล้วให้ใช้สูตร VLookup แบบ Approaching Match =VLOOKUP(B14,ClassTable,2) สร้างลงไปในเซลล์ B10 ซึ่งมีชื่อว่า Class

Class =VLOOKUP(B14,ClassTable,2)												
A	B	C	D	E	F	G	H	I	J	K	L	
1	Class1			Class2			Class3					
2	Id	Name	Amount	Id	Name	Amount		Id	Name	Amount		
3	a001	a	10	b001	f	11		c001	j	111		
4	a002	b	20	b002	g	22		c002	k	222		
5	a003	c	30	b003	h	33		c003	l	333		
6	a004	d	40	b004	i	44		c004	m	444		
7	a005	e	50					c005	n	555		
8								c006	o	666		
9	Class			ClassTable								
10	2			Id	Class				Class	=B\$10		
11				a001	1				Class1	=B\$3:\$D\$7		
12	ตารางคำตอบ			b001	2				Class2	=F\$3:\$H\$6		
13	Id	Name	Amount	c001	3				Class3	=J\$3:\$L\$8		
14	b002	g	22							ClassTable	=F\$10:\$G\$12	
15												
16	C14 : '=VLOOKUP(\$B\$14, CHOOSE(Class,Class1,Class2,Class3), 2, 0)											
17	D14 : =VLOOKUP(\$B\$14, CHOOSE(Class,Class1,Class2,Class3), 3, 0)											

วิธีใช้สูตร Index กำหนดขอบเขตตาราง

สูตร Index เป็นสูตรทำหน้าที่ดึงค่าออกมาจากตารางที่กำหนดตามตำแหน่ง row และ column ที่ต้องการ เช่น ตามภาพตัวอย่างต่อไปนี้ =Index(Class1,3,2) จะดึงชื่อ c ออกมาให้เพราะอยู่ในตำแหน่ง row ที่ 3 ตัดกับ column ที่ 2 ภายในพื้นที่ตารางชื่อ Class1 หรือถ้าไม่ต้องการกำหนดตำแหน่ง row หรือ col ก็ให้กำหนดเลขตำแหน่งรายการที่ต้องการแทนเพียงค่าเดียว เช่น =Index(CustId1,3) จะดึงรหัส a003 ออกมาเพราะรหัสนี้มีตำแหน่งเป็นลำดับที่ 3 ในตารางชื่อ CustId1 (ทั้งนี้ CustId1 มีลักษณะเป็น Single Column จากเซลล์ B3:B7)

C11 =INDEX((Class1,Class2,Class3), TrNum, 2, Class)												
	A	B	C	D	E	F	G	H	I	J	K	L
1	Class1				Class2				Class3			
2		Id	Name	Amount		Id	Name	Amount		Id	Name	Amount
3		a001	a	10		b001	f	11		c001	j	111
4		a002	b	20		b002	g	22		c002	k	222
5		a003	c	30		b003	h	33		c003	l	333
6		a004	d	40		b004	i	44		c004	m	444
7		a005	e	50						c005	n	555
8										c006	o	666
9	ตารางค่าตอบ					ClassTable						
10		Id	Name	Amount		Id	Class					
11		b002	g	22		a001	1			Class1	=B\$3:\$D\$7	
12						b001	2			Class2	=F\$3:\$H\$6	
13		Class	TRNum			c001	3			Class3	=J\$3:\$L\$8	
14		2	2							ClassTable	=F\$10:\$G\$12	
15										CustId1	=B\$3:\$B\$7	
16		B14 : =VLOOKUP(B11,ClassTable,2)								CustId2	=F\$3:\$F\$6	
17		C14 : =MATCH(B11, CHOOSE(Class,CustId1,CustId2,CustId3), 0)								CustId3	=J\$3:\$J\$8	
18		C11 : '=INDEX((Class1,Class2,Class3), TrNum, 2, Class)								Class	=B\$14	
19		D11 : =INDEX((Class1,Class2,Class3), TrNum, 3, Class)								TrNum	=C\$14	

สูตร Index ยังมีโครงสร้างสูตรแบบพิเศษ ซึ่งช่วยเลือกตำแหน่งตารางที่ต้องการได้ทันทีโดยไม่ต้องใช้สูตร If หรือ Choose เข้ามาช่วย เช่น

สูตรในเซลล์ C11 ใช้หาชื่อลูกค้า g ตามรหัส b002 ที่พิมพ์ไว้ในเซลล์ B11

=INDEX((Class1,Class2,Class3), TrNum, 2, Class)

- ในสูตรนี้ พื้นที่ตารางส่วนของ (Class1,Class2,Class3) เป็น 3 ตารางที่ต้องการเลือก
- TrNum เป็นเลขที่ตำแหน่งรายการของรหัส b002 ที่พิมพ์ไว้ในเซลล์ B11 มาจากสูตร
=MATCH(B11, CHOOSE(Class, CustId1,CustId2,CustId3), 0)
- เลข 2 คือ เลขชี้ตำแหน่ง column ของชื่อลูกค้า
- Class เป็นเลขที่ตารางซึ่งหามาจาก ClassTable ด้วยสูตร
=VLOOKUP(B11, ClassTable, 2)
- CHOOSE(Class, CustId1,CustId2,CustId3) ทำหน้าที่เลือกตำแหน่งตารางรหัสลูกค้าตามเลขที่ของ Class

นอกจากนี้ถ้าสร้างสูตร Index แบบ Array จะทำหน้าที่หาดำแหน่งตารางทั้งหมดตามแนวนอน หรือแนวตั้งได้อีกด้วยโดยกำหนดเลขที่ row หรือ column ให้เท่ากับ 0 เช่น

=Index(Class1, 0, 3)

จะได้ค่าทุก row ของ column ที่ 3 คือ ยอด Amount = {10;20;30;40;50} หรือ

=Index(Class1, 3, 0)

จะได้ค่าทุก column ของ row ที่ 3 คือ รหัส a003 = {"a003","c",30}

วิธีใช้สูตร Offset กำหนดขอบเขตตารางแบบยืดหยุ่น (Dynamic Range)

สูตร VLookup และสูตร Index มีลักษณะการใช้งานเหมือนกันประการหนึ่ง คือ ใช้กับพื้นที่ตารางซึ่งกำหนดขอบเขตไว้ตายตัวอยู่แล้ว แม้จะนำสูตร IF หรือ Choose เข้ามาช่วยก็เป็นเพียงช่วยทำให้สูตร VLookup และสูตร Index สามารถเลือกพื้นที่ตารางจากต่างพื้นที่กันมาใช้งาน แต่จะไม่รับรู้ขอบเขตใหม่หากภายหลังมีการเพิ่มลดจำนวนรายการ

ส่วนการใช้คำสั่ง Table เหมาะสำหรับตารางฐานข้อมูลซึ่งมีลักษณะถูกต้องและจะช่วยขยายตำแหน่งอ้างอิงตามจำนวนรายการ แต่จะไม่ลดขนาดเว้นแต่ผู้ใช้จะสั่ง delete row

สูตร Offset เป็นสูตรที่ช่วยกำหนดขอบเขตตารางให้สามารถเพิ่มลดขนาดตามจำนวนรายการ และสามารถนำไปใช้กับตารางทุกประเภท โดยไม่จำกัดว่าจะต้องเป็นตารางฐานข้อมูลเท่านั้น แต่มีข้อจำกัดว่า หากใช้สูตร Offset ดึงข้อมูลข้ามแฟ้มมาใช้ จะต้องเปิดแฟ้มต้นทางขึ้นมาใช้ร่วมกับแฟ้มปลายทางที่ใช้สูตรนี้ด้วยเสมอ

	A	B	C	D
1				
2		Id	Name	Amount
3		a001	a	10
4		a002	b	20
5		a003	c	30
6		a004	d	40
7		a005	e	50
8				
9				

จากภาพนี้เมื่อต้องการขอบเขตตารางตั้งแต่ B3:D7 ให้ใช้สูตร =OFFSET(\$B\$3,0,0,5,3)

- \$B\$3 เป็นตำแหน่งเซลล์อ้างอิงเพื่อกำหนดเซลล์อ้างอิงเริ่มต้นของตาราง
- เลข 0 ตัวแรกในสูตร เป็นตำแหน่งจำนวน row ถัดไป สาเหตุที่ใช้เลข 0 เพื่อชี้ว่าจุดเริ่มต้นของตารางยังคงอยู่ใน row เดียวกันกับเซลล์อ้างอิงเริ่มต้นของตาราง \$B\$3
- เลข 0 ตัวที่สองในสูตร เป็นตำแหน่งจำนวน column ถัดไป สาเหตุที่ใช้เลข 0 เพื่อชี้ว่าจุดเริ่มต้นของตารางยังคงอยู่ใน column เดียวกันกับเซลล์อ้างอิงเริ่มต้นของตาราง \$B\$3
- เลข 5 เป็นส่วนของความสูงของตาราง ในที่นี้แสดงว่ามีความสูง 5 row
- เลข 3 เป็นส่วนของความกว้างของตาราง ในที่นี้แสดงว่ามีความกว้าง 3 column

หากต้องการให้สูตรนี้ปรับขนาดแบบยืดหยุ่น (Dynamic Range) โดยทั่วไปเฉพาะส่วนของความสูงของตารางเท่านั้นที่จะเพิ่มลดตามจำนวนรายการที่บันทึกไว้ เมื่อต้องการทำให้สูตร Offset ปรับขนาดความสูงตาม สามารถหาความสูงได้หลายวิธี ดังนี้

1. สูตร CountA(\$B:\$B)-1 เพื่อนับจำนวนเซลล์ใน column B ที่มีข้อมูลแล้วลบทิ้งเสีย 1 เพราะไม่ต้องการนับรวมหัวตารางเซลล์ B2 ซึ่งมีคำว่า Id
2. ใช้สูตร CountA(\$B\$3:\$B\$10000) เพื่อนับจำนวนเซลล์ที่มีข้อมูลตั้งแต่รายการแรกลงมา อาจเผื่อขนาดไว้ถึง row 10000

การใช้สูตร CountA ให้เลือกใช้กับ column ที่มั่นใจว่ามีข้อมูลบันทึกไว้ครบทุกรายการเสมอ ห้ามมีเซลล์เว้นว่างแทรกอย่างเด็ดขาด ซึ่งโดยทั่วไปในตารางหนึ่งๆต้องมี column หนึ่งซึ่งต้องบันทึกครบทุกรายการอยู่แล้ว โดยเฉพาะ column ของรหัสหรือวันที่

หากเป็นตารางซึ่งมีช่องว่างแทรกไว้ไม่แน่นอน สามารถคำนวณหาความสูงของตารางได้ โดยใช้สูตร Array {=MAX(IF(B:D<>0,ROW(B:D)))-MIN(IF(B:D<>0,ROW(B:D)))}

- สูตร Array { =MAX(IF(B:D<>0,ROW(B:D)))} หาเลขที่ row สุดท้ายของตาราง
- สูตร Array { =MIN(IF(B:D<>0,ROW(B:D)))} หาเลขที่ row แรกของตาราง

หมายเหตุ สูตร Array มีวงเล็บปีกกาปิดหัวท้ายสูตร เกิดจากการกดปุ่ม Ctrl+Shift+Enter พร้อมกันแทนการกดปุ่ม Enter และอาจต้องปรับเพิ่มลดตัวเลขให้เท่ากับความสูงที่แท้จริงหากตารางในช่วง column B:D มีข้อมูลอื่นซึ่งไม่เกี่ยวข้องบันทึกอยู่

เมื่อหาความสูงของตารางด้วยสูตรได้แล้ว จากเดิมที่เป็นสูตร =OFFSET(\$B\$3,0,0,5,3) ให้นำสูตรคำนวณหาความสูงมาซ้อนกันกลายเป็นสูตร

=OFFSET(\$B\$3,0,0, CountA(\$B:\$B)-1, 3)

หรือจะลิงค์เฉพาะผลลัพธ์ของความสูงที่คำนวณได้มาใช้ก็ได้ กลายเป็นสูตร

=OFFSET(\$B\$3,0,0, เซลล์หาความสูง, 3)

สูตร Offset ที่ทำหน้าที่เป็น Dynamic Range นี้ หากสร้างลงไปในเซลล์เดียวจะเกิด error

#VALUE! เสมอ เนื่องจากสูตรนี้หาค่าหลายค่ามาให้จึงไม่สามารถแสดงครบทุกค่าในเซลล์ใด

เซลล์หนึ่งได้ หากต้องการตรวจสอบว่าสูตรทำงานได้ถูกต้องหรือไม่ ให้กดปุ่ม F2 ตามด้วย F9 จะพบว่าสูตรคืนค่าเป็นค่าทั้งหมดในตารางนั้นออกมาให้เห็นแบบ Array

={"a001","a",10;"a002","b",20;"a003","c",30;"a004","d",40;"a005","e",50}

เมื่อเห็นค่าที่แท้จริงของสูตรแล้ว ให้กดปุ่ม Esc เพื่อแก้กลับคืนเป็นสูตรตามเดิม

อีกวิธีหนึ่งในการตรวจสอบสูตร ให้คลิกลงไปในช่วง Formula Bar ลากเลือกสูตรทั้งหมด

=OFFSET(\$B\$3,0,0, CountA(\$B:\$B)-1, 3) แล้วกดปุ่ม F5 แล้วกดปุ่ม Enter เพียงครั้งเดียว

จะพบว่า Excel เปลี่ยนสูตร Offset เป็นตำแหน่งอ้างอิง =B3:D7 พร้อมกับแสดงตำแหน่งในตารางให้เห็น จากนั้นให้กดปุ่ม Esc เพื่อแก้กลับคืนเป็นสูตรตามเดิม

เมื่อพบว่าสูตร Offset แสดงค่าหรือชี้ตำแหน่งได้ถูกต้อง ให้ตรวจสอบตำแหน่งอ้างอิงทุก

ตำแหน่งในสูตรว่าเป็น Absolute Reference เช่น \$B\$3 และ \$B:\$B ก่อน จากนั้นจึงนำสูตรนี้

ไปตั้งชื่อให้เป็น Formula Name โดยใช้คำสั่ง Formulas > Define Name ต่อมาเมื่อต้องการ

สร้างสูตร VLookup, Match, หรือ Index ให้นำชื่อ Formula Name ที่ตั้งขึ้นนี้ไปใช้แบบ

เดียวกับการใช้ชื่อ Range Name โดยทั่วไป

ถ้าต้องการปรับสูตร Offset ให้คืนค่าเป็นพื้นที่ตารางเพียงบางส่วน ให้ปรับส่วนของตำแหน่ง row, column, ความสูง และความกว้างในสูตร เช่น

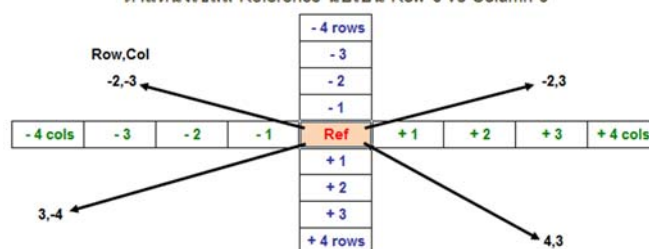
	A	B	C	D
1				
2		Id	Name	Amount
3		a001	a	10
4		a002	b	20
5		a003	c	30
6		a004	d	40
7		a005	e	50
8				
9				

- =OFFSET(\$B\$3,2,0) หมายถึง B5
- =OFFSET(\$B\$3,4,2) หมายถึง D7
- =OFFSET(\$B\$3,2,0,1,3) หมายถึง B5:D5
- =OFFSET(\$B\$3,4,0,1,3) หมายถึง B7:D7
- =OFFSET(\$B\$3,2,0,2,3) หมายถึง B5:D6
- =OFFSET(\$B\$3,0,0, CountA(\$B:\$B)-1, 1) หมายถึง B3:B7
- =OFFSET(\$B\$3,0,1, CountA(\$B:\$B)-1, 1) หมายถึง C3:C7
- =OFFSET(\$B\$3,0,2, CountA(\$B:\$B)-1, 1) หมายถึง D3:D7
- =OFFSET(\$B\$3,0,0, CountA(\$B:\$B)-1, 2) หมายถึง B3:C7
- =OFFSET(\$B\$3,0,1, CountA(\$B:\$B)-1, 2) หมายถึง C3:D7
- =OFFSET(\$B\$3, CountA(\$B:\$B)-2,0,1,3) หมายถึง B7:D7 หรือรายการสุดท้าย

นอกจากนี้ ผู้ใช้สูตร Offset สามารถกำหนดตำแหน่งเซลล์อ้างอิงเซลล์แรกไว้ที่เซลล์ใดก็ได้ (ไม่จำเป็นต้องใช้เซลล์ B3 ซึ่งเป็นเซลล์แรกของรายการแรก) และยังสามารถใช้หาค่าในทุกทิศทาง เพียงกำหนดจำนวน row ถัดไปหรือเลข column ถัดไปให้เป็นค่าลบ

= OFFSET (Ref , จำนวน Row ถัดไป , จำนวน Column ถัดไป)

ตำแหน่งเซลล์ Reference นับเป็น Row 0 vs Column 0



วิธีใช้ Text กำหนดขอบเขตตารางแบบยืดหยุ่น (Dynamic Range)

แมสสูตร Offset จะช่วยกำหนดขอบเขตตารางให้อัตโนมัติก็ตาม แต่ขอบเขตที่ได้ขึ้นอยู่กับปริมาณข้อมูลหรือจำนวนรายการที่บันทึกไว้ ซึ่งในบางกรณีอาจไม่ตรงกับความต้องการ บางครั้งผู้ใช้งานอาจต้องการกำหนดขอบเขตตารางให้แปรเปลี่ยนไปตามเงื่อนไขพิเศษ

	I6 =SUM(INDIRECT(I4))									
	A	B	C	D	E	F	G	H	I	J
1									Reference	
2		First						FirstCell	\$B\$2	I2 : '=CELL("Address",First)
3					2			LastCell	\$F\$15	I3 : '=CELL("Address",Last)
4		2						Range	\$B\$2:\$F\$15	I4 : '=FirstCell&"."&LastCell
5										
6					2			Total	16	I6 : =SUM(INDIRECT(I4))
7			2							
8										
9								First	=B\$2	
10								FirstCell	=I\$2	
11				2				Last	=F\$15	
12		2				2		LastCell	=I\$3	
13										
14				2						
15										
16										
17										

จากภาพนี้สมมุติว่าต้องการหาผลรวมของตัวเลขในพื้นที่ตารางจาก B2:F15 ซึ่งตามปกติสามารถใช้สูตร =SUM(B2:F15) ได้อยู่แล้ว เมื่อต้องการเพิ่มลดขนาดตาราง B2:F15 ให้ต่างไปจากเดิมก็ต้องแก้ไขตำแหน่งอ้างอิงในสูตร หรือสั่ง Insert/Delete Row หรือ Column ซึ่งยอมทำให้ตำแหน่งอ้างอิงในสูตร Sum ปรับตาม แต่ผลจากการ Insert/Delete Row หรือ Column อาจกระทบกับพื้นที่ตารางใกล้เคียงที่อยู่ในแนวเดียวกันและทำให้ต้อง Copy สูตรที่สร้างไว้เดิมมีลำดับเซลล์ที่อ้างอิงไว้ผิดเพี้ยนไปจากที่ควรก็ได้

เริ่มต้นจากใช้สูตรในเซลล์ I2 และ I3 =Cell("Address",cell) หาตำแหน่งอ้างอิงของเซลล์หัวมุมซ้ายบนสุด(First)และเซลล์หัวมุมขวาล่างสุด(Last) จะได้ตำแหน่งอ้างอิง \$B\$2 และ \$F\$15 ตามลำดับ

นำตำแหน่งอ้างอิงที่ได้มาเชื่อมต่อกันในเซลล์ I4 ด้วยสูตร =FirstCell&"."&LastCell ทำให้ได้ตำแหน่งอ้างอิง \$B\$2:\$F\$15 ซึ่งมีลักษณะเป็น **Text ที่บอกตำแหน่ง** พร้อมทั้งจะนำไปใช้กับสูตร Indirect

เมื่อต้องการหาผลรวม ให้ใช้สูตรในเซลล์ I6 =SUM(INDIRECT(I4)) หรือจะนำสูตร =INDIRECT(\$I\$4) ไปตั้งชื่อเป็น Formula Name เพื่อเพิ่มความสะดวกในการนำไปใช้ร่วมกับสูตรอื่นใดก็ได้ โดยไม่จำกัดเฉพาะสูตร Sum

หากต้องการปรับเปลี่ยนขนาดตาราง ให้ย้ายเซลล์ First หรือ Last ไปวางลงในตำแหน่งที่ต้องการ

	A	B	C	D	E	F	G	H	I	J
1									Reference	
2					2			FirstCell	\$C\$5	I2 : '=CELL("Address",First)
3								LastCell	\$E\$9	I3 : '=CELL("Address",Last)
4		2	First					Range	\$C\$5:\$E\$9	I4 : '=FirstCell&"."&LastCell
5								Total	4	I6 : '=SUM(INDIRECT(I4))
6			2							
7										
8										
9					Last			First	=B\$2	
10								FirstCell	=I\$2	
11				2				Last	=F\$15	
12		2				2		LastCell	=I\$3	
13										
14				2						
15										
16										

จากภาพนี้เมื่อย้ายเซลล์ First กับ Last ไปวางไว้ที่เซลล์ C5 และ E9 ตามลำดับ จะทำให้สูตร Sum หาผลรวมในขอบเขตตาราง C5:E9 ตามไปด้วย

ประเด็นสำคัญมิได้ขึ้นกับการสั่งย้ายเซลล์ไปตำแหน่งอื่น เพราะการสั่งย้ายเซลล์ย่อมต้องเกิดจากฝีมือของมนุษย์ แต่ขึ้นกับการใช้สูตร =Cell("Address",cell) ต่างหาก เพราะสูตรนี้จะคืนค่าเป็นตำแหน่งของเซลล์ที่กำหนดไว้ภายในวงเล็บของสูตร เมื่อใช้สูตร IF, Choose, Index, หรือ Offset ซ้อนเข้าไปก็จะได้ตำแหน่งเซลล์แปรเปลี่ยนตามให้ทันที เช่น

=CELL("Address",INDEX(MyNum,FirstRow))

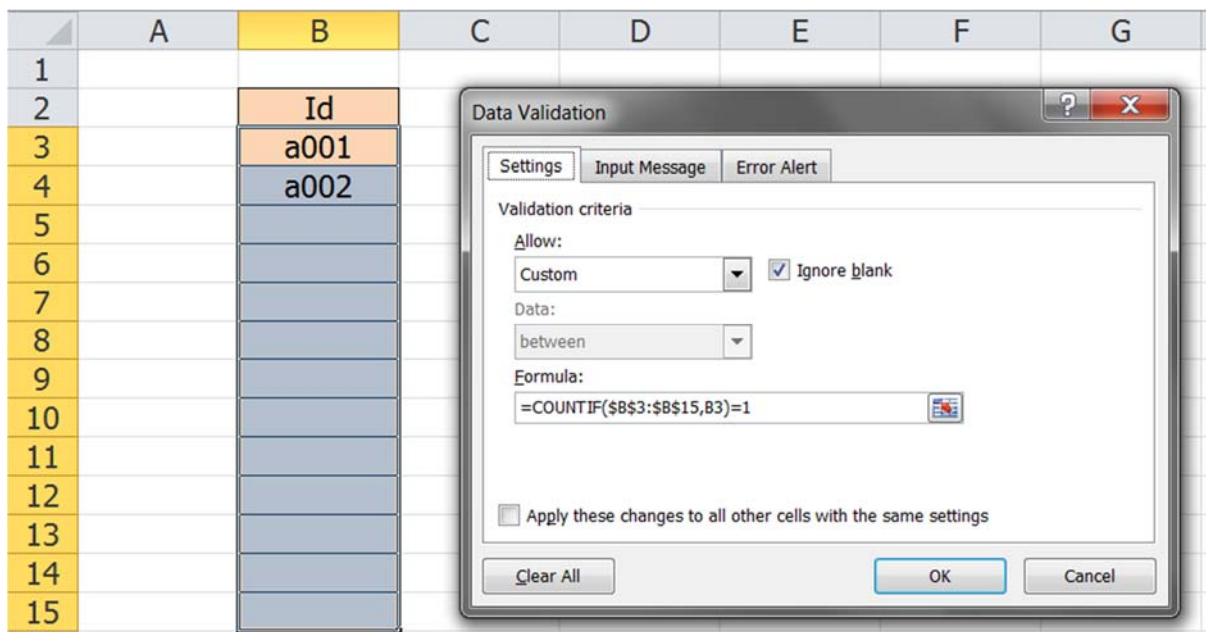
	FirstCell			=CELL("Address",INDEX(MyNum,FirstRow))	
	A	B	C	D	E
1		MyNum			
2		11	FirstRow	2	
3		22	LastRow	4	
4		33		Reference	
5		44	FirstCell	\$B\$3	D5 : '=CELL("Address",INDEX(MyNum,FirstRow))
6		55	LastCell	\$B\$5	D6 : '=CELL("Address",INDEX(MyNum,LastRow))
7		66	Range	\$B\$3:\$B\$5	D7 : '=FirstCell&"."&LastCell
8		77			
9		88	Total	99	D9 : =SUM(INDIRECT(D7))
10		99			
11		110			
12		121			
13		132			
14		143			
15		154			

เมื่อซ้อนสูตร INDEX(MyNum,FirstRow) เข้าไปในสูตร Cell จะเปลี่ยนหน้าที่จากการหาค่าที่ 2 คือเลข 22 จากพื้นที่ตารางชื่อ MyNum มาเป็นการบอกตำแหน่งเซลล์ B3

การจัดการค่าซ้ำ

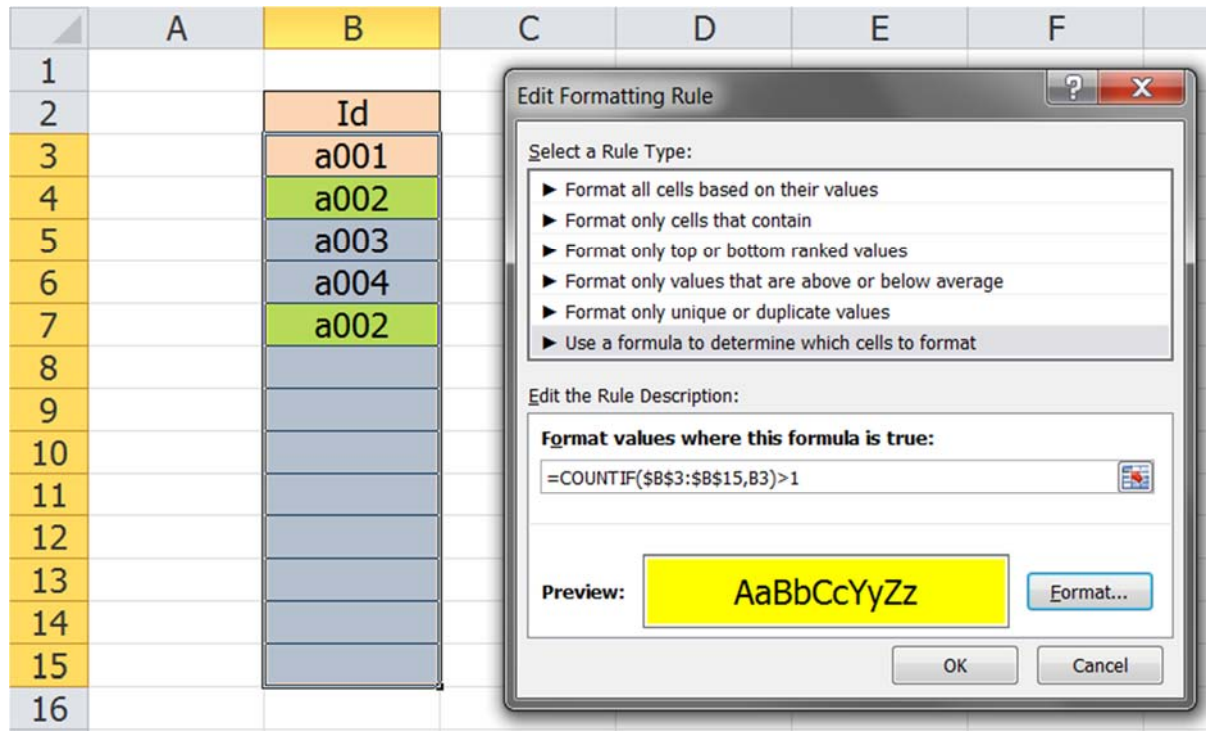
ในหลักการบันทึกข้อมูลที่ดีต้องควบคุมการบันทึกค่าซ้ำให้เกิดขึ้นได้เท่าที่จำเป็นเท่านั้น เช่นในงานบันทึกข้อมูลสินค้าที่ขาย จะมีตารางบันทึกรายละเอียดของสินค้าเอาไว้ จากนั้นเมื่อต้องการรายละเอียดของสินค้าไปใช้ที่เซลล์อื่น ไม่จำเป็นต้องพิมพ์รายละเอียดของสินค้านั้นๆซ้ำอีก ค่าที่จำเป็นต้องพิมพ์ซ้ำมีเพียงรหัส พอพิมพ์รหัสลงไปแล้วให้ใช้สูตรดึงข้อมูลของรหัสสินค้านั้นมาแสดง โดยใช้สูตร VLookup, Match, Index, หรือสูตรอื่นๆดึงรายละเอียดของรหัสมาแสดง

อย่างไรก็ตามมีบางงานที่ผู้ใช้ Excel อาจเผลอพิมพ์ค่าซ้ำ โดยเฉพาะงานที่ต้องกระจายให้พนักงานหลายคนช่วยกันบันทึกข้อมูล เครื่องมือที่ใช้ช่วยเตือนทันทีที่มีการบันทึกค่าซ้ำได้แก่คำสั่ง Data Validation เพื่อป้องกันหรือเตือนการบันทึกซ้ำและ Conditional Formatting เพื่อเปลี่ยนสีเซลล์ที่มีค่าซ้ำ โดยใช้งานร่วมกับสูตร CountIF



หากต้องการสั่งให้ Excel เตือนเมื่อมีการพิมพ์ค่าซ้ำ ให้เริ่มจากเลือกพื้นที่ตารางตั้งแต่ B3:B15 ซึ่งเตรียมไว้สำหรับบันทึกข้อมูล จากนั้นสั่ง Data > Data Validation > Data Validation เลือก Allow แบบ Custom แล้วพิมพ์สูตร =COUNTIF(\$B\$3:\$B\$15,B3)=1 ลงไปในช่อง Formula โดยไม่ต้องใส่เครื่องหมาย \$ ในส่วนของ B3 ที่อยู่ด้านท้ายในวงเล็บของสูตร เพื่อให้ Excel ใช้ตำแหน่งถัดไปในการตรวจสอบให้เอง

ถ้าต้องการเปลี่ยนสีเซลล์ที่มีการบันทึกค่าซ้ำ ให้เริ่มจากเลือกพื้นที่ตารางตั้งแต่ B3:B15 แล้วสั่ง Home > Conditional Formatting > New Rule โดยเลือกใช้เงื่อนไขแบบ Use a formula to determine which cells to format แล้วใช้สูตร =COUNTIF(\$B\$3:\$B\$15,B3)>1 ซึ่งถ้าสูตรนี้คืนค่าเป็น True ย่อมแสดงว่าสูตร CountIF นับข้อมูลแล้วพบว่ามีค่าซ้ำจริง



โปรดสังเกตว่าถ้าใช้ Data Validation จะใช้สูตร =COUNTIF(\$B\$3:\$B\$15,B3)=1 แต่เมื่อใช้กับ Conditional Formatting ต้องเปลี่ยนสมการจากเท่ากับ 1 เป็นมากกว่า 1 แทน ด้วยสูตร =COUNTIF(\$B\$3:\$B\$15,B3)>1 เพราะ Data Validation จะยอมให้บันทึกค่าต่อเมื่อนับแล้วมีเพียงค่าเดียวคือนับแล้วเท่ากับ 1 ค่าเท่านั้น ส่วนการใช้ Conditional Formatting จะเปลี่ยนสีเพื่อเตือนเมื่อมีค่าซ้ำคือเมื่อนับแล้วที่ค่ามากกว่า 1

Data Validation เป็นเครื่องมือที่ควรนำมาใช้ร่วมกับ Conditional Formatting เสมอ เนื่องจาก Data Validation จะช่วยตรวจสอบค่าซ้ำให้ต่อเมื่อเกิดค่าซ้ำขึ้นจากการพิมพ์ลงไปในตารางเท่านั้น แต่จะไม่ตรวจสอบหากมีการ copy ค่าซ้ำวางลงไปแต่อย่างใด ดังนั้นเมื่อเกิดค่าซ้ำขึ้นเมื่อใดก็จะเห็นสีในเซลล์เปลี่ยนแปลงต่างไปจากเดิมจากคำสั่ง Conditional Formatting เป็นปรากฏการณ์สุดท้ายของการตรวจสอบอีกชั้นหนึ่ง

ในภาพต่อไปหากต้องการเปลี่ยนสีเซลล์เพื่อช่วยแสดงตำแหน่งข้อมูลรายการที่ต้องการให้เห็นชัดเจนขึ้น

	A	B	C	D	E	F	G
1							
2		Num	ID	Name	Amount	ID	
3		1	a001	a	10	a001	
4		2	a002	b	20		
5		3	a003				
6		4	a004				
7		5	a005				
8		6	a001				
9		7	a007				
10		8	a002				
11		9	a001				
12		10	a010				
13		11	a001				
14		12	a002				
15		13	a003				
16		14	a004				
17		15	a001				
18		16	a006				
19		17	a007				
20							
21							

ตารางด้านซ้ายมือเป็นตารางฐานข้อมูล ให้เลือกพื้นที่ตารางเฉพาะส่วนของรายการทั้งหมด ตั้งแต่ B3:E20 แล้วสั่ง Home > Conditional Formatting > New Rule โดยเลือกใช้เงื่อนไขแบบ Use a formula to determine which cells to format แล้วใช้สูตร $=\$C3=\$G\$3$

The screenshot shows an Excel spreadsheet with a table of data. The table has columns: Num, ID, Name, Amount, and ID. The data is as follows:

Num	ID	Name	Amount	ID
1	a001	a	10	a002
2	a002	b	20	
3	a003	c	30	
4	a004	d	40	
5	a005	e	50	
6	a001	f	60	
7	a007	g	70	
8	a002	h	80	
9	a001	i	90	
10	a010	j	100	
11	a001	a	110	
12	a002	b	120	
13	a003	c	130	
14	a004	a	140	
15	a001	b	150	
16	a006	c	160	
17	a007	a	170	

The 'Edit Formatting Rule' dialog box is open, showing the formula $=\$C3=\$G\$3$ and a preview of the formatting. The preview shows a cell with the text 'AaBbCcYyZz' in a yellow background.

เซลล์ G3 เป็นเซลล์ที่ใช้กรอกรหัสที่ต้องการใช้สืบทอดตำแหน่ง ดังนั้นในสูตร $=\$C3=\$G\$3$ ที่ใช้นี้จึงกำหนดตำแหน่งแบบคงที่ $\$G\3 เอาไว้ ส่วนเซลล์ C3 เป็นเซลล์แรกของรายการรหัส ID ซึ่งในสูตรนี้กำหนดให้เลื่อนตำแหน่งไปเรื่อยๆใน column C จึงต้องใช้ $\$C3$

การหาเลขตำแหน่งรายการข้อมูลที่ซ้ำด้วยสูตร Multiple Match

สูตร VLookup และสูตร Match มีจุดอ่อนตรงที่สูตรเหล่านี้จะทำงานกับข้อมูลที่ไม่ซ้ำเท่านั้น หากนำไปใช้กับตารางที่มีข้อมูลซ้ำจะให้คำตอบของข้อมูลรายการแรกที่ซ้ำกัน ส่วนคำสั่ง Data > Advanced หรือ Filter จะทำงานต่อเมื่อถูกสั่งใหม่เป็นครั้งๆไป ดังนั้นหากต้องการแสดงรายละเอียดของข้อมูลที่ซ้ำกันในทันทีที่มีการเปลี่ยนแปลง ต้องหันมารู้จักกับสูตรที่เรียกกันว่า สูตร Multiple Match

สูตร Multiple Match ถือเป็นสูตรที่ถูกถามกันบ่อยครั้งมาก ถึงแม้ได้เคยเขียนอธิบายไว้อย่างละเอียดหลายครั้งหลายโอกาสแล้วก็ตาม มักไม่เข้าใจว่าสูตรนี้สร้างกันได้อย่างไรหากไม่ได้เห็นขั้นตอนการสร้างทีละขั้นกับตาตัวเอง

L3 {=SMALL(IF(ID=G3,Num),Num)}																	
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O		
1																	
2		Num	ID	Name	Amount		ID		Step1	Step2	Step3	Final		Name	Amount		
3		1	a001	a	10		a001		TRUE	1	1	1		a	10		
4		2	a002	b	20				FALSE	FALSE	6	6		f	60		
5		3	a003	c	30				FALSE	FALSE	9	9		i	90		
6		4	a004	d	40				FALSE	FALSE	11	11		a	110		
7		5	a005	e	50				FALSE	FALSE	15	15		b	150		
8		6	a001	f	60				TRUE	6	#NUM!	#NUM!		no more	#NUM!		
9		7	a007	g	70				FALSE	FALSE	#NUM!	#NUM!		no more	#NUM!		
10		8	a002	h	80				FALSE	FALSE	#NUM!	#NUM!		no more	#NUM!		
11		9	a001	i	90				TRUE	9	#NUM!	#NUM!		no more	#NUM!		
12		10	a010	j	100				FALSE	FALSE	#NUM!	#NUM!		no more	#NUM!		
13		11	a001	a	110				TRUE	11	#NUM!	#NUM!					
14		12	a002	b	120				FALSE	FALSE	#NUM!	#NUM!					
15		13	a003	c	130				FALSE	FALSE	#NUM!	#NUM!					
16		14	a004	a	140				FALSE	FALSE	#NUM!	#NUM!					
17		15	a001	b	150				TRUE	15	#NUM!	#NUM!					
18		16	a006	c	160				FALSE	FALSE	#NUM!	#NUM!					
19		17	a007	a	170				FALSE	FALSE	#NUM!	#NUM!					
20									FALSE	FALSE	#NUM!	#NUM!					
21																	
22									สูตรในเซลล์ \$I\$3 : {=ID=G3}								
23									สูตรในเซลล์ \$J\$3 : {=IF(I3:I20,Num)}								
24									สูตรในเซลล์ \$K\$3 : {=SMALL(J3:J20,Num)}								
25									สูตรในเซลล์ \$L\$3 : {=SMALL(IF(ID=G3,Num),Num)}								
26									สูตรในเซลล์ \$N\$3 : '=IF(ISERROR(L3), "no more", INDEX(Name,L3))								
27									สูตรในเซลล์ \$O\$3 : =INDEX(Amount,L3)								

ตารางด้านซ้ายมือของภาพนี้เป็นตารางฐานข้อมูลซึ่งมีค่าซ้ำกันบันทึกไว้ เช่น ข้อมูลของรหัส a001 ถูกบันทึกไว้ในรายการที่ 1, 6, 9, 11, และ 15 โดยตัวเลขตำแหน่งรายการของรหัส a001 เหล่านี้สามารถหาได้ด้วยสูตร Multiple Match {=SMALL(IF(ID=G3,Num),Num)} ซึ่งสร้างไว้ในเซลล์ L3:L12 แบบ Array

ตัวอย่างนี้ใช้การตั้งชื่อ Range Name ว่า Num, ID, Name, และ Amount ให้กับพื้นที่ข้อมูล B3:B20, C3:C20, D3:D20, และ E3:E20 ตามลำดับ ส่วนเซลล์ G3 เป็นเซลล์รับค่ารหัสที่ต้องการค้นหาตำแหน่งรายการ

L3 {=SMALL(IF(ID=G3,Num),Num)}															
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1															
2		Num	ID	Name	Amount		ID		Step1	Step2	Step3	Final		Name	Amount
3		1	a001	a	10		a001		TRUE	1	1	1		a	10
4		2	a002	b	20				FALSE	FALSE	6	6		f	60
5		3	a003	c	30				FALSE	FALSE	9	9		i	90
6		4	a004	d	40				FALSE	FALSE	11	11		a	110
7		5	a005	e	50				FALSE	FALSE	15	15		b	150
8		6	a001	f	60				TRUE	6	#NUM!	#NUM!		no more	#NUM!
9		7	a007	g	70				FALSE	FALSE	#NUM!	#NUM!		no more	#NUM!
10		8	a002	h	80				FALSE	FALSE	#NUM!	#NUM!		no more	#NUM!
11		9	a001	i	90				TRUE	9	#NUM!	#NUM!		no more	#NUM!
12		10	a010	j	100				FALSE	FALSE	#NUM!	#NUM!		no more	#NUM!
13		11	a001	a	110				TRUE	11	#NUM!	#NUM!			
14		12	a002	b	120				FALSE	FALSE	#NUM!	#NUM!			
15		13	a003	c	130				FALSE	FALSE	#NUM!	#NUM!			
16		14	a004	a	140				FALSE	FALSE	#NUM!	#NUM!			
17		15	a001	b	150				TRUE	15	#NUM!	#NUM!			
18		16	a006	c	160				FALSE	FALSE	#NUM!	#NUM!			
19		17	a007	a	170				FALSE	FALSE	#NUM!	#NUM!			
20									FALSE	FALSE	#NUM!	#NUM!			
21															
22															
23															
24															
25															
26															
27															

สูตรในเซลล์ \$I\$3 : {=ID=G3}
 สูตรในเซลล์ \$J\$3 : {=IF(I3:I20,Num)}
 สูตรในเซลล์ \$K\$3 : {=SMALL(J3:J20,Num)}
 สูตรในเซลล์ \$L\$3 : {=SMALL(IF(ID=G3,Num),Num)}
 สูตรในเซลล์ \$N\$3 : {=IF(ISERROR(L3), "no more", INDEX(Name,L3))}
 สูตรในเซลล์ \$O\$3 : =INDEX(Amount,L3)

เพื่อแสดงให้เห็นที่ไปที่มาของสูตร {=SMALL(IF(ID=G3,Num),Num)} ว่าเกิดจากการคำนวณหลายขั้นตอนและแต่ละขั้นทำหน้าที่หาค่าอะไรบ้าง ขออธิบายสูตรที่สร้างไว้ใน Step1 - Step3 ดังนี้ (เวลาใช้งานจริงไม่จำเป็นต้องสร้างสูตรเหล่านี้แต่อย่างใด)

Step1 : เริ่มจากเลือกเซลล์ I3:I20 พร้อมกันแล้วสร้างสูตร =ID=G3 แล้วกดปุ่ม Ctrl+Shift+Enter จะมีวงเล็บปีกกาปิดหัวท้ายสูตรกลายเป็น {=ID=G3} คั่นค่าออกมาเป็น True คู่ขนานตรงกับตำแหน่งรายการที่มีรหัส a001

Step2 : ปรับตำแหน่งที่เป็น True จาก Step1 ให้เป็นเลขที่รายการ Num โดยเลือกเซลล์ J3:J20 พร้อมกันแล้วสร้างสูตร =IF(I3:I20,Num) แล้วกดปุ่ม Ctrl+Shift+Enter จะมีวงเล็บปีกกาปิดหัวท้ายสูตรกลายเป็น {=IF(I3:I20,Num)} จะพบว่าตำแหน่งที่เคยเป็นค่า True กลายเป็นเลขตำแหน่งรายการ 1, 6, 9, 11, 15 ขึ้นมาให้เห็นแทน

Step3 : นำตัวเลขตำแหน่งรายการจาก Step2 มาเรียงจากน้อยไปมากโดยนำไปซ่อนในสูตร Small โดยเลือกเซลล์ K3:K20 พร้อมกันแล้วสร้างสูตร =SMALL(J3:J20,Num) แล้วกดปุ่ม Ctrl+Shift+Enter จะมีวงเล็บปีกกาปิดหัวท้ายสูตรกลายเป็น {=SMALL(J3:J20,Num)}

เมื่อนำสูตรจาก Step1 – Step3 มาซ้อนกันจะกลายเป็นสูตร Multiple Match ในเซลล์ L3:L12 {=SMALL(IF(ID=G3,Num),Num)} ทั้งนี้โปรดสังเกตว่าไม่จำเป็นต้องสร้างสูตรลงไปในเซลล์ L3:L20 ที่ยาวลงมาถึง row 20 เช่นที่ใช้กับสูตรใน Step1 – Step3 แต่อย่างใด แต่ให้เลือกจำนวน row จากเซลล์ L3:L12 หรือ 10 row เท่าที่คิดว่ารหัสจะมีโอกาสซ้ำกันสูงสุดก็ครั้งก็พอ จากนั้นให้ใช้สูตร Index ดึงข้อมูลตามลำดับรายการที่หาได้มาแสดงต่อไป

การหาเลขตำแหน่งรายการข้อมูลที่เป็น Unique Item

Unique Item คือ ข้อมูลที่ไม่ซ้ำกับข้อมูลอื่น ซึ่งในกรณีที่มีการบันทึกรายการซ้ำกันไว้หลายรายการ รายการแรกของแต่ละชุดที่ซ้ำกันก็คือ Unique Item โดยใช้สูตรคล้ายกับสูตร Multiple Match เพียงมีขั้นตอนเพิ่มขึ้นอีกหนึ่งขั้นตอน

I3 {=SMALL(IF(MATCH(ID&"",ID&"",0)=Num,Num),Num)}										
A	B	C	D	E	F	G	H	I	J	K
1										
2	Num	ID		Step1	Step2	Step3	Step4	Final		Unique ID
3	1	a001		1	TRUE	1	1	1		a001
4	2	a002		2	TRUE	2	2	2		a002
5	3	a003		3	TRUE	3	3	3		a003
6	4	a004		4	TRUE	4	4	4		a004
7	5	a005		5	TRUE	5	5	5		a005
8	6	a001		1	FALSE	FALSE	7	7		a007
9	7	a007		7	TRUE	7	10	10		a010
10	8	a002		2	FALSE	FALSE	16	16		a006
11	9	a001		1	FALSE	FALSE	#NUM!	#NUM!		#NUM!
12	10	a010		10	TRUE	10	#NUM!	#NUM!		#NUM!
13	11	a001		1	FALSE	FALSE	#NUM!			
14	12	a002		2	FALSE	FALSE	#NUM!			
15	13	a003		3	FALSE	FALSE	#NUM!			
16	14	a004		4	FALSE	FALSE	#NUM!			
17	15	a001		1	FALSE	FALSE	#NUM!			
18	16	a006		16	TRUE	16	#NUM!			
19	17	a007		7	FALSE	FALSE	#NUM!			
20				18	FALSE	FALSE	#NUM!			
21										
22				สูตรในเซลล์ \$E\$3 : {=MATCH(ID&"",ID&"",0)}						
23				สูตรในเซลล์ \$F\$3 : {=E3:E20=Num}						
24				สูตรในเซลล์ \$G\$3 : {=IF(F3:F20,Num)}						
25				สูตรในเซลล์ \$H\$3 : {=SMALL(G3:G20,Num)}						
26				สูตรในเซลล์ \$I\$3 : {=SMALL(IF(MATCH(ID&"",ID&"",0)=Num,Num),Num)}						
27				สูตรในเซลล์ \$K\$3 : '=INDEX(ID,I3)						

ในภาพนี้ตารางรหัส ID จาก C3:C20 มีรหัสบันทึกไว้ซ้ำกัน รหัส ID ที่เป็น Unique Item คือ รหัสรายการที่ 1, 2, 3, 4, 5, 7, 10, และ 16 ซึ่งหาตัวเลขตำแหน่งรายการเหล่านี้ได้จากสูตร Array ในเซลล์ I3:I12 { =SMALL(IF(MATCH(ID&"",ID&"",0) = Num, Num), Num)} โดยสูตรนี้ต้องสร้างพร้อมกันลงทีเดียวไปในเซลล์ I3:I12 แล้วกดปุ่ม Ctrl+Shift+Enter เพื่อให้มีเครื่องหมายวงเล็บปีกกาปิดหัวท้ายสูตรให้เอง (ทั้งนี้ให้เลือกจำนวนเซลล์ I3:I12 มากหรือน้อยกว่านี้ก็ได้แล้วแต่ว่าจะมีจำนวน Unique Item ทั้งหมดมากน้อยเท่าใด)

Step1 : สูตร MATCH(ID&"",ID&"",0) ทำหน้าที่หาเลขตำแหน่งรายการของ ID แต่ละตัวสาเหตุที่ต้องปรับให้ใช้ ID&"" เพื่อให้สูตรนี้ยังใช้งานได้หากมีเซลล์ว่าง เช่น เซลล์ C20

Step2 : นำค่าจาก Step1 ไปเทียบกับเลขลำดับ Num จะพบว่าตำแหน่งรายการที่ตรงกัน หรือแสดงค่าเท่ากับ True คือตำแหน่งที่เป็น Unique Item

Step3 : เปลี่ยนค่า True จาก Step2 ให้เป็นเลขที่รายการด้วยสูตร

$\text{IF}(\text{MATCH}(\text{ID}\&"",\text{ID}\&"",0) = \text{Num}, \text{Num})$

Step4 : เรียงเลขตำแหน่งรายการที่ได้จาก Step3 ให้เรียงจากน้อยไปมากด้วยสูตร Small

เมื่อได้ตัวเลขตำแหน่งรายการที่เป็น Unique แล้วให้ใช้สูตร Index แสดงชื่อรหัส ID ต่อไป

นอกจากสูตรข้างต้นแล้ว ยังมีสูตรค้นหาชื่อรายการที่เป็น Unique หรือจะใช้คำสั่ง Data > Advanced แบบ Unique records only

D4		{=INDEX(ID,MATCH(0,COUNTIF(\$D\$3:D3,ID),0))}		
	A	B	C	D
1				
2		ID	UniqueID	
3		a001	a001	สูตรในเซลล์ \$D\$3 : '=B3
4		a002	a002	สูตรในเซลล์ \$D\$4 : {=INDEX(ID,MATCH(0,COUNTIF(\$D\$3:D3,ID),0))}
5		a003	a003	สูตรในเซลล์ \$D\$5 : {=INDEX(ID,MATCH(0,COUNTIF(\$D\$3:D4,ID),0))}
6		a004	a004	สูตรในเซลล์ \$D\$6 : {=INDEX(ID,MATCH(0,COUNTIF(\$D\$3:D5,ID),0))}
7		a005	a005	สูตรในเซลล์ \$D\$7 : {=INDEX(ID,MATCH(0,COUNTIF(\$D\$3:D6,ID),0))}
8		a001	a007	สูตรในเซลล์ \$D\$8 : {=INDEX(ID,MATCH(0,COUNTIF(\$D\$3:D7,ID),0))}
9		a007	a010	สูตรในเซลล์ \$D\$9 : {=INDEX(ID,MATCH(0,COUNTIF(\$D\$3:D8,ID),0))}
10		a002	a006	สูตรในเซลล์ \$D\$10 : {=INDEX(ID,MATCH(0,COUNTIF(\$D\$3:D9,ID),0))}
11		a001	0	สูตรในเซลล์ \$D\$11 : {=INDEX(ID,MATCH(0,COUNTIF(\$D\$3:D10,ID),0))}
12		a010	#N/A	สูตรในเซลล์ \$D\$12 : {=INDEX(ID,MATCH(0,COUNTIF(\$D\$3:D11,ID),0))}
13		a001	#N/A	สูตรในเซลล์ \$D\$13 : {=INDEX(ID,MATCH(0,COUNTIF(\$D\$3:D12,ID),0))}
14		a002	#N/A	สูตรในเซลล์ \$D\$14 : {=INDEX(ID,MATCH(0,COUNTIF(\$D\$3:D13,ID),0))}
15		a003	#N/A	สูตรในเซลล์ \$D\$15 : {=INDEX(ID,MATCH(0,COUNTIF(\$D\$3:D14,ID),0))}
16		a004	#N/A	สูตรในเซลล์ \$D\$16 : {=INDEX(ID,MATCH(0,COUNTIF(\$D\$3:D15,ID),0))}
17		a001		
18		a006	ที่มา : http://www.get-digital-help.com/	
19		a007		
20				
21				

เซลล์ D3 สร้างสูตร =B3 เพื่อนำรหัสรายการแรกยกมาเป็น Unique Item ไว้ก่อน จากนั้นใน

เซลล์ D4 ให้สร้างสูตร Array {=INDEX(ID,MATCH(0,COUNTIF(\$D\$3:D3,ID),0))} แล้ว

copy สูตรไปต่อลงไปเซลล์ถัดไปด้านล่าง โปรดดูรายละเอียดของสูตรนี้ได้จาก

<http://www.get-digital-help.com/2009/03/30/how-to-extract-a-unique-list-and-the-duplicates-in-excel-from-one-column/>

การคำนวณยอดรวมเพื่อจัดทำรายงาน

การจัดเก็บข้อมูลไว้ในโครงสร้างฐานข้อมูลที่ถูกต้อง เป็นองค์ประกอบสำคัญช่วยให้เกิดความสะดวกและง่ายในการหาคำตอบที่ต้องการ แต่ผู้ใช้ Excel แบบมักง่ายอาจหาทางที่ง่ายกว่านั้นโดยปรับแต่งโครงสร้างฐานข้อมูลให้กลายเป็นหน้ารายงาน ทำให้ไม่สามารถใช้ฐานข้อมูลได้เต็มที่อีกต่อไป ยิ่งกว่านั้นหากรายงานที่ต้องจัดพิมพ์มีร้อยหน้า ก็มักออกแบบตารางรายงานเลียนแบบหน้ารายงานทั้งร้อยหน้า ทำให้แฟ้มมีขนาดใหญ่ขึ้นอย่างมากเนื่องจากต้องเตรียมพื้นที่ในชีทเดียวกัน ต่างชีท หรือต่างแฟ้ม รอไว้สำหรับแสดงข้อมูลทุกหน้า

พึงระลึกไว้เสมอว่าฐานข้อมูลที่ดีต้องคงสภาพโครงสร้างเช่นเดิมไว้ตลอด การเปลี่ยนแปลงที่เกิดขึ้นกับตารางฐานข้อมูลจะมีเพียงการเพิ่มรายการใหม่ต่อท้ายรายการเดิมลงไปเรื่อยๆ หากไม่จำเป็นจะไม่มีการ insert row แทรก หรือ delete row ลบรายการที่บันทึกไว้ทิ้ง และหน้ารายงานต้องหาทางสร้างขึ้นเพียงหน้าเดียวหรือสูตรเดียว แล้วหาทางเปลี่ยนหน้าเดียวหรือสูตรเดียวนั้นให้แสดงคำตอบหน้าอื่นๆหรือเรื่องอื่นๆได้โดยไม่ต้องสร้างหน้าใหม่หรือสูตรใหม่ขึ้นอีก

หลังจากฐานข้อมูลมีโครงสร้างที่แน่นอนชัดเจนแล้ว เมื่อต้องการหาคำตอบเกี่ยวข้องกับตัวเลขยอดรวม มีทางเลือกให้ใช้คำสั่งหรือสูตรได้ ดังนี้

- ใช้สูตร SubTotal ร่วมกับคำสั่ง Filter และ Custom Views
- ใช้คำสั่ง Data > SubTotal เพื่อสร้างยอดรวมรายการแต่ละประเภทขึ้นภายในตารางฐานข้อมูลนั้น
- ใช้คำสั่ง Data > Consolidate เพื่อรวมตัวเลขของหลายตารางตามตำแหน่งเรื่องเดียวกันเข้าด้วยกัน
- ใช้สูตร Array SumIF หรือ DSum เพื่อหายอดรวมแยกประเภท แล้วใช้คำสั่ง Data > What-if Analysis > Data Table เพื่อหายอดรายการอื่นๆต่อ
- ใช้คำสั่ง Insert > Pivot Table เพื่อสร้างตารางเปรียบเทียบตัวเลขคำตอบ

ทางออกที่ดีกว่าการใช้สูตรบวกหรือสูตร Sum

ก่อนจะอธิบายสูตร SubTotal หลายคนอาจจะสงสัยว่าทำไมจึงไม่อธิบายวิธีใช้สูตร Sum ไว้ก่อน ทั้งนี้ไม่ใช่เพราะสูตร Sum เป็นสูตรง่ายที่ทุกคนควรรู้จักกันดีอยู่แล้วหรอก แต่เนื่องจาก Sum เป็นสูตรที่เหมาะสมสำหรับใช้หา

รวมจากตารางซึ่งมีโครงสร้างแน่นอน

ถาวร และในขนาดต้องไม่มีการ

เปลี่ยนแปลงโครงสร้างอีกต่อไป

ในภาพนี้ เซลล์ B12 หาผลรวมของ

เซลล์ B2:B11 ด้วยสูตร =SUM(B2:B11)

ได้คำตอบยอดรวมเท่ากับ 55

	B12				
	A	B	C	D	E
1					
2		1			
3		2			
4		3			
5		4			
6		5			
7		6			
8		7			
9		8			
10		9			
11		10			
12		55			
13					

หากต่อมา มีการปรับโครงสร้างตาราง โดย

กำหนดให้หาผลรวมของเซลล์ B2:B6 เพิ่ม และให้แทรก row ใหม่ระหว่าง row 6 และ row 7

แล้วสร้างสูตร =SUM(B2:B6) ลงไปในเซลล์ B7 ได้ยอดรวมเท่ากับ 15 แต่จะส่งผลให้สูตร

Sum เดิมที่หาผลรวมทั้งหมดเท่ากับ 55 กลายเป็นยอดรวมใหม่เท่ากับ 77 ซึ่งเป็นคำตอบที่ผิด

เนื่องจากนำยอดรวมของ Sum ในเซลล์ B7 ไปรวมเข้าไปอีก

	B7				
	A	B	C	D	E
1					
2		1			
3		2			
4		3			
5		4			
6		5			
7		15			
8		6			
9		7			
10		8			
11		9			
12		10			
13		70			
14					

แทนที่จะเลือกใช้สูตร Sum ตั้งแต่แรก ถ้าเปลี่ยน =SUM(B2:B11) ไปสร้างสูตรบวกแต่ละเซลล์เข้าด้วยกันเป็น =B2+B3+B4+B5+B6+B7+B8+B9+B10+B11 แล้วหากภายหลังมีการแทรก row และใส่ค่าใหม่เพิ่มเข้าไปในเซลล์เช่นที่ยกตัวอย่างข้างต้น ก็จะไม่เกิดปัญหาทำให้สูตรบวกคำนวณค่าผิดพลาด เพราะสูตรบวกยังคงเลือกเฉพาะเซลล์ตามเดิมที่กำหนดไว้ให้เสมอ แต่สูตร

บวกกันทีละเซลล์ไม่ใช่ทางออกที่ดีนักหากมีเซลล์ที่ต้องการนำมาบวกกันนับพันนับหมื่นเซลล์ วิธีที่ดีกว่าการใช้สูตรบวกหรือสูตร Sum ก็คือ การใช้สูตร SubTotal

การใช้สูตร SubTotal ร่วมกับคำสั่ง Filter และ Custom Views

หากแปลคำว่า SubTotal กันตรงๆอาจเข้าใจว่าสูตร SubTotal นี้ทำหน้าที่หายอดรวม Total ของรายการแต่ละกลุ่มที่ Sub เอาไว้ ซึ่งก็แปลงได้ไม่ผิดนัก แต่สูตรนี้มีได้หาคำตอบให้เฉพาะ ยอดรวมเท่านั้น หากยังสามารถดัดแปลงให้คำนวณหาค่าเฉลี่ยหรือนับได้อีกด้วย โดยเลือก กำหนดให้สูตร SubTotal คำนวณหายอดตัวเลขได้ถึง 11 แบบตามโครงสร้างสูตรต่อไปนี้

=SubTotal(ตัวเลขประเภทการคำนวณ, พื้นที่เซลล์)

ตัวเลขประเภทการคำนวณ กำหนดตัวเลขตั้งแต่เลข 101 – 111 แต่เลขที่ใช้กันทั่วไปได้แก่

- 101 กำหนดให้ SubTotal ทำหน้าที่แบบสูตร Average
- 102 กำหนดให้ SubTotal ทำหน้าที่แบบสูตร Count
- 103 กำหนดให้ SubTotal ทำหน้าที่แบบสูตร CountA
- 109 กำหนดให้ SubTotal ทำหน้าที่แบบสูตร Sum

SubTotal มีความสามารถพิเศษเหนือกว่าสูตร Sum กล่าวคือ SubTotal เป็นสูตรที่จะไม่นำค่าจากเซลล์ที่มีสูตร SubTotal มาคำนวณซ้ำอีก

B13		=SUBTOTAL(109,B2:B12)				
	A	B	C	D	E	F
1						
2		1				
3		2				
4		3				
5		4				
6		5				
7		15	สูตรในเซลล์ \$B\$7 : =SUBTOTAL(109,B2:B6)			
8		6				
9		7				
10		8				
11		9				
12		10				
13		55	สูตรในเซลล์ \$B\$13 : =SUBTOTAL(109,B2:B12)			
14						

สังเกตว่าในเซลล์ B7 และ B13 ใช้สูตร SubTotal เพื่อหายอดรวมของตัวเลขจากเซลล์ B2:B6 และ B2:B12 ตามลำดับ โดยในเซลล์ B13 ใช้สูตร =SUBTOTAL(109,B2:B12) แม้ว่าได้กำหนดขอบเขตของตารางที่ต้องการหายอดรวมไว้ตั้งแต่เซลล์ B2:B12 ซึ่งรวมถึงเซลล์ B7 ที่มียอดตัวเลขรวมจากเซลล์ B2:B6 ไว้ก็ตาม สูตร SubTotal ในเซลล์ B13 จะละเลยไม่นำค่าในเซลล์ B7 มาคำนวณซ้ำอีก ทำให้ยังคงได้คำตอบเท่ากับ 55

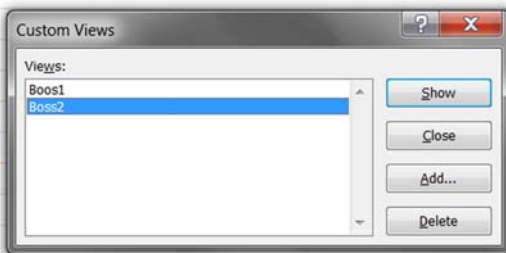
ถ้านำสูตร SubTotal ไปใช้ร่วมกับตารางที่ถูกซ่อน row/column hide หรือกรองจากคำสั่ง Data > Filter จะพบว่า สูตร SubTotal แสดงตัวเลขคำตอบเฉพาะที่ได้มาจากเซลล์ที่ยังมองเห็นตัวเลขอยู่เท่านั้น (Visible Cells Only) โดยตัวสูตรเองไม่ได้ต้องเปลี่ยนแปลงใหม่แต่อย่างใด

(Excel รุ่นเก่ากว่า Excel 2003 จะรับตัวเลขประเภทการคำนวณตั้งแต่เลข 1 - 11 และใช้ร่วมกับการ Filter เท่านั้น แต่ถ้าใช้ Excel 2003 เป็นต้นมาแนะนำให้ใช้ตัวเลข 101 - 111 แทน จะสามารถใช้ร่วมกับ Filter หรือจะซ่อน row/column เองได้อีกด้วย)

B14		=SUBTOTAL(109,B3:B13)	
	A	B	C
1			
2		Num	
5		3	
6		4	
8		7	สูตรในเซลล์ \$B\$8 : =SUBTOTAL(109,B3:B7)
9		6	
10		7	
13		10	
14		30	สูตรในเซลล์ \$B\$14 : =SUBTOTAL(109,B3:B13)
15			

หลังจากที่กรองหรือจัดโครงสร้างตารางเพื่อแสดงผลให้เห็นบนจอตามต้องการได้แล้ว ควรใช้คำสั่ง View > Custom Views > Add เพื่อตั้งชื่อให้กับการแสดงผลตามที่จัดไว้

B14		=SUBTOTAL(109,B3:B13)	
	A	B	C
1			
2		Num	
5		3	
6		4	
7		5	
8		12	
9		6	
10		7	
11		8	
12		9	
13		10	
14		52	สูตรในเซลล์ \$B\$14 : =SUBTOTAL(109,B3:B13)
15			



จากนั้นเมื่อต้องการแสดงโครงสร้างตารางตามแบบใด ให้สั่ง View > Custom Views แล้วคลิกเลือกชื่อแบบที่ตั้งไว้ในช่อง Views แล้วกดปุ่ม Show จะได้โครงสร้างตารางตามแบบที่ตั้งชื่อไว้พร้อมกับสูตร SubTotal แสดงตัวเลขคำตอบเฉพาะเซลล์ที่ยังมองเห็นให้พร้อมกันไป (นอกจากนั้น Custom Views ยังช่วยจำกัดการกำหนด Print Setting แต่ละแบบให้อีกด้วย)

การใช้คำสั่ง Data > Subtotal

Data > SubTotal ใช้กับตารางฐานข้อมูลซึ่งจัดเรียงข้อมูลแยกตามกลุ่มที่ต้องการไว้ก่อนแล้ว และต้องเป็นตารางซึ่งไม่มีสูตรใดๆสร้างไว้ เมื่อใช้คำสั่งนี้จะเกิดสูตร SubTotal หายอดรวมแยกตามกลุ่มให้ทันที พร้อมกับสร้าง Outline เพื่อใช้คลิกเลือกแสดงรายละเอียดทีละชั้นตามต้องการให้อีก และเมื่อนำมาใช้งานร่วมกับ Custom Views จะเป็นเครื่องมือช่วยงานฐานข้อมูลได้อย่างไม่น่าเชื่อ

เริ่มจากจัดเรียงข้อมูลเพื่อจัดกลุ่มข้อมูลที่ต้องการหายอดรวม แยกเฉพาะแต่ละกลุ่มก่อน โดยสั่ง Data > Sort จากนั้นคลิกลงในตารางข้อมูลเซลล์ใดก็ได้ แล้วสั่ง Data > Subtotal

	A	B	C	D	E	F	G	H
1								
2		Month	Region	Section	Sales			
3		Jan	North	Food	100			
4		Jan	North	Machine	200			
5		Jan	North	Retail	200			
6		Jan	South	Retail	100			
7		Jan	South	Machine	120			
8		Jan	South	Food	220			
9		Jan	West	Food	130			
10		Jan	West	Retail	150			
11		Jan	West	Machine	200			
12		Feb	North	Machine	100			
13		Feb	North	Food	190			
14		Feb	North	Retail	200			
15		Feb	South	Food	100			

Subtotal

At each change in:
Month

Use function:
Sum

Add subtotal to:
☐ Month
☐ Region
☐ Section
☒ Sales

☒ Replace current subtotals
☒ Page break between groups
☒ Summary below data

Remove All OK Cancel

- ช่อง At each change in: ให้เลือกชื่อ Header ที่ต้องการหาผลรวมตามกลุ่มของ Header นั้นๆ
- ช่อง Use function: ให้เลือก Sum
- ช่อง Add subtotal to: ให้เลือกชื่อ Header ของ Field ที่ต้องการหาผลรวม
- กดปุ่ม OK จะพบว่า มี Row ใหม่แทรกขึ้นเองและมีสูตร Subtotal คำนวณยอดให้ทันที

หากต้องการสร้าง Subtotal ระดับรองซ้อน Subtotal ระดับแรก ให้สั่ง Data > Subtotal ซ้ำอีกครั้งแล้วเปลี่ยน Header ใหม่ในช่อง At each change in: แล้วให้ตัดกาเครื่องหมายถูกในช่อง Replace current subtotals ออก แล้วกด OK

	1	2	3	4	A	B	C	D	E	F	G	H
	1											
	2					Month	Region	Section	Sales			
	3					Jan	North	Food	100			
	4					Jan	North	Machine	200			
	5					Jan	North	Retail	200			
	6						North Total		500			
	7					Jan	South	Retail	100			
	8					Jan	South	Machine	120			
	9					Jan	South	Food	220			
	10						South Total		440			
	11					Jan	West	Food	130			
	12					Jan	West	Retail	150			
	13					Jan	West	Machine	200			
	14						West Total		480			
	15					Jan Total			1,420			
	16					Feb	North	Machine	100			
	17					Feb	North	Food	190			
	18					Feb	North	Retail	200			
	19											

Subtotal

At each change in:

Region

Use function:

Sum

Add subtotal to:

Month

Region

Section

☒ Sales

☐ Replace current subtotals

☒ Page break between groups

☒ Summary below data

Remove All OK Cancel

เมื่อกดปุ่มเครื่องหมายลบหรือปุ่มหมายเลขระดับชั้นของ Outline ด้านบนซ้าย รายการข้อมูลจะ หุบ และปุ่ม Outline จะกลายเป็นเครื่องหมายบวกแทน หากต้องการ Copy เฉพาะ Visible Cells ให้เลือกพื้นที่ตารางแล้วกด F5 > Special > เลือก Visible cells only จากนั้นสั่ง Copy ไป Paste จะได้เฉพาะพื้นที่ที่เหลือจากการกดปุ่ม Outline เท่านั้น

ข้อดีของการใช้คำสั่งนี้ นอกจาก Excel จะสร้างสูตร SubTotal ให้เองแล้ว หากต้องการสั่ง ย้อนกลับไปเป็นตารางฐานข้อมูลตามแบบเดิม ให้กดปุ่ม Remove All

หมายเหตุ หากต้องการสร้าง Outline ขึ้นมาใช้เอง ให้เลือกเซลล์ที่ต่อเนื่องกันแล้วใช้คำสั่ง Data > Group

การใช้คำสั่ง Data > Consolidate

	A	B	C	D	E	F
1						
2		DIVISION 1				
3			CATEGORY A	CATEGORY B	CATEGORY C	BAHT
4		JANUARY	34,000	84,000	62,000	180,000
5		FEBRUARY	26,000	76,000	54,000	156,000
6		MARCH	12,000	62,000	40,000	114,000
7		QUARTER 1	72,000	222,000	156,000	450,000
8						
9		DIVISION 4				
10			CATEGORY A	CATEGORY B	CATEGORY C	BAHT
11		JANUARY	56,000	106,000	84,000	246,000
12		FEBRUARY	43,000	93,000	71,000	207,000
13		MARCH	25,000	75,000	53,000	153,000
14		QUARTER 1	124,000	274,000	208,000	606,000
15						

จากภาพนี้ตารางของ Division 1 และ Division 4 มีโครงสร้างเหมือนกัน ถ้าต้องการหายอดรวมตัวเลขเรื่องเดียวกันจากหลายตารางเข้าด้วยกัน ให้เริ่มจากเปิดชีทใหม่ขึ้นมาก่อน แล้วสั่ง Data > Consolidate จากนั้นคลิกเลือกพื้นที่ตารางข้อมูลชุดแรกให้แสดงในช่อง Reference แล้วกดปุ่ม Add แล้วเลือกพื้นที่จากตารางอื่นมา Add ซ้ำจนกว่าจะครบทุกตารางที่ต้องการ (โดยจะเลือกจากตารางที่อยู่ต่างชีทหรือต่างแฟ้มกันก็ได้) แล้วอย่าลืมกาช่อง Use labels in พร้อมกับกาช่อง Create links to source data ให้ครบเพื่อจะได้สูตรลิงค์ค่าจากตารางต้นทางมาใช้ด้วย

The screenshot shows an Excel spreadsheet with a consolidated table. The table has columns A through H. Column A contains labels for months and quarters. Column B contains the word 'Data'. Column C contains the word 'Data'. Column D contains the word 'Data'. Column E contains the word 'Data'. Column F contains the word 'Data'. Column G contains the word 'Data'. Column H contains the word 'Data'. The formula bar shows the formula =SUM(D3:D4). The Consolidate dialog box is open, showing the function Sum, the reference Consolidate!\$B\$3:\$F\$7, and the all references Consolidate!\$B\$3:\$F\$7 and Consolidate!\$B\$10:\$F\$14. The dialog box also has checkboxes for Use labels in (Top row, Left column) and Create links to source data.

	A	B	C	D	E	F	G	H
1								
2				CATEGORY A	CATEGORY B	CATEGORY C	BAHT	
3			Data	34,000	84,000	62,000	180,000	
4			Data	56,000	106,000	84,000	246,000	
5		JANUARY		90,000	190,000	146,000	426,000	
6			Data	26,000				
7			Data	43,000				
8		FEBRUARY		69,000				
9			Data	12,000				
10			Data	25,000				
11		MARCH		37,000				
12			Data	72,000				
13			Data	124,000				
14		QUARTER 1		196,000				
15								
16								
17								
18								

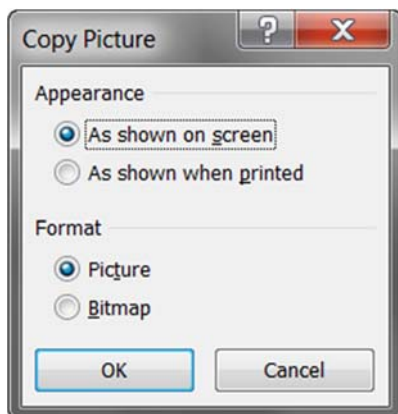
วิธีสร้างสูตรบนกราฟ

กราฟ Excel มีความพิเศษอย่างหนึ่ง คือ แทบทุกสิ่งซึ่งประกอบกันเป็นรูปกราฟ สามารถนำสูตร link ค่าจากตารางไปที่กราฟ ไม่ว่าจะเป็นตัว marker หรือค่าอธิบายแต่ละส่วน เราสามารถสร้างสูตร link ในแต่ละจุดได้โดยใช้วิธีง่ายๆ ต่อไปนี้

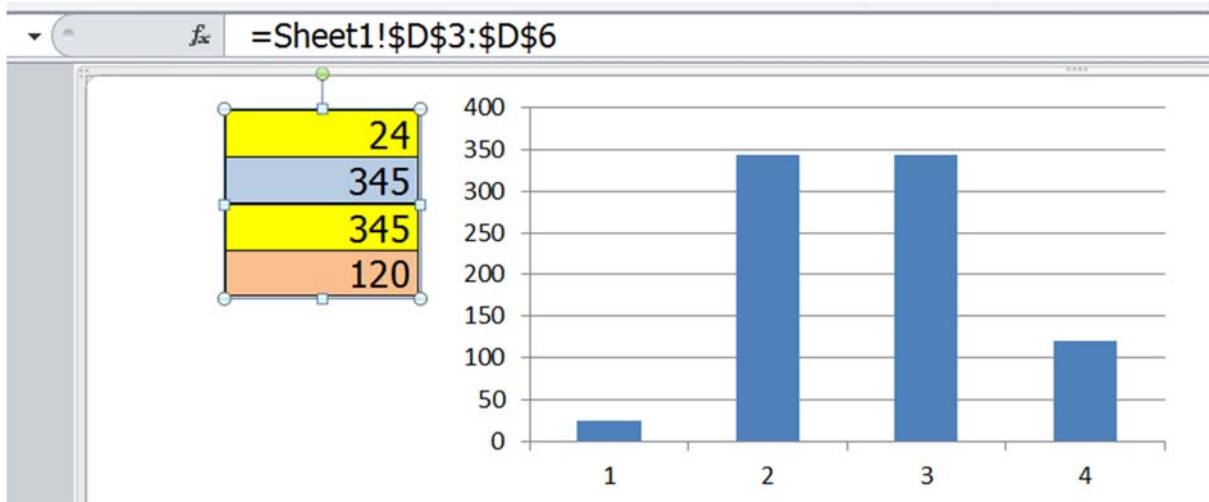
1. สร้างกราฟให้เสร็จก่อนตามวิธีปกติ
2. คลิกครั้งหนึ่ง ลงไปที่ตัว marker หรือข้อความใดๆก็ได้บนกราฟ แล้วคลิกลงไปที่อีกที่ที่เต็ม จะพบว่า จุดนั้นกลายเป็นกล่อง
3. ให้คลิกลงบน formula bar แล้วพิมพ์เครื่องหมายเท่ากับ =
4. คลิกกลับไปยังเซลล์ข้อมูลต้นต่อในตาราง แล้วกด enter
5. จะพบว่าตัว marker กลายเป็นสูตร link ข้อมูลมาจากเซลล์

ด้วยวิธีเดียวกันนี้ เราสามารถสร้างสูตร link นำข้อมูลมาทั้งตาราง นำมาแสดงไว้บนกราฟได้เลย ทั้งตาราง เริ่มต้นโดย

1. เลือกตารางข้อมูลซึ่งต้องการนำไปแสดงบนกราฟ
2. กด Shift พร้อมกับคลิกเมนูคำสั่ง Edit > Copy Picture แล้วกดปุ่ม OK



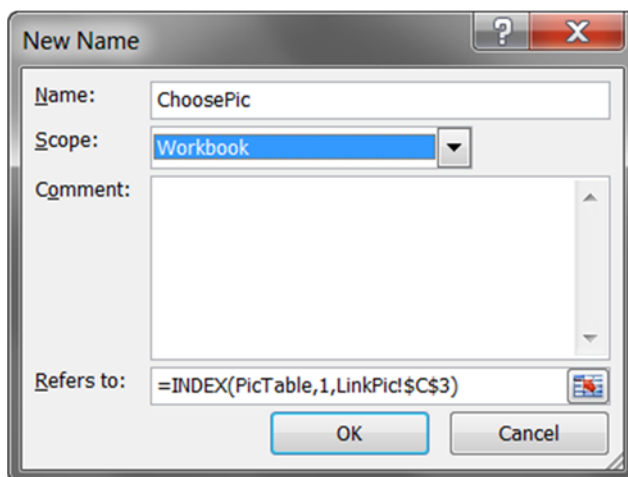
3. คลิกลงไปในหน้ากราฟ แล้วสั่ง Paste แล้วย้ายรูปตารางไปยังตำแหน่งว่างบนกราฟ จะได้เห็นตารางชัดเจนไม่ทับกับรูปส่วนอื่นๆ
4. คลิกลงไปใน formula bar แล้วพิมพ์เครื่องหมายเท่ากับ =
5. คลิกกลับไปยังพื้นที่ตารางข้อมูลซึ่งสั่ง copy picture มาจะได้สูตร link มาที่รูป
6. กด Enter



วิธีสร้างสูตรเพื่อดึงรูปภาพ

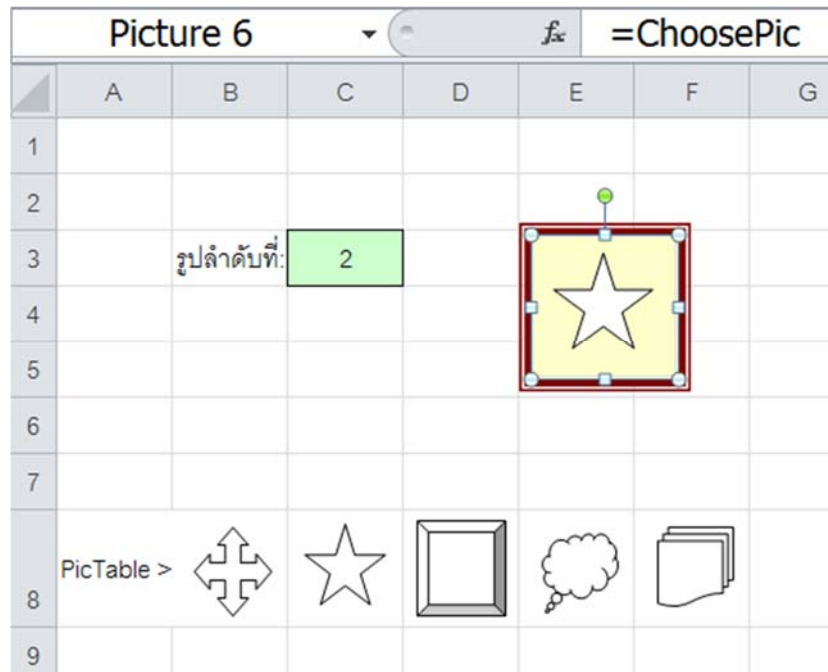
สูตร Excel สามารถใช้ link รูปภาพมาแสดงผลได้เช่นเดียวกับการสร้างสูตร link เพื่อนำค่าในเซลล์อื่นมาแสดง โดยเริ่มต้นจาก

1. สร้างตารางเก็บรูปภาพ เซลล์ละรูป แล้วกำหนดชื่อตารางเก็บรูปนี้ว่า PicTable หรือชื่ออื่นก็ได้
2. ใช้เมนูคำสั่ง Formulas > Define Name สร้าง Formula Name ชื่อว่า ChoosePic กำหนดให้ Refers to: สูตร =INDEX(PicTable,1,Sheet1!\$A\$1)



3. เลือกเซลล์ว่างๆเซลล์ใดก็ได้ แล้วสั่ง Home > Copy as Picture
4. เลือกตำแหน่งเซลล์ที่จะแสดงรูป แล้วสั่ง Paste จะได้รูปเซลล์ว่างๆที่เรา copy มา
5. คลิกลงไปใน Formula Bar พิมพ์สูตร =ChoosePic

6. กำหนดเลขที่รูปลงไปในเซลล์ A1 เช่น ใส่เลข 2 จะได้รูปที่ 2 ถูกนำมาแสดงในรูปเซลล์ที่ copy มา
7. คลิกขวาที่รูปเซลล์ แล้วจัดรูปแบบของสีและเส้นกรอบตามต้องการ



การจัดการกับความอ้วนอ้าย...ของแฟ้ม

แฟ้ม Excel ที่เราเพิ่งเปิดใช้งานขึ้นใหม่ในวันนี้ จะเริ่มมีขนาดแฟ้มใหญ่ขึ้นเรื่อยๆ แล้วเมื่อวันเวลาผ่านไปสักปีสองปี คอมพิวเตอร์ที่ใช้อยู่ก็จะทำงานช้าลงไปจนผิดสังเกตโดยมีเหตุมาจากแฟ้มใหญ่ขึ้นมากนี้แหละ ตัวผู้ใช้ Excel มักชอบวิ่งไปบอกหัวหน้า ขอให้ซื้อคอมพิวเตอร์เครื่องใหม่เพื่อจะได้ทำให้ Excel ทำงานเร็วเหมือนเดิม ซึ่งถ้าผมเป็นหัวหน้าของคุณล่ะก็ ... แทนที่จะเปลี่ยนคอมพิวเตอร์เครื่องใหม่ ขอเปลี่ยนตัวพนักงานเป็นคนอื่นจะดีกว่า !!! เพราะแม้จะได้ใช้คอมพิวเตอร์เครื่องใหม่แล้วก็ตาม ต่อมาพอแฟ้มเริ่มใหญ่ขึ้นอีก คอมพิวเตอร์ก็จะช้าลงไปอีก แล้วเราก็ต้องซื้อเครื่องใหม่ที่เร็วขึ้นกว่าอีกหรืออย่างไรงั้น

ก่อนที่จะสายเกินไป ผู้ใช้ Excel ควรเข้าใจก่อนว่า ที่ว่าแฟ้ม Excel ใหญ่ขึ้นนั้น อะไรบ้างที่เป็นเหตุให้แฟ้มมีขนาดใหญ่ขึ้น แล้วอย่านึกต่อไปว่า แฟ้มใหญ่ต้องใช้เวลาคำนวณนานกว่าแฟ้มที่เล็กกว่าเสมอไป ซึ่งเมื่อเข้าใจประเด็นเหล่านี้แล้วเราต้องหาทางออกแบบแฟ้มให้ใหญ่พอดีๆ ไม่ใหญ่ ไม่เล็กเกินไป และไม่ใช้เวลาคำนวณช้าจนรอไม่ไหว

สิ่งที่มีผลต่อขนาดของแฟ้ม

1. จำนวนเซลล์ที่ใช้
2. จำนวนเซลล์ที่ถูกอ้าง
3. รูปแบบหรือ Format ที่ใช้

จำนวนเซลล์ที่ใช้

แฟ้ม Excel ในยุคแรก มีขนาดแฟ้มใหญ่มากเพราะกำหนดว่า พื้นที่ตารางทุกเซลล์ตั้งแต่เซลล์ A1 ไปจนถึงเซลล์หัวมุมขวาล่างสุดที่มีข้อมูลอยู่ถือเป็นพื้นที่ซึ่งถูกใช้งาน ต่อมา Excel ได้พัฒนาให้ขนาดของแฟ้มไม่ต้องนับจากเซลล์ A1 แต่ให้นับเฉพาะเซลล์ที่ถูกใช้งานเท่านั้น ส่วนเซลล์ที่เราไม่ได้เข้าไปแตะต้องแก้ไขใส่อะไรลงไป ไม่มีส่วนกระทบต่อขนาดของแฟ้มแต่อย่างใด ซึ่งช่วยให้เราสามารถใส่พื้นที่ตารางได้สะดวก ไม่ต้องออกแบบตารางให้ชิดติดกันเป็นพืดเพื่อหาทางลดขนาดแฟ้มอย่างแต่ก่อน

ขอให้สังเกตว่า ขนาดของแฟ้ม ไม่ได้ขึ้นอยู่กับจำนวนอักขระที่มีอยู่ในเซลล์ ไม่ว่าจะพิมพ์เลข 1 ลงไปในเซลล์ 100 เซลล์ หรือพิมพ์เลข 123 ลงไปในเซลล์ 100 เซลล์ แฟ้มก็ยังคงมีขนาด

ใหญ่เท่ากันเพราะใช้จำนวนเซลล์ 100 เซลล์เท่ากัน แต่อย่างไรก็ตามในเซลล์จำนวน 100 เซลล์ที่สมมติว่าถูกพิมพ์ค่าลงไปนั้น หากพิมพ์ตัวอักษรหรือสร้างสูตรลงไปก็จะทำให้แฟ้มมีขนาดใหญ่ขึ้นตามลักษณะของข้อมูลที่พิมพ์ลงไป ซึ่งลำดับขนาดแฟ้มจากเล็กไปใหญ่ขึ้นกับประเภทของข้อมูล ดังนี้

1. ถ้าเป็นตัวเลข แฟ้มจะเล็กที่สุด ไม่ว่าจะเป็นเลข 1 หรือเลข 123 แฟ้มยังคงมีขนาดเท่ากัน
2. ถ้าเป็นตัวอักษร แฟ้มจะใหญ่ขึ้น ไม่ว่าจะเป็นตัวอักษร A หรือ ABC แฟ้มยังคงมีขนาดเท่ากัน (ใน Excel 2010 พบว่า ข้อ 1 และ 2 นี้ไม่ได้มีขนาดแฟ้มต่างกัน)
3. ถ้าเป็นสูตรและคืนค่าเป็นตัวเลข แฟ้มจะใหญ่ขึ้น
4. ถ้าเป็นสูตรและคืนค่าเป็นตัวอักษร แฟ้มจะใหญ่มากที่สุด

จำนวนเซลล์ที่ถูกอ้าง

ในกรณีที่สร้างสูตร link ข้อมูลข้ามแฟ้ม จะพบว่าแฟ้มปลายทางมีขนาดใหญ่ขึ้นมาก หรือบางครั้งจะมีขนาดใหญ่กว่าแฟ้มต้นทางเสียอีก ทั้งๆที่ในแฟ้มปลายทางมีเซลล์ที่ใช้งานเพียงเซลล์เดียวก็ตาม เนื่องจากในสูตร link ข้ามแฟ้มที่สร้างไว้ในเซลล์ได้นำข้อมูลจากแฟ้มต้นทางมาเก็บไว้ในแฟ้มปลายทาง เรียกว่า Cache Data

ยกตัวอย่างเช่น สูตร =VLOOKUP(A1,[Input.xls]Sheet1!\$B\$2:\$H\$200,2) ที่หาค่าจากแฟ้ม Input.xls ในชีท Sheet1 ตั้งแต่เซลล์ B2:H200 นี้ จะทำให้ Excel แอบสร้างตารางที่เรามองไม่เห็นเก็บข้อมูลจากแฟ้มต้นทางทั้งหมดไว้ในแฟ้มปลายทางที่มีสูตรนี้อยู่ ซึ่งตารางที่มองไม่เห็นนี้แหละเป็น Cache Data

Cache Data ช่วยทำให้เมื่อเปิดแฟ้มปลายทางขึ้นมาใช้เพียงแฟ้มเดียว เราก็สามารถใช้สูตร VLookup ค้นหาข้อมูลจากแฟ้มต้นทางได้โดยไม่ต้องเปิดแฟ้มต้นทางขึ้นมาใช้พร้อมกับแฟ้มปลายทางแต่อย่างใด แต่ก็ทำให้แฟ้มปลายทางมีขนาดใหญ่ขึ้น

รูปแบบหรือ Format ที่ใช้

การกำหนดรูปแบบให้กับชีททั้งชีทพร้อมกันทีเดียวทุกเซลล์ จะทำให้แฟ้มมีขนาดเล็กกว่าการกำหนดรูปแบบเพียงบางส่วนของตาราง เพราะ Excel จะใช้วิธีรับรู้ที่เดียวว่าชีทนั้นๆใช้รูปแบบใด

แต่ถ้าในตารางมีการกำหนดรูปแบบต่างกันไปในแต่ละส่วน จะทำให้ Excel ต้องรับรู้ละเอียดมากขึ้นว่า พื้นที่ตารางส่วนใดใช้รูปแบบใด จึงทำให้แฟ้มมีขนาดใหญ่ขึ้น

ตามปกติ Excel ได้กำหนดไว้ในคำสั่ง Excel Options ให้กาช่อง Extend data range formats and formulas ไว้แล้วตั้งแต่ต้น เพื่อช่วยให้เซลล์ข้อมูลตัวเลขที่ใช้งานติดต่อกันไว้ตามแนวตั้ง ตั้งแต่ 3 เซลล์ขึ้นไป ส่งรูปแบบที่กำหนดไว้ใน 3 เซลล์นั้นต่อไปยังเซลล์ที่ 4 และเซลล์ต่อไปให้ทันที แล้วหากมีเซลล์ที่สร้างสูตรไว้ติดต่อกันตามแนวตั้ง 4 เซลล์อยู่แล้ว คำสั่งนี้จะช่วย copy สูตรและรูปแบบจากเซลล์ที่ 4 ไปยังเซลล์ที่ 5 ต่อไปเรื่อยๆให้เช่นกัน (ขอให้ท่องไว้ว่า ครบ 3 ได้ 4 ครบ 5 ได้สูตร หมายความว่า เดิมครบ 3 เซลล์ จะได้รูปแบบต่อไปในเซลล์ที่ 4 แล้วพอเริ่มครบ 5 เซลล์ จะได้ทั้งสูตรและรูปแบบตามต่อไปให้เอง) อย่างไรก็ตามคำสั่งนี้จะทำงานต่อเมื่อข้อมูลตามแนวตั้งที่บันทึกต่อกันไปเป็นตัวเลขทุกเซลล์ เช่น ถ้าบันทึกตัวเลขเอาไว้ ก็ต้องบันทึกตัวเลขในเซลล์ต่อไปเท่านั้น หากเมื่อใดที่บันทึกค่าเป็นตัวอักษรลงไปในเซลล์ที่ควรเป็นตัวเลข ต้องลบค่านั้นทิ้งไปก่อน จึงจะกระตุ้นให้คำสั่งนี้ทำงานดังเดิม

สิ่งที่ผลต่อความเร็ว

ในเรื่องความเร็วนี้ไม่ได้เกี่ยวข้องกับความเร็วในการคำนวณของโปรแกรม Excel เท่านั้น แต่ยังส่งผลกระทบต่อความเร็วของมนุษย์ในการใช้ Excel อีกด้วย นับตั้งแต่จะช้าหรือเร็วในการสร้างงาน สร้างเสร็จแล้วเมื่อนำมาแก้ไขจะต้องเสียเวลาแก้ไขช้าหรือเร็วเพียงใดกว่าจะแก้ไขเสร็จ บางคนใช้เวลาสร้างแฟ้มนิดเดียวแต่พอย้อนกลับมาแก้ไขแฟ้มเดิมกลับต้องเสียเวลาแก้ไขนานขึ้นมาก

มนุษย์นี้แหละต้นเหตุของความช้าอืดอ้าย

แม้ยุคนี้เราใช้เครื่องคอมพิวเตอร์แทนเครื่องคิดเลขกันมากขึ้น แต่แทนที่เครื่องคอมพิวเตอร์จะช่วยให้ทำงานได้เร็วขึ้น หลายนคนกลับทำงานเสร็จช้าลง จนแทบอยากจะย้อนไปกดปุ่มบนเครื่องคิดเลขที่ตนใช้ได้เร็วกว่า ทั้งนี้เนื่องจากคนเราไม่ชอบการเปลี่ยนแปลง ชอบทำอะไรใช้อะไรที่ตัวคุ้นเคยมาก่อน พอจะใช้คอมพิวเตอร์ก็คิดไม่ออกว่าจะใช้โปรแกรมอะไร ฟังเขามาว่าถ้าเรื่องของตัวเลขต้องใช้โปรแกรม Excel แต่ถ้าเรื่องเก็บข้อมูลต้องใช้ Access แล้วถ้าข้อมูลมีทั้งตัวเลขและตัวอักษรละ จะใช้โปรแกรมอะไรดี บางคนอาจคิดง่ายๆว่า ถ้าข้อมูลมีเลขมากกว่าก็ให้ใช้ Excel หรือถ้าข้อมูลมีตัวอักษรมากกว่าตัวเลข ก็ให้ใช้ Access แล้วกัน

ในประเด็นของการเลือกใช้โปรแกรม Excel หรือ Access นี้ ในเบื้องต้นสำหรับคนที่เพิ่งจับคอมพิวเตอร์ ขอแนะนำให้เลือกใช้ Excel กับงานที่ต้องการความยืดหยุ่น หรืองานที่ผู้ใช้อย่างไม่มั่นใจว่าจะจัดโครงสร้างหน้าตาของตารางข้อมูลที่ต้องการเก็บไว้อย่างไร เพราะ Excel มีคำสั่งช่วยสำหรับการโยกย้ายข้อมูลเตรียมไว้พร้อม และเราอาจจะย้ายตารางไปเก็บไว้ที่ชีทอื่นแฟ้มอื่น ก็จะเห็นผลทันทีบนหน้าจอกับตา แถม Excel ยังช่วยแก้ไขสูตรที่เกี่ยวข้องกับตารางซึ่งถูกย้ายให้เสร็จ แล้วต่อมาภายหลังพอมีข้อมูลปริมาณมากขึ้น เริ่มมีมาตรฐานของโครงสร้างตารางแน่นอนตายตัวแล้ว จึงคิดถึง Access

ครั้นเลือกใช้ Excel สร้างตารางเก็บข้อมูลต่อไปสักพัก คอมพิวเตอร์แสนเร็วของเรากับผู้ใช้ Excel ก็จะพร้อมใจกันทอดคอทำงานช้าลงไปเรื่อยๆ โดยมีสาเหตุยอดนิยมดังต่อไปนี้

1. ออกแบบตารางเก็บข้อมูลไม่เป็น พอเปิดแฟ้มขึ้นดู แล้วงงไปหมดว่าข้อมูลอะไรอยู่ตรงไหน
2. เก็บข้อมูลที่เกี่ยวข้องกับวันที่ แต่วันที่ซึ่งเก็บบันทึกไว้ ไม่สามารถนำมาใช้คำนวณต่อได้
3. เก็บข้อมูลซ้ำซ้อนกัน พอจะแก้ไขเป็นข้อมูลใหม่ต้องเสียเวลาหาตำแหน่งเซลล์ให้พบ ก่อนว่าอยู่ที่ชีทใดแฟ้มใด
4. แยกข้อมูลที่เก็บไว้ในตารางเดียวกันออกจากกันไม่ได้ ต้องฝืนบันทึกข้อมูลต่อไปเรื่อยๆ นับพันนับหมื่นบรรทัด
5. สร้างสูตรยากๆยาวๆแบบฉันทองคนสร้างคนเดียวเท่านั้นที่จะแกะออก
6. เลือกใช้สูตรไม่เป็น ชอบใช้สูตรสั้นๆที่คนชอบ แต่ Excel ไม่ชอบเพราะโปรแกรมจะทำงานช้าลง
7. แยกตารางสูตรที่ไม่จำเป็นต้องคำนวณพร้อมกันออกจากกันไม่ได้ ทำให้ต้องเสียเวลารอการคำนวณสูตรซ้ำโดยไม่จำเป็น
8. ไม่รู้จักแยกค่าตัวแปรที่อาจเปลี่ยนแปลงภายหลังไว้นอกสูตร
9. สร้างชีทที่ใช้คำนวณเรื่องเดียวกันซ้ำกันเต็มไปหมด แต่ละชีทมีตารางหน้าตาเดียวกันต่างกันแต่ค่าที่นำมาใช้คำนวณ
10. เลือกใช้ VBA กับงานที่ยังไม่ควรใช้ VBA แม้ VBA จะช่วยให้เสร็จงานเร็วขึ้น แต่ก็ไม่ใช่เร็วขึ้นได้เต็มที่อย่างที่ควร

สาเหตุยอดนิยมเหล่านี้แหละที่ทำให้เราได้ผลลัพธ์จาก Excel ช้าลง แต่โดยทั่วไปผู้ใช้ Excel ที่ไม่เคยนำผลงานไปเปรียบเทียบกับผู้อื่นมักจะหลงภาคภูมิใจในผลงานของตน และเข้าใจผิดไปว่าตนได้ใช้ Excel ช่วยสร้างผลงานที่เร็วขึ้น ดังนั้นการให้พื้นฐานในเรื่องแนวทางการใช้ Excel ที่ดีและถูกต้องจึงเป็นเรื่องที่สำคัญมาก

แนวทางการใช้ Excel ที่ดีและถูกต้อง

- เราต้องมุ่งหวังที่จะใช้ Excel ให้ได้ผลงานมากขึ้นและเร็วขึ้นแบบคอมพิวเตอร์ นั่นคือต้องสามารถนำแฟ้มที่สร้างไว้แล้วกลับมาใช้ซ้ำแล้วซ้ำอีก โดยไม่จำเป็นต้องแก้ไขสูตรหรือโครงสร้างตารางที่สร้างไว้แล้วอีกเลย
- แยกโครงสร้างตารางเป็น 3 ประเภท คือ ตารางที่ใช้เก็บข้อมูล(ตัวเลขและหรือตัวอักษร) ตารางสำหรับสร้างสูตรคำนวณ และตารางสำหรับแสดงเป็นรายงาน
- พึงระลึกไว้เสมอว่า สักวันหนึ่งข้อมูลในแฟ้มจะถูกเก็บไว้จนทำให้แฟ้มใหญ่และ Excel ทำงานช้าลง ดังนั้นต้องเตรียมโครงสร้างตารางที่พร้อมต่อการโยกย้ายไปเก็บไว้ในแฟ้มอื่น
- ข้อมูลที่เก็บไว้ในตารางที่ใช้เก็บข้อมูลต้องจัดเก็บไว้ในลักษณะโครงสร้างฐานข้อมูลที่สามารถนำไปใช้กับเมนู Data หรือใช้กับสูตรพวก Lookup ได้ทันที
- ข้อมูลที่เป็นตัวเลข ต้องมีค่าเป็นตัวเลขอย่างแท้จริง ซึ่งมีลักษณะที่ต้องขีดขวาของเซลล์โดยไม่จำเป็นต้องใช้ Format
- ห้ามพิมพ์ข้อมูลใดๆซ้ำโดยไม่จำเป็น แต่ให้ใช้สูตรนำค่าส่งต่อไปใช้เมื่อต้องการใช้ข้อมูลนั้นซ้ำ
- ในชีทหนึ่งๆ พยายามสร้างสูตรเดียวซึ่งสามารถนำไปใช้ได้กับทุกตำแหน่งเซลล์ในชีทนั้นๆ ไม่ว่าจะมีการแก้ไขที่ทำให้ตำแหน่งโครงสร้างตารางเปลี่ยนไปอย่างไรก็ตาม
- สูตรที่สร้างขึ้นต้องใช้หลักการคำนวณที่ง่ายต่อการทำความเข้าใจของตนเองและผู้อื่น และเมื่อที่เงื่อนไขเปลี่ยนแปลงไปจากเดิมก็ไม่จำเป็นต้องแก้ไขโครงสร้างสูตรให้ต่างไปจากเดิมมากนัก
- ต้องใช้สูตรปรับค่าในเซลล์ให้ได้ค่าถูกต้องตามที่ต้องการก่อนแล้วจึงใช้คำสั่ง Format ปรับการแสดงผลของค่านั้นในภายหลัง
- ในโครงสร้างสูตร ห้ามใส่ค่าคงที่ใดๆไว้อย่างเด็ดขาด
- แทนที่จะสร้างสูตรยาวๆยาวๆที่มีส่วนการคำนวณที่ซ้ำกันลงไปในตารางนับพันนับหมื่นเซลล์ ซึ่งทำให้แฟ้มใหญ่และคำนวณช้าลง ควรหาทางกระจายโครงสร้างสูตรออกเป็นสูตรสั้นๆหลายๆเซลล์แต่ใช้การคำนวณครั้งเดียว เพื่อให้แฟ้มเล็กลงและคำนวณเร็วขึ้น
- อย่าลืมว่า ตารางที่เป็นสูตร จะมีขนาดแฟ้มใหญ่กว่าตารางที่เป็นตัวเลขหรือตัวอักษร ดังนั้นแทนที่จะใช้สูตรคำนวณค่าใดๆที่เป็นค่าคงที่ เช่น เลขลำดับ แนะนำให้ใช้ตารางที่เป็นตัวเลขผลจากการคำนวณแทนไปเลยดีกว่า

- ในกรณีที่ต้องสร้างสูตรที่อ้างอิงกับตารางแนวนอนแนวดิ่ง แนะนำให้เลือกตารางตามแนวนอนก่อนตารางแนวดิ่งเพื่อสร้างมาตรฐานวิธีสร้างสูตรของตนเอง และนอกจากนั้นยังสอดคล้องกับโครงสร้างของสูตร Excel เช่น Index หรือ Offset จะใช้ตำแหน่งอ้างอิงตามแนวนอนก่อนแนวดิ่งเสมอ
- หากสร้างสูตรที่ต้องนำมาใช้ซ้ำแล้วซ้ำอีก ควรตั้งชื่อตำแหน่งเซลล์ (Range name) ที่สื่อถึงที่ไปที่มาของสูตร แทนการอ้างอิงกับตำแหน่งอ้างอิงโดยตรง
- ควรแยกสูตรที่ไม่จำเป็นต้องคำนวณพร้อมกันออกเป็นแฟ้มแยกจากกัน (หรืออีกนัยหนึ่งให้เก็บเฉพาะสูตรที่ต้องคำนวณพร้อมกันเท่านั้นไว้ในแฟ้มเดียวกัน) เพื่อให้เราสามารถเลือกเปิดแฟ้มตามขอบเขตการคำนวณที่ชัดเจน และทำให้แฟ้มมีขนาดเล็กลง
- ให้เลือกใช้ VBA เฉพาะเมื่อต้องการใช้ Excel ที่เหนือกว่าปกติจากสูตรและเมนูคำสั่งที่ทำได้ และรหัส VBA ที่ใช้ต้องสามารถใช้ได้ต่อเนื่องไปตลอดแม้โครงสร้างตารางจะเปลี่ยนแปลงต่างไปจากเดิมก็ตาม
- ทุกอย่างมีได้ก็ต้องมีเสีย ได้อย่างก็ต้องเสียอย่าง ขอให้พิจารณาเปรียบเทียบข้อดีข้อเสียก่อนที่จะสร้างงาน เพื่อให้เหมาะสมกับเงื่อนไขและสภาพแวดล้อมของการทำงาน

แนวทางการลดความอ้วน

ลองคิดดูว่า ถ้าเราเก็บรายการข้อมูลลงไปในชีทเต็มทั้งหมดทุก row แล้ว จะนำข้อมูลรายการใหม่ไปเก็บไว้ที่ไหน จะเริ่มต้นเก็บไว้ในชีทใหม่ในแฟ้มเดิมหรือจะเปิดแฟ้มใหม่?

ผู้ใช้ Excel ส่วนมากชอบเก็บข้อมูลไว้เรื่อยๆในชีทเดิมแฟ้มเดิมเพราะสะดวกในการค้นหาเรียกใช้ ซึ่งหากใช้คอมพิวเตอร์รุ่นใหม่มีความเร็วสูงก็ไม่ต้องรอนานนัก แต่ถ้าเก็บข้อมูลไว้ในแฟ้มเดิมต่อไปเรื่อยๆ สุดท้ายเจ้าเครื่องคอมพิวเตอร์ที่ว่าทำงานเร็วก็จะทำงานช้าลงช้าลง นี่เป็นเพราะเราชอบเก็บข้อมูลไว้ในแฟ้มเดิมต่อไป และก็ไม่เคยมีความจำเป็นต้องใช้ข้อมูลทั้งหมดที่เก็บไว้พร้อมกันสักครั้งเดียว

แล้วสูตรคำนวณที่สร้างไว้ มันต้องถูกนำมาคำนวณใหม่พร้อมกันทุกครั้งที่มีข้อมูลเปลี่ยนแปลงไปแค่เซลล์เดียวหรือไม่ สูตรที่คำนวณหาต้นทุนสินค้าหรือค่าใช้จ่ายซึ่งมีค่าคงที่ ซึ่งนานๆจึงจะมีค่าเปลี่ยนแปลงไปสักที จำเป็นหรือที่รวมตารางคำนวณค่าคงที่เหล่านั้นไว้ในแฟ้มซึ่งมีสูตรคำนวณยอดกำไรจากการขาย

จากนั้นเมื่อถึงคราวต้องนำข้อมูลนับพันนับหมื่นบรรทัด หรือตารางรายงานที่ต้องพิมพ์นับร้อยนับพันหน้ามาแสดงให้ดูนั้น เราใช้สายตาดูตารางทั้งหมดหรือกระดาษทุกแผ่นที่พิมพ์ออกมาพร้อมกันทุกหน้าหรือไม่?

ผู้สร้างงานต้องพิจารณาถึงประเด็นดังกล่าวข้างต้นไว้เสมอ ต้องพร้อมที่จะแยกข้อมูลที่เก็บไว้ในแฟ้มไปเก็บไว้ในแฟ้มอื่น ข้อมูลใดที่จำเป็นต้องเก็บไว้ใช้ เพื่อหาผลลัพธ์ที่ต้องการจากการคำนวณร่วมกันทุกครั้ง ก็ควรจัดเก็บไว้ในแฟ้มเดียวกับตารางที่เป็นสูตรคำนวณ แต่ถ้าไม่จำเป็นต้องใช้คำนวณพร้อมกันก็ควรแยกออกเป็นแฟ้มต่างหากทำให้แฟ้มมีขนาดเล็กลง แล้วจึงใช้สูตร link ข้อมูลข้ามแฟ้มมาใช้คำนวณต่อตามแต่เราจะเลือกเอาข้อมูลมาใช้จากแฟ้มไหน ซึ่งจะช่วยทำให้ใช้ตารางสูตรหน้าเดียวตารางเดียวแต่สามารถคำนวณผลลัพธ์ได้ตามแต่ข้อมูลที่ใช้ link มาใช้อีกต่อหนึ่ง

ในกรณีที่ link ข้อมูลข้ามชีท ให้ใช้สูตร Indirect เลือกนำข้อมูลจากชื่อชีทที่ต้องการมาใช้ในชีทคำนวณ ส่วนกรณีที่ link ข้ามแฟ้ม ให้ใช้คำสั่ง Change Source เพื่อเลือกชื่อแฟ้มที่ต้องการมาใช้ในแฟ้มคำนวณ

ผลจากสูตรที่มีต่อขนาดแฟ้มและความเร็วในการคำนวณ

ขอนำผลการทดสอบจากการใช้สูตรคำนวณหลายหลายประเภทมาแล้วสู่กันฟัง และขอให้ลองหาข้อสรุปกันเองว่า สูตรแบบไหนดีกว่ากันและเพราะเหตุใด (โดยการทดสอบนี้ได้จากการสร้างสูตรลงไปในตารางขนาดใหญ่ตั้งแต่เซลล์ D1:D55,000 และใช้ Excel 2003 กับ Pentium D 3.0 GHz)

กรณีใช้สูตร SumIF เพื่อหายอดรวมตามเงื่อนไขเดียว

- สูตร SumIF ใช้เวลา 4 วินาที / ขนาดแฟ้ม 2,984 KB เช่น
=SumIF(IDRange,ID,NumRange)
- สูตร SumIFArray ใช้เวลา 38 วินาที / ขนาดแฟ้ม 6,089 KB เช่น
{=Sum(IF(IDRange=ID,NumRange))}
- สูตร SumArray ใช้เวลา 47 วินาที / ขนาดแฟ้ม 5,547 KB เช่น
{=Sum((IDRange=ID)*NumRange)}
- สูตร SumProduct แบบใส่เครื่องหมาย -- ใช้เวลา 44 วินาที / ขนาดแฟ้ม 2,987 KB เช่น =SumProduct(--(IDRange=ID),NumRange)
- สูตร SumProduct แบบใส่เครื่องหมายคูณ ใช้เวลา 50 วินาที / ขนาดแฟ้ม 2,986 KB เช่น =SumProduct((IDRange=ID)*NumRange)

กรณีใช้สูตร SumIF เพื่อหายอดรวมตามเงื่อนไข 2 เงื่อนไขต่อเนื่องกัน

- สูตร SumIF ใช้ได้เพียงเงื่อนไขเดียวจึงไม่ได้นำมาทดสอบ
- สูตร SumIFArray ใช้เวลา 2 นาที 23 วินาที / ขนาดแฟ้ม 9,716 KB เช่น
{=Sum(IF(IDRange=ID,IF(KeyRange=Key,NumRange)))}
- สูตร SumArray ใช้เวลา 2 นาที 39 วินาที / ขนาดแฟ้ม 8,633 KB เช่น
{=Sum((IDRange=ID)*(KeyRange=Key)*NumRange)}
- สูตร SumProduct แบบใส่เครื่องหมาย -- ใช้เวลา 2 นาที 30 วินาที / ขนาดแฟ้ม 5,168 KB เช่น =SumProduct(--(IDRange=ID),--(KeyRange=Key),NumRange)
- สูตร SumProduct แบบใส่เครื่องหมายคูณ ใช้เวลา 2 นาที 33 วินาที / ขนาดแฟ้ม 5,165 KB เช่น =SumProduct((IDRange=ID)*(KeyRange=Key),NumRange)

- สูตร SumProduct แบบใส่เครื่องหมายคูณทุกตัว ใช้เวลา 2 นาที 42 วินาที / ขนาดแฟ้ม 5,166 KB เช่น

$$=SumProduct((IDRange=ID)*(KeyRange=Key)*NumRange)$$
- สูตร SumProduct แบบใส่เครื่องหมายคูณ 1 ใช้เวลา 2 นาที 50 วินาที / ขนาดแฟ้ม 5,171 KB เช่น
$$=SumProduct((IDRange=ID)*1,(KeyRange=Key)*1,NumRange)$$

กรณีใช้สูตร VLookup VS Match

- VLookup แบบ Approaching Match ใช้เวลาไม่ถึง 1 วินาที / ขนาดแฟ้ม 2,979 KB แต่ถ้าหาค่าไม่พบ (Not Available) ใช้เวลา 3 วินาที เช่น

$$=VLookup(ID,DataRange,2)$$
- Match แบบ Approaching Match ใช้เวลาไม่ถึง 1 วินาที / ขนาดแฟ้ม 2,976 KB แต่ถ้าหาค่าไม่พบ (Not Available) จะแสดงตำแหน่งสุดท้าย ใช้เวลาไม่ถึง 1 วินาที เช่น

$$=Match(ID,IDRange)$$
- VLookup แบบ Exact Match ขนาดแฟ้ม 2,981 KB โดยจะใช้เวลานานขึ้นหากตำแหน่งค่าที่ใช้หาห่างจากบรรทัดแรกมากขึ้น แต่ถ้าหาค่าไม่พบ (Not Available) ใช้เวลา 7 วินาที เช่น
$$=VLookup(ID,DataRange,2,0)$$
- Match แบบ Exact Match ขนาดแฟ้ม 2,979 KB โดยจะใช้เวลานานขึ้นหากตำแหน่งค่าที่ใช้หาห่างจากบรรทัดแรกมากขึ้น เช่น
$$=Match(ID,IDRange,0)$$

นอกจากนี้ยังได้ทดสอบสูตรบวกเลขคูณหาร ซึ่ง Excel คำนวณเร็วมากใช้เวลาคำนวณไม่ถึงวินาที แต่พบว่าขนาดแฟ้มจะเล็กหรือใหญ่ขึ้นกับความยาวของสูตรที่ใช้ เช่น $=1+2+3+4+5$ มีขนาดแฟ้มใหญ่กว่า $=A2+5$ โดยที่ A2 มีสูตร $=1+2+3+4$ เพื่อคำนวณส่วนแรกแยกไว้ก่อน

เมื่อต้องเลือกระหว่างขนาดแฟ้มหรือความเร็วในการคำนวณ

ประเด็นเรื่องความอ้วนอืดของแฟ้มตามที่กล่าวมาทั้งหมดในบทความนี้จะไม่เป็นสาระสำคัญ หากคิดแต่เพียงว่า ถ้าแฟ้มใหญ่ขึ้น หรือคอมพิวเตอร์ช้าลง เราก็ซื้อคอมพิวเตอร์เครื่องใหม่หรือซื้อ RAM มาเพิ่มหน่วยความจำในเครื่องให้มากขึ้นก็แก้ปัญหาได้แล้ว แต่ถ้าคิดกันอย่างนี้ ปัญหาที่เกิดจากการใช้ Excel จะยิ่งมีความสลับซับซ้อนมากขึ้น โดยเฉพาะอย่างยิ่งเมื่อใช้ Excel 2007 ขึ้นไป โดยในซีทหนึ่งมีกว่า 10,000 column และกว่า 1 ล้าน row ซึ่งย่อมทำให้ผู้ใช้ Excel ที่ขาดความรู้พื้นฐาน และขาดวินัยในการสร้างงาน สร้างแฟ้มงานที่จะย้อนกลับมาทำร้ายตัวเองในที่สุด

แต่ถ้าต้องเลือกระหว่างขนาดแฟ้มหรือความเร็วในการคำนวณ ในความคิดเห็นส่วนตัวขอเลือกความเร็วในการคำนวณว่ามีความสำคัญมากกว่าเรื่องของขนาด เพราะปัญหาเรื่องขนาดแฟ้มเป็นปัญหาที่แก้ไขได้ไม่ยาก แคหาทางแยกตารางที่ไม่จำเป็นต้องถูกเรียกใช้งานพร้อมกันออกไปเก็บเป็นแฟ้มต่างหากก็ทำให้แฟ้มเล็กลงเรียบร้อยแล้ว และแม้จะแยกตารางออกเป็นแฟ้มเล็กๆ ไม่ได้ เรายังสามารถจัดเก็บแฟ้มลงไปบนสื่อได้หลายชนิด ไม่ว่าจะเป็น USB Drive, CD, DVD, หรือฮาร์ดดิสก์ ซึ่งปัจจุบันมีความจุมากขึ้นกว่าแต่ก่อนมากในขณะที่มีราคาลดลง

ส่วนประเด็นในเรื่องความเร็วของการคำนวณนั้นแม้ถือว่าสำคัญกว่าเรื่องของขนาดแฟ้ม แต่ความเร็วเป็นเรื่องของความรู้สึก เป็นเรื่องยากที่จะหาเกณฑ์มาวัดว่าอย่างนี้เร็วหรือช้า มักใช้ความรู้สึกของคนที่ต้องนั่งรอคอยคำตอบเป็นตัวชี้วัด ถ้ายังรอกันได้ ก็ยังไม่เดือดร้อนหาทางอื่นที่จะทำให้แฟ้มคำนวณเร็วขึ้น ดังนั้นประเด็นเรื่องความเร็วในการคำนวณ จึงต้องถือเป็นภาระหน้าที่ของผู้สร้างงาน ต้องคอยติดตามเสมอว่าสูตรหรือวิธีที่ตนเพิ่งสร้างใหม่ลงไปตาราง ส่งผลให้แฟ้มงานนั้นคำนวณช้าลงไปกว่าเดิมอย่างผิดสังเกตหรือไม่ และต้องลองสร้างสูตรหลายๆแบบที่สามารถให้คำตอบเดียวกัน แต่ใช้เวลาคำนวณเร็วที่สุด ซึ่งควรพิจารณาผลในระยะยาวที่มีต่อตัวเองหรือผู้อื่นที่ร่วมใช้แฟ้มงานเดียวกันด้วยว่า ทุกคนสามารถใช้แฟ้มงานได้สะดวก เมื่อต้องการย้อนกลับมาปรับปรุงแก้ไข ต้องสามารถทำได้ง่ายในระยะเวลารวดเร็ว แม้จะทำให้แฟ้มมีขนาดใหญ่ขึ้น และคำนวณช้าลงก็ตาม

การออกแบบโครงสร้างตารางคำนวณ

ในโปรแกรม Microsoft Excel มีองค์ประกอบอยู่สองอย่างแทบไม่เปลี่ยนแปลงเลยในหลายยุคหลายสมัยที่ผ่านไป ไม่ว่าจะเป็น Excel รุ่นใดก็ตามยังคงมีสิ่งนี้ใช้กันเรื่อยมา สิ่งที่ว่านี้ก็คือ ตารางที่ประกอบด้วยเซลล์มีหลาย row หลาย column และสูตรสำเร็จรูปอีกหลายร้อยสูตรติดมาพร้อมกับตัวโปรแกรม ซึ่งผู้ใดสามารถนำตารางและสูตรมาใช้ร่วมกันได้เป็นอย่างดี จะทำให้ใช้โปรแกรม Excel ที่มีราคาไม่ก็หมื่นบาททำงานได้ยืดหยุ่นกว่าโปรแกรมสำเร็จรูปราคาแพงเป็นแสนเป็นล้านบาททีเดียว

บทความนี้จะแนะนำคุณให้หาทางใช้ตาราง Excel อย่างสร้างสรรค์ เพื่อใช้คำนวณหาผลลัพธ์ในปัญหาต่างๆ ได้สารพัด เพียงแค่ตารางและสูตรสำเร็จรูปของ Excel มาใช้ร่วมกันอย่างมีหลักการ อย่างเป็นขั้นเป็นตอน และอย่างมีเหตุผล จะช่วยให้คุณได้รับประโยชน์มากขึ้นกว่าแต่ก่อน เช่น

1. สามารถคำนวณหาคำตอบในปัญหาที่ไม่เคยคิดว่า Excel จะทำได้มาก่อน
2. สูตรสั้นลง และสามารถแก้ไขสูตรได้ง่ายและรวดเร็ว
3. สามารถตรวจสอบความถูกต้องของการคำนวณในทุกขั้นตอน
4. สามารถขยายขนาดหรือเคลื่อนย้ายตำแหน่งเซลล์ เพื่อนำตารางไปใช้ซ้ำในเซลล์อื่น ซื้ที่อื่นหรือแฟ้มอื่นได้ทันที
5. สามารถนำตารางไปใช้ในงานคำนวณอื่นที่ใช้หลักการเดียวกันได้ทันที หรือหลังจากผ่านการปรับปรุงเพียงเล็กน้อย
6. สามารถนำตารางไปใช้งานร่วมกับเมนูคำสั่งของ Excel ได้ทุกเมื่อ แม้แต่การนำข้อมูลไปแสดงผลต่อในรูปภาพ
7. แฟ้มคำนวณเร็วและมีขนาดแฟ้มเล็กลง แม้จะมีตารางขนาดใหญ่ก็ตาม

ตารางคำนวณคืออะไร

เพื่อให้เข้าใจตรงกัน จึงขอให้คำจำกัดความในความหมายของตารางคำนวณก่อนว่า ตารางคำนวณในบทความนี้มีได้หมายถึงพื้นที่ตารางว่างๆ ที่คุณจะได้เห็นทันทีบนจอเมื่อเปิดโปรแกรม Excel ขึ้นมาใช้งาน แต่ตารางคำนวณในที่นี้มีความหมายถึงตารางซึ่งคุณสร้างขึ้นมาเองโดยมีข้อมูลบันทึกไว้แล้ว อาจเป็นข้อมูลตัวเลข ตัวอักษร หรือสูตรใดๆ ก็ได้ ทั้งนี้มีเจตนาเพื่อใช้พื้นที่

เซลล์หรือตารางนั้นในการคำนวณหาค่าตอบที่คุณต้องการโดยเฉพาะ เช่น ตารางคำนวณค่าแรงที่ต้องจ่ายให้ลูกจ้างทำงานในแต่ละกะ ตารางคำนวณยอดต้นทุนขายแบบ First-in First-out ตารางคำนวณหาจำนวนสินค้าที่ต้องวางแผนสั่งผลิต (Material Requirements Planning) หรือตารางคำนวณเพื่อวางแผนการลงทุน (Feasibility Study) เป็นต้น

ตารางที่ไม่ถือว่าเป็นตารางคำนวณในบทความนี้ ได้แก่ ตารางฐานข้อมูลซึ่งใช้บันทึกข้อมูลดิบ (ไม่มีสูตร) ตารางรายงานที่จัดโครงสร้างเพื่อพิมพ์ในกระดาษหรือแสดงผลให้เห็นบนจอ หรือตารางที่ Excel จัดโครงสร้างให้เองอัตโนมัติจากการใช้คำสั่งบนเมนู SubTotals หรือ Pivot Table

ตารางคำนวณโดยทั่วไปประกอบด้วยพื้นที่ตารางย่อย 2 ส่วน ส่วนแรกเป็นพื้นที่ตารางสำหรับเก็บค่าตัวแปรที่ใช้ในการคำนวณ และพื้นที่ตารางอีกส่วนหนึ่งใช้สำหรับสร้างสูตรคำนวณซึ่งรับค่ามาจากค่าตัวแปรในพื้นที่ส่วนแรกนั่นเอง โดยผู้ใช้งานจะใช้พื้นที่ส่วนแรกในการพิมพ์ค่าตัวแปรตัวใหม่ลงไป จากนั้นจะแสดงคำตอบจากการคำนวณให้เห็นในตารางส่วนที่เป็นสูตรคำนวณ

ข้อควรพิจารณาก่อนเริ่มสร้างตารางคำนวณ

สมมติว่าคุณกำลังหาทางคำนวณค่าแรงที่ต้องจ่ายให้ลูกจ้างทำงานในแต่ละกะ จะต้องคิดพิจารณาอะไรบ้าง

1. กำหนดขอบเขตของความต้องการของคุณที่เป็นไปได้ว่ามีอะไรบ้าง เช่น ต้องการใช้ตารางในชีทหนึ่งเพื่อคำนวณค่าแรงของพนักงานคนหนึ่งซึ่งทำงานในวันหนึ่ง หรือต้องการใช้ตารางในชีทหนึ่งเพื่อคำนวณค่าแรงของพนักงานคนหนึ่งซึ่งทำงานในเดือนหนึ่งโดยแสดงเป็นรายวันต่อเนื่องกันไป หรือต้องการคำนวณค่าแรงของพนักงานทุกคนในวันหนึ่งๆ หรือต้องการคำนวณค่าแรงให้กับพนักงานทุกคนในเดือนหนึ่งโดยแสดงเป็นรายวันต่อเนื่องกันไป หรือมีความต้องการอื่นๆอีกบ้างไหม ทั้งนี้เพื่อหาทางสร้างตารางคำนวณเพียงตารางเดียวแต่ต้องมีความยืดหยุ่นสามารถนำไปใช้คำนวณได้ทุกกรณีตามที่คุณต้องการ(ทั้งที่เคยต้องการในอดีตและอาจจะมามากขึ้นในอนาคต)
2. ค้นหาเงื่อนไขทั้งหมดที่เกี่ยวข้องในการคำนวณ ซึ่งโดยทั่วไปผู้ที่คลุกคลีกับการบริหารการจ้างงานย่อมต้องมีเงื่อนไขที่ตนนำไปใช้ในงานอยู่แล้ว เช่น เงื่อนไขเกี่ยวกับช่วงเวลาหรือกะที่กำหนดให้ทำงานในแต่ละวันแบ่งเป็นกี่ช่วง แต่ละช่วงใช้เวลาทำงานมากน้อยเพียงไร แบ่งเป็นช่วงเวลาหยุดพักกี่ครั้ง แต่ละครั้งให้หยุดพักได้นานเท่าใด ลูกจ้างต้องตอกบัตรหรือบันทึกการเข้าทำงานไว้อย่างไร บริษัทมีเงื่อนไขในการจ่ายค่าจ้างแรงงานเป็นอย่างไร ซึ่งเงื่อนไขเหล่านี้จะต้องหาทางนำมาใช้ในการคำนวณ

3. ในอดีตจนถึงปัจจุบันใช้วิธีการคำนวณค่าแรงอย่างไร มีขั้นตอนการคำนวณที่ชัดเจนเป็นมาตรฐานที่ใช้กันได้ตลอดมาหรือไม่ หรือมีกรณีพิเศษใดบ้างที่ทำให้ต้องคำนวณวิธีอื่นต่างจากที่เป็นมาตรฐานนั้น ให้เก็บรวบรวมตัวเลขที่ใช้ในแต่ละช่วงแต่ละกรณีเอาไว้เพื่อนำมาใช้เป็นตัวอย่างสร้างตารางคำนวณใน Excel และเพื่อใช้เปรียบเทียบผลคำนวณที่เกิดขึ้นว่าตรงกันหรือไม่
4. ค้นหาความเป็นไปได้ทุกทางที่ลูกจ้างคนหนึ่งจะทำงานในช่วงเวลาที่กำหนด ซึ่งในกรณีปัญหาการเข้างานนี้พบว่ามีโอกาสความเป็นไปได้ที่จะมีช่วงเวลาทำงานซึ่งสัมพันธ์กับเวลาที่กำหนดหรือกะหนึ่งๆถึง 7 กรณี เช่น ถ้าสมมติว่ากำหนดให้กะทำงานจาก 8:00 ถึง 10:00 น. ลูกจ้างสามารถมีช่วงทำงานตรงกับกะ(8:00-10:00) อยู่ภายในกะ(8:30-9:00) คร่อมกะทั้งก่อนและหลัง(7:00-12:00) คร่อมกะเพียงช่วงเริ่มงาน(7:00-9:00) คร่อมกะเพียงช่วงเลิกงาน(9:00-12:00) นอกกะช่วงก่อนกะ(6:00-7:00) และนอกกะช่วงหลังกะ(11:00-12:00) ซึ่งคุณจะต้องหาทางคิดสูตรเพียงสูตรเดียวให้สามารถคำนวณได้ครบทั้ง 7 กรณี
5. ขอให้ยึดหลักว่า อย่าพยายามคิดหาสูตรลัดเพื่อคำนวณหาคำตอบให้แล้วเสร็จโดยใช้เซลล์สูตรเพียงเซลล์เดียว เพราะนอกจากสูตรจะต้องยาวเหยียดซับซ้อนสูตรกันหลายชั้นจนแกะแทบไม่ออกแล้ว ยังอาจเป็นไปได้เลยที่จะคิดหาสูตรขึ้นมาใช้ได้ และคุณอาจป่วยเป็นโรคกระเพาะเพราะมัวแต่คิดสูตรอยู่ทั้งวันทั้งคืน
6. แทนที่จะคิดสูตรลัดสูตรเดียวเซลล์เดียว ขอให้แบ่งการคำนวณออกเป็นขั้นๆ แล้วหาทางใช้สูตรสำเร็จรูปที่ Excel มีอยู่ให้ได้ก่อน แต่หากไม่มีสูตรสำเร็จรูปที่หาคำตอบได้โดยตรง คุณต้องหาทางนำสูตรสำเร็จรูปที่มีมาใช้ร่วมกันโดยอาจนำมาซ้อนสูตรกันในเซลล์เดียว หรือใช้เซลล์แยกคำนวณทีละขั้น แล้วจึงนำผลลัพธ์ที่ได้จากเซลล์หนึ่งส่งไปคำนวณต่อในสูตรของอีกเซลล์หนึ่ง กลายเป็นที่มาของตารางคำนวณนั่นเอง
7. ลองค้นหาวิธีการคำนวณจากอินเทอร์เน็ต เช่น ค้นหาจาก Google เพื่อเรียนรู้จากตัวอย่างของคนอื่น คุณอาจโชคดีได้ตัวอย่างที่ตรงตามที่คุณต้องการ สามารถนำไปใช้ได้เลย หรือแค่นำไปดัดแปลงเพียงเล็กน้อยก่อนก็ใช้กับงานของคุณได้แล้ว ซึ่งแม้ว่าจะได้ตัวอย่างไม่ตรงกับสิ่งที่ต้องการก็ตาม แต่ตัวอย่างเหล่านั้นน่าจะเป็นแนวทางให้คุณได้คิดพิจารณาและเกิดมุมมองที่กว้างขวางและชัดเจนกว่าเดิม

ขั้นตอนการสร้างตารางคำนวณ

หลังจากที่ได้พิจารณาขอบเขตความต้องการและพอได้เห็นแนวทางสร้างงานที่คนอื่นใช้กันมาก่อนแล้วบ้าง คราวนี้ก็ถึงประเด็นสำคัญว่าพอเปิดโปรแกรม Excel ขึ้นมาแล้ว คุณจะเริ่มต้นออกแบบตารางกันอย่างไรดี ซึ่งลักษณะของตารางคำนวณนี้เองที่จะชี้ให้เห็นฝีมือว่าใครเยี่ยมยุทธ์กว่ากัน ถ้าเทียบกับการสร้างบ้านสักหลังหนึ่งแล้ว ข้อพิจารณาที่ผ่านไปเป็นเพียงแค

การออกแบบคร่าวๆและเตรียมซื้ออุปกรณ์ก่อสร้าง หิน ทราย ปูนซีเมนต์ เตรียมไว้ไม่ให้ขาด ไม่ให้เกินกว่าความจำเป็น ชั้นจากนี้เป็นหน้าที่ของสถาปนิกและวิศวกรที่จะต้องลงมือก่อสร้าง บ้านของจริงให้สวยหรู ดูดี อยู่สบาย อากาศถ่ายเทสะดวก และสามารถตกแต่งต่อเติมให้น่าอยู่ ไปชั่วลูกชั่วหลาน

ก่อนอื่นผมจำเป็นต้องขอเตือนให้ทราบข้อผิดพลาดที่ผู้ใช้ Excel ทั่วไปมักคิดออกแบบตาราง อย่างรวบรัดเกินไป โดยการพยายามนำตารางคำนวณไปรวมกับตารางที่ใช้พิมพ์ออกไปเป็น รายงานหรือแสดงผลลัพธ์ที่ต้องการบนหน้าจอคอมพิวเตอร์ไว้เป็นตารางเดียวกัน ซึ่งความคิด เช่นนี้จะกลายเป็นอุปสรรคสำคัญทำให้ไม่สามารถใช้เซลล์ต่อเนื่องกันเพื่อจัดการคำนวณที่ละ ลำดับ จึงขอแนะนำให้คิดแยกตารางคำนวณออกเป็นพื้นที่ต่างหากแยกจากตารางที่ใช้พิมพ์ รายงาน แม้ต้องทำให้เสียพื้นที่เซลล์มากขึ้นโดยอาจต้องแยกออกเป็นชีทหลายๆชีท แต่จะช่วยให้สามารถออกแบบโครงสร้างตารางคำนวณได้อย่างอิสระ แล้วต่อเมื่อตารางคำนวณนี้ สามารถคำนวณหาคำตอบได้ตามที่ต้องการแล้ว จึงค่อยนำผลลัพธ์ที่ได้ link ไปประกอบกันเป็น ตารางรายงานในที่สุด

โครงสร้างพื้นฐานของตารางคำนวณ ประกอบด้วยส่วนที่เป็นพื้นที่ตารางสำหรับรับค่าตัวแปร และพื้นที่ตารางสำหรับสร้างสูตรคำนวณ โดยให้ยึดหลักว่าสูตรที่สร้างขึ้นในตารางคำนวณนั้น หากเป็นไปได้ขอให้คิดสร้างสูตรลงไปเซลล์แรกเซลล์เดียวที่หัวมุมซ้ายบนสุดของตาราง จากนั้นเมื่อ copy สูตรนี้ไปใช้ที่เซลล์ส่วนอื่นในตารางต้องสามารถใช้สูตรเดิมนั้นได้โดยไม่ จำเป็นต้องแก้ไขสูตรให้แตกต่างกันไปจากเดิมแต่อย่างใด หรืออีกนัยหนึ่งให้ใช้สูตรเดียวกับทุก เซลล์ในตารางคำนวณให้ได้ แต่ถ้าไม่สามารถสร้างสูตรเดียวที่เซลล์หัวมุม ก็ขอให้พยายาม สร้างสูตรเดียวในแต่ละแนว row หรือ column

เพื่อให้ให้เห็นโครงสร้างตารางคำนวณแบบง่ายๆ ขอยกตัวอย่างการสร้างตารางคำนวณสูตรคูณ มาพิจารณากันตามรูปต่อไปนี้

	C3		fx		=C\$2*\$B3	
	A	B	C	D	E	F
1						
2			2	3	4	5
3		1	2	3	4	5
4		2	4	6	8	10
5		3	6	9	12	15
6		4	8	12	16	20
7		5	10	15	20	25
8						
9						

รูปตารางนี้เป็นตัวอย่างการคำนวณสูตรคูณแม่ 2 ถึง 5 โดยกำหนดให้นำไปคูณเลข 1 ถึง 5

- ให้แยกเซลล์รับตัวแปรวางไว้ตามแนวคู่ขนานกับพื้นที่คำนวณโดยอาจวางไว้เป็นแนวหัวตารางด้านบน (C2:F2) หรือหัวตารางด้านข้าง (B3:B7) ซึ่งตัวแปรที่ว่านี้เป็นเซลล์รับตัวเลขที่คุณอาจมีความต้องการเปลี่ยนแปลงเป็นค่าอื่นในอนาคต
- C3:F7 เป็นพื้นที่เซลล์ที่คุณต้องหาทางสร้างสูตรลงไปทุกเซลล์ โดยให้สร้างสูตรลงไปทีเซลล์ C3 ซึ่งเป็นตำแหน่งเซลล์หัวมุมซ้ายบนเพียงเซลล์เดียว ให้มีสูตร $=C\$2*\$B3$ จากนั้นให้ copy ไป paste ทุกเซลล์ในพื้นที่ C3:F7

โปรดสังเกตว่าตัวอย่างสูตรคูณนี้เป็นตัวอย่างการนำค่าไปคำนวณแบบ 1 ต่อ 1 หมายถึง เซลล์สูตรหนึ่งๆจะรับค่าจากเซลล์ตัวแปรตามแนวนอนหรือแนวตั้งเพียงข้างละ 1 เซลล์เท่านั้น ซึ่งในปัญหาอื่นอาจจำเป็นต้องใช้การคำนวณที่ซับซ้อนมากขึ้น โดยเซลล์สูตรหนึ่ง จะใช้เซลล์ตัวแปรมากกว่าเซลล์เดียวก็เป็นได้โดยใช้สูตรพวก Lookup ช่วยในการเลือกข้อมูลมาใช้คำนวณ

นอกจากเซลล์สูตรจะรับค่ามาจากเซลล์ตัวแปรแล้ว เซลล์สูตรยังอาจรับค่าต่อมาเซลล์สูตรคำนวณในขั้นก่อนได้อีก ซึ่งจะเห็นได้ชัดจากตารางคำนวณงบการเงินหรือวางแผนการผลิตที่มียอดคงเหลือปลายงวดส่งต่อไปตั้งเป็นยอดคงเหลือต้นงวดของช่วงเวลาถัดไป

ตารางคำนวณที่ดีต้องมีความยืดหยุ่นสามารถคำนวณให้ผลลัพธ์ได้ตามต้องการตลอดไปไม่ว่าจะมีการโยกย้ายหรือแยกตารางตัวแปรไปวางไว้ที่ชีทอื่นเหนือแฟ้มอื่น ดังนั้นพอสร้างตารางคำนวณเสร็จ ขอให้ทดลอง cut ตารางไปวางไว้ที่ตำแหน่งอื่นหรือโยกย้ายเซลล์ตัวแปรแยกออกจากกัน เพื่อพิสูจน์ว่าตารางคำนวณยังคงสามารถให้ผลลัพธ์ถูกต้องตามเดิมหรือไม่

ตัวอย่างตารางคำนวณหาระยะเวลาที่ใช้ในกะ

สมมติว่าลูกจ้างคนหนึ่งเข้างานตั้งแต่เวลา 7:00-9:00 น. ให้สร้างตารางคำนวณหาระยะเวลาที่เขาทำงานในกะที่กำหนดซึ่งเริ่มตั้งแต่เวลา 8:00-10:00 น.

จากคำถามข้างต้นนี้ คุณคงคิดคำตอบในใจได้ว่าต้องทำงานในกะเป็นเวลา 1 ชั่วโมงใช่ไหม เพราะลูกจ้างคนนี้ออกจากงานเวลา 9:00 น. จึงนับระยะเวลาดังแต่เริ่มกะตอน 8:00 น. ไปจนถึงเวลาที่เขาออกจากงาน ได้เป็นระยะเวลา 1 ชั่วโมง (8:00-9:00) ซึ่งคงหาทางสร้างสูตรคำนวณได้ไม่ยาก แต่ถ้าเขามีช่วงที่เข้าทำงานหลากหลายกรณีจะใช้สูตรอย่างไร

1. ตรงกับกะ (8:00-10:00)
2. อยู่ภายในกะ (8:30-9:00)

3. คร่อมกะ ทั้งก่อนและหลัง (7:00-12:00)
4. คร่อมกะ เพียงช่วงเริ่มงาน (7:00-9:00)
5. คร่อมกะเพียงช่วงเลิกงาน (9:00-12:00)
6. นอกกะ ช่วงก่อนกะ (6:00-7:00)
7. นอกกะ ช่วงหลังกะ (11:00-12:00)

H4		=MAX(0, MIN(F4,D4) - MAX(E4,C4)							
	A	B	C	D	E	F	G	H	I
1									
2		เวลาเข้างาน			ช่วงเวลาของกะ		เวลาที่ใช้ในกะ		
3		Job	Start	Stop	From	To	คำตอบ		
4		1	08:00	10:00	08:00	10:00	2:00		
5		2	08:30	09:00	08:00	10:00	0:30		
6		3	07:00	12:00	08:00	10:00	2:00		
7		4	07:00	09:00	08:00	10:00	1:00		
8		5	09:00	12:00	08:00	10:00	1:00		
9		6	06:00	07:00	08:00	10:00	0:00		
10		7	11:00	12:00	08:00	10:00	0:00		
11									

สูตรคำนวณหาระยะเวลาทำงานที่ใช้ในกะ ในเซลล์ H4 =

MAX(0, MIN(F4,D4) - MAX(E4,C4)) โดยแยกอธิบายแต่ละส่วนในสูตรได้ ดังนี้

- **MIN(F4,D4)** เป็นการนำเวลาปลายกะมาเทียบกับเวลาปลาย job (เวลาออกจากงาน) เพื่อหาเวลาสุดท้ายที่เป็นไปได้ที่เกิดขึ้นก่อน
- **MAX(E4,C4)** เป็นการนำเวลาต้นกะมาเทียบกับเวลาต้น job (เวลาเริ่มเข้างาน) เพื่อหาเวลาสุดท้ายที่เป็นไปได้ที่เกิดขึ้นหลังสุด
- **MIN(F4,D4)-MAX(E4,C4)** คำนวณหาระยะเวลาทำงานที่ใช้ในกะ
- **MAX(0,xxx)** ปรับผลลัพธ์ที่คำนวณได้ให้ไม่มีทางต่ำกว่า 0

สูตร = MAX(0,MIN(F4,D4)-MAX(E4,C4)) นี้ เป็นสูตรสั้นๆที่สามารถใช้แทนสูตร IF เพราะหากจะใช้ IF ในการคำนวณก็ต้องนำ IF มาซ้อนกันถึง 7 ชั้น

ตารางคำนวณต้องมาก่อน ส่วนสูตรถัดมาทีหลัง

ทุกวันนี้คุณสามารถค้นหาสูตรคำนวณเพื่อหาคำตอบในปัญหาต่างๆได้โดยตรงจากอินเทอร์เน็ต หรือยกคำถามไปหาหรือในฟอรัมถามตอบปัญหา Excel ได้ไม่ยาก หลายๆครั้งที่เดียวที่จะได้รับคำตอบเป็นสูตรซ้อนกันยาวเหยียดเป็นหน้า จนอดทึ่งในความสามารถของคนที่สร้างสูตรเหล่านั้นไม่ได้ว่าเขาคิดสูตรกันมาได้อย่างไร

หากจะไล่เรียงลำดับความเก่งเกี่ยวกับการใช้ Excel แล้ว การที่เก่งมีความสามารถสร้างสูตร ยากๆยาวๆได้ ถือว่าเป็นความเก่งขั้นสูงที่มักเป็นความสามารถเฉพาะตัว ยากที่คนอื่นจะเลียนแบบหรือเข้าใจที่ไปที่มาของสูตรที่นำมาซ้อนกัน เพราะคนเก่งแต่ละคนมักเลือกใช้สูตรที่ตนถนัดแตกต่างกันไป แต่อย่างไรก็ตาม ในเรื่องสูตรที่ตัวผมเองคิดขึ้นนั้น หลายๆครั้งที่เดียวที่พบว่าไม่สามารถแกะสูตรยาวๆที่ตัวเองเป็นคนต้นคิด จึงเป็นสิ่งที่น่าเป็นห่วงอย่างยิ่งว่า ในเมื่อแม้แต่ตัวผู้คิดสูตรขึ้นเองยังไม่สามารถเข้าใจที่ไปที่มาของแต่ละส่วนในสูตรยาวๆได้เลย แล้วใครคนอื่นจะเข้าใจสูตรที่ตนไม่ได้สร้างเองหรือสามารถนำสูตรไปดัดแปลงใช้งานได้เล่า ดังนั้น การสร้างสูตรยากๆยาวๆให้แก่ผู้อื่นใช้ จึงเป็นการสร้างปัญหาระยะยาวให้เกิดขึ้นกับผู้ใช้สูตรที่ตนเองไม่ได้เป็นคนสร้าง เพราะว่าเมื่อใดที่เงื่อนไขในการคำนวณแตกต่างไปจากเดิม เมื่อนั้นก็จำต้องย้อนกลับมาตั้งคำถามให้คนอื่นช่วยสร้างสูตรใหม่ให้อีก

แทนที่จะใช้สูตรลัดซึ่งเป็นสูตรซ้อนกันยาวเหยียด แนะนำให้กระจายการคำนวณออกเป็นหลายเซลล์เพื่อแยกคำนวณแต่ละอย่างก่อน แล้วจึงนำผลลัพธ์ที่ได้ไปคำนวณหาผลลัพธ์สุดท้าย จะเป็นวิธีช่วยสร้างความเข้าใจให้เกิดขึ้นถึงที่ไปที่มาของสูตรว่าคิดกันมาได้อย่างไร

ยกตัวอย่างสูตรคำนวณหาชื่อแฟ้ม ซึ่งดูแล้วยากที่จะจำ และไม่มีทางรู้ว่ามีหลักการคิดมาได้
อย่างไร

```
=MID(CELL("filename"),FIND("[",CELL("filename"))+1,FIND("]",CELL("filename"))-FIND("[",CELL("filename"))-1)
```

สูตรนี้มีที่มาจากสูตร

```
=MID(ข้อความที่มีชื่อแฟ้ม, เลขตำแหน่งเครื่องหมาย [, เลขจำนวนตัวอักษรของชื่อแฟ้ม)
```

แยกคำนวณแต่ละส่วนได้ตามรูปต่อไปนี้

B7		=MID(CELL("filename"), FIND("[",CELL("filename"))+1, FIND("]",CELL("filename"))-FIND("[",CELL("filename"))-1)	
	A	B	C
1	D:\Dummy\[CalculationTable.xlsx]Sheet9		A1 : '=CELL("filename")
2			
3	ตำแหน่งเครื่องหมาย [10	B3 : =FIND("[",A\$1)
4	ตำแหน่งเครื่องหมาย]	32	B4 : =FIND("]",A\$1)
5	ชื่อแฟ้ม	CalculationTable.xlsx	B5 : '=MID(A1,B3+1,B4-B3-1)
6			
7	ชื่อแฟ้ม	CalculationTable.xlsx	
8			
9			

แทนที่จะใช้สูตรยาวจนจำแทบไม่ไหวในเซลล์ B7 ให้เริ่มจากการใช้สูตรหาค่าที่เกี่ยวข้องกับการแยกค่าทีละชั้น ดังนี้

1. เซลล์ A1 ใช้สูตร =CELL("filename") เพื่อหาชื่อแฟ้ม โดยคำตอบที่ได้จะแสดงทั้งชื่อ drive ชื่อโฟลเดอร์ ชื่อแฟ้ม และชื่อชีท
2. เซลล์ B3 และ B4 ใช้สูตร =FIND("[",A\$1) และ =FIND("]",A\$1) เพื่อคำนวณหาตำแหน่งของเครื่องหมายวงเล็บ [] ตามลำดับ
3. เซลล์ B5 ใช้สูตร =MID(A1,B3+1,B4-B3-1) เพื่อแยกเฉพาะส่วนที่เป็นชื่อแฟ้มออกมาใช้งาน

การออกแบบโครงสร้างตารางคำนวณร่วมกับสูตรลด

จากสูตร = MAX(0,MIN(F4,D4)-MAX(E4,C4)) ซึ่งใช้คำนวณหาระยะเวลาที่ใช้ในกะซึ่งได้อธิบายที่ไปที่มาของแต่ละส่วนในสูตรแล้วว่าเป็นอย่างไร จากนั้นขอให้คุณคิดต่อไปอีกว่าจะนำสูตรนี้ไปใช้ในการวางแผนโดยจำเป็นต้องนำสูตรนี้ไปใช้ร่วมกับตัวแปรอื่นๆ เช่น ช่วงเวลาของกะอื่นๆ ช่วงเวลาทำงานในวันอื่นๆหรือเวลาทำงานของลูกจ้างคนอื่น ตลอดจนหาทางคำนวณค่าแรงที่ต้องจ่ายออกเป็นตัวเงินนั้น ต้องออกแบบตารางคำนวณให้มีหน้าตาอย่างไรดี

E6	=MAX(0, MIN(E\$3, \$D6) - MAX(E\$2, \$C6))																		
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	กำหนดเวลาแต่ละช่วงกะ																		
2					From	06:00	08:00	10:00	12:00	13:00	17:00	19:00	22:00						
3					To	08:00	10:00	12:00	13:00	17:00	19:00	22:00	24:00						
4	ตารางคำนวณหาระยะเวลาที่ใช้ในแต่ละช่วงกะทำงาน													อัตราค่าแรง					
5	Job	Start	Stop	06:00	08:00	10:00	12:00	13:00	17:00	19:00	22:00	Normal	OT	1	1.5				
6	1	07:00	08:00	01:00									01:00	0.00	1.50				
7	2	08:00	12:00		02:00	02:00						04:00		4.00	0.00				
8	3	12:00	13:00				01:00						01:00	0.00	1.50				
9	4	13:00	16:00					03:00				03:00		3.00	0.00				
10	5	16:00	20:00					01:00	02:00	01:00		03:00	01:00	3.00	1.50				
11																			

ตัวอย่างนี้ขอสมมติว่าในกำหนดเวลาทำงานแต่ละวัน ตั้งแต่ 6:00-24:00 น.นั้น ถูกแบ่งออกเป็น 8 กะ แต่ละกะเริ่มจากช่วงเวลา 6:00, 8:00, 10:00, 12:00, 13:00, 17:00, 19:00, และ 22:00 โดยกำหนดค่าแรงในช่วงกะที่เริ่ม 8:00, 10:00, 13:00, และ 17:00 ให้ใช้อัตราค่าแรงปกติ ส่วนกะอื่นๆมีอัตราค่าแรง OT 1.5 เท่าของอัตราปกติ

แรกสุดผมตัดสินใจว่า จะยังไม่นำประเด็นเรื่องค่าแรงมาใช้ในการคำนวณ โดยต้องการคำนวณให้ได้ระยะเวลาที่ใช้ในแต่ละกะให้ได้ก่อน เพื่อเผื่อว่าจะสามารถใช้ตัวเลขระยะเวลาไปใช้ในการวางแผนอื่นๆที่อาจเกี่ยวข้องได้อีก เช่น หากต้องการวัดประสิทธิภาพในการทำงาน ก็อาจนำระยะเวลาที่ใช้ในแต่ละกะไปคูณกับอัตราค่าจ้างผลิตมาตรฐาน แล้วนำมาเปรียบเทียบกับผลงานที่ผลิตได้จริงในแต่ละกะว่าดีกว่าหรือต่ำกว่ามาตรฐานอย่างไร

ในการออกแบบตารางคำนวณให้เลือกวางแนวตารางที่อาจจะมีรายการเพิ่มให้เพิ่มได้ในแนวนอน จะสะดวกกว่าการวางตารางที่เพิ่มไว้ตามแนวดิ่ง ดังนั้นจึงกำหนดให้แบ่งหัวตารางที่เป็นกะต่างๆไว้ตั้งแต่เซลล์ E5:L5 เพราะเชื่อว่าส่วนนี้คงไม่มีการปรับเปลี่ยนจากเดิมมากนัก แล้วกำหนดให้ใช้ตารางด้านซ้ายสุดเรียงลำดับงานแต่ละงานไปเรื่อยๆ จากเซลล์ B6:D10

ตัวเลขเวลาเริ่มกะที่บันทึกไว้ในเซลล์ E5:L5 นั้น ยังถูกนำไปกระจายแยกเป็นเวลาเริ่มกะ (From) และเวลาสิ้นสุดของกะ (To) ไว้ที่เซลล์ E2:L3 เพื่อแสดงให้เห็นชัดเจนขึ้นว่าแต่ละกะมีเวลาเริ่มต้นและสิ้นสุดอย่างไร และยังเพื่อไว้ใช้กับสูตรคำนวณหาระยะเวลาที่ใช้ในกะอีกด้วย จะได้ทำให้สูตรมีการใช้ตำแหน่งอ้างอิงในแนวเดียวกันไปตลอด ช่วยให้สร้างและกะที่ไปที่มาได้ง่ายขึ้น

ส่วนด้านของรายละเอียดการทำงานนั้น ในช่วงแรกนี้ใช้เลขที่งานเรียงไปเรื่อยๆไว้ก่อนในเซลล์ B6:B10 ซึ่งในอนาคตอาจเปลี่ยนจากเลขที่งาน ไปเป็นชื่อคน หรือเลขเครื่องจักรแทนก็ได้ ทั้งนี้เพื่อใช้แสดงชื่อรายการเท่านั้นว่าเป็นการทำงานของใคร และสามารถพิมพ์รายการต่อไปได้อีก นับหมื่นรายการตามแนวนอน

กำหนดให้บันทึกเวลาเริ่มงาน (Start) และออกจากงาน (Stop) คู่ขนานกับเลขที่ Job ไว้ในเซลล์ตั้งแต่ C6:C10 และ D6:D10 ตามลำดับ

ตารางตั้งแต่เซลล์ E6:L10 ใช้สูตร $=\text{MAX}(0, \text{MIN}(E\$3, \$D6) - \text{MAX}(E\$2, \$C6))$ เพื่อคำนวณหาระยะเวลาที่ทำงานในแต่ละกะ โดยสร้างสูตรนี้ลงไปในเซลล์ E6 เซลล์เดียวแล้ว copy ไป paste เพื่อใช้คำนวณทุกเซลล์ตั้งแต่ E6:L10

เพื่อแสดงให้เห็นเฉพาะตัวเลขเวลาที่คำนวณได้ ไม่ให้แสดงตัวเลขเวลาที่เท่ากับ 0 จึงเลือกใช้ Format แบบ hh:mm;; และใช้สีพื้นแยกแต่ละกะที่ใช้อัตราค่าแรงปกติออกจากพื้นตารางส่วนที่ใช้อัตราค่าแรงพิเศษ (OT)

จากนั้นสร้างสูตรรวมหายอดระยะเวลาที่ทำงานในช่วงอัตราปกติและช่วงอัตรา OT ไว้ในตารางด้านขวาดังแต่เซลล์ N6:O10 ซึ่งเมื่อนำไปคูณกับเลขอัตราในเซลล์ Q4 และ R4 แล้ว ทำให้คำนวณหาค่าแรงแยกประเภทของอัตราแต่ละตัวแสดงให้เห็นในตาราง Q6:R10

วิเคราะห์ลักษณะโครงสร้างตารางคำนวณที่ใช้

1. พื้นที่ตารางที่ใช้รับตัวแปร เช่น B6:B10, C6:C10, D6:D10, และ E5:L5 ถูกวางไว้เป็นแนวคู่ขนานกับแนวตารางสูตรคำนวณ E6:L10
2. สูตรคำนวณในพื้นที่ตาราง E6:L10 เกิดจากสูตรในเซลล์ E6 เพียงเซลล์เดียวที่นำไป paste ต่อให้เซลล์อื่นๆ ทำให้สะดวกในการแก้ไขสูตรไม่ต้องค้นหาและแก้ไขสูตรในเซลล์แต่ละเซลล์
3. ตารางกำหนดเวลาแต่ละช่วงกะ E2:L3 ช่วยชี้ชัดว่าแต่ละกะมีเวลาใด ช่วยให้สร้างสูตรคำนวณได้ง่ายขึ้นและยังทำให้มีความเข้าใจชัดเจนมากขึ้นสำหรับผู้ใช้งานทั่วไป ซึ่งในอนาคตหากไม่ต้องการแสดงพื้นที่ส่วนนี้ก็สามารถโยกย้ายไปซ่อนที่อื่น

4. จุดเด่นของโครงสร้างตารางซึ่งช่วยให้สามารถคำนวณค่าแรงในแต่ละอัตราได้ คือ การใช้ตารางแบ่งออกเป็นแต่ละกะเพื่อแยกคำนวณหาระยะเวลาที่ใช้ในกะนั้นๆก่อน และยังช่วยให้สามารถตรวจสอบความถูกต้องของการคำนวณได้ง่ายชัดเจนกว่าการคิดสูตรยากๆยาวๆเพียงเพื่อคำนวณแยกแต่ละกะให้แล้วเสร็จในเซลล์เดียว
5. ในแง่ของเลขตัวเงินค่าแรงที่โดยทั่วไปถือเป็นเรื่องปกปิด ได้ถูกแยกคำนวณในตาราง Q6:R10 ช่วยให้สามารถใช้ตารางส่วนนี้ในการบริหารค่าแรงได้โดยตรง และสามารถโยกย้ายตารางไปแยกเก็บไว้เป็นเรื่องลับในชีทอื่นหรือแฟ้มอื่น
6. เลขที่ Job ในตาราง B6:B10 นั้น ในอนาคตสามารถบันทึกรหัสหรือชื่อลงไปแทน จากนั้นจึงนำสูตร Vlookup หาเวลาทำงานมาแสดงโดยอัตโนมัติในตาราง C6:D10 ได้ด้วย หรืออาจใช้คำสั่ง Data > Table เพื่อส่งตัวแปรลงไปแทนเลขที่ Job เพื่อคำนวณสัปดาห์ยอดค่าแรงได้ทันทีโดยไม่ต้องย้อนกลับมาเพิ่มรายการใหม่
7. สามารถใช้กับการคำนวณหาค่าแรงได้หลายกรณี เช่น ค่าแรงของลูกจ้างคนหนึ่งในวันหนึ่ง หรือค่าแรงของลูกจ้างคนหนึ่งรายวัน หรือค่าแรงของลูกจ้างเรียงตัวไปทุกคนในแต่ละวัน ซึ่งคุณอาจใช้ชีทเดิมชีทเดียวคำนวณ หรือ copy เป็นชีทใหม่เพื่อคำนวณแยกตามรายบุคคล

หมายเหตุ :

1. การไล่ลำดับงานในตัวอย่างนี้เป็นการไล่จากบนมาล่างหรือจากซ้ายไปขวา ซึ่งในแง่การวางแผนกำหนดการณ์แล้ว เรียกว่า แบบ Push เพื่อหากำหนดเวลาทำงานเสร็จ
2. ถ้าไล่ลำดับงานจากล่างไปบนหรือจากขวามาซ้าย จะกลายเป็นแบบ Pull เพื่อหากำหนดเวลาที่ต้องเริ่มงาน
3. หากต้องการแทรกกระยะเวลาหยุดพักของแต่ละ Job ซึ่งย่อมมีผลทำให้กำหนดเวลาเริ่มต้นของ Job ถัดไปช้าลง สามารถปรับเปลี่ยนได้ง่ายๆโดยสร้างสูตรหา กำหนดเวลาเริ่มงานใน Job ถัดไปให้มีค่าเท่ากับ เวลาออกจากงานของ Job ก่อนบวกด้วยระยะเวลาที่หยุดพัก

การออกแบบโครงสร้างตารางคำนวณ ภาค 2

ตารางคำนวณในภาคแรกเกิดจากการใช้สูตรที่มีโครงสร้างเดียวกันซ้ำในทุกเซลล์ของตารางคำนวณ โดยกำหนดให้หาทางสร้างสูตรขึ้นเพียงสูตรเดียวในเซลล์แรกตรงหัวมุมซ้ายบนสุดของตาราง แล้วจัดการ Copy สูตรไปใช้ทั้งตาราง และจากการที่ใช้สูตรที่มีโครงสร้างเดียวกันไปตลอดนี้ย่อมแสดงว่าเป็นสูตรที่คำนวณหาคำตอบในเรื่องเดียวกัน จากนั้นเราจึงใช้โครงสร้างของตารางเป็นตัวช่วยในการเลือกใช้คำตอบจากตำแหน่งของเซลล์ที่เกี่ยวข้องกับคำตอบ ซึ่งตารางคำนวณแบบแรกนี้ไม่เหมาะกับการคำนวณที่มีความสลับซับซ้อนมากนัก

ในบทความภาคที่ 2 จะนำโครงสร้างตารางคำนวณที่มีความซับซ้อนมากขึ้นมาให้ศึกษากันโดยขอเรียกตารางคำนวณแบบนี้ว่า ตารางคำนวณแบบ Module ซึ่งสามารถนำไปใช้งานกันได้ 2 แบบ คือ แบบ Module เดี่ยว (Single Module) และแบบ Module รวม (Compound Module)

ความหมายของ Module

ผู้ที่เคยใช้ Excel สร้างตารางคำนวณงบการเงินมาก่อน ต้องทราบดีกว่าจะคำนวณหายอดกำไรขาดทุนมาได้ต้องคำนวณผ่านยอดรายรับหักต้นทุนขายเป็นกำไรขั้นต้น จากนั้นจึงนำยอดตัวเลขกำไรขั้นต้นไปหักด้วยค่าใช้จ่ายต่างๆและภาษีเพื่อเป็นยอดกำไรขาดทุน ซึ่งลักษณะของการคำนวณงบการเงินนี้แหละคือลักษณะของตารางคำนวณแบบ Module

ในงานด้านวางแผนการผลิตมีลักษณะการใช้ Excel แบบ Module เช่นกัน นับตั้งแต่รับคำสั่งซื้อแล้วต้องนำยอดไปเปรียบเทียบกับสินค้าคงคลังเพื่อคำนวณหายอดวัตถุดิบที่ต้องใช้ตามกำลังการผลิตที่มีอยู่ อีกทั้งต้องกำหนดเวลาให้กับตัวเลขการสั่งผลิตหรือจัดหาวัตถุดิบตามโครงสร้างผลิตภัณฑ์ให้ได้ดีกว่า ต้องการใช้ในปริมาณเท่าใดและต้องเริ่มสั่งของเมื่อใดเพื่อให้สามารถดำเนินการตามขั้นตอนการผลิตแล้วได้สินค้าสำเร็จรูปตามเวลาที่กำหนด

Module คือ ตารางที่ประกอบด้วยเซลล์สูตรคำนวณหลายๆเซลล์ซึ่งคำนวณต่อเนื่องกันไปทีละขั้น โดยผลการคำนวณที่ได้ในแต่ละขั้นไม่จำเป็นต้องมีความหมายใดเลยก็ได้ แต่เมื่อนำมาคำนวณร่วมกันหรือคำนวณต่อเนื่องกันแล้วต้องทำให้ได้ผลลัพธ์ตามต้องการ

จากคำจำกัดความของ Module ข้างต้น อาจทำให้คุณหลงภูมิใจว่าตนเองเคยสร้างตารางคำนวณแบบ Module นี้มานานแล้ว บทความนี้ไม่เห็นมีอะไรที่แปลกใหม่ต่างจากที่คุณรู้ได้อยู่

แล้ว จึงขอให้ลองนึกดูว่าหากคุณสามารถใช้ Excel อย่างคอมพิวเตอร์ได้สมบูรณ์จริง คุณต้องหาทางนำ Module ที่สร้างไว้นั้นมาใช้ซ้ำแล้วซ้ำอีกโดยไม่จำเป็นต้องสร้าง Module ขึ้นใหม่ และด้วยตารางคำนวณแบบ Module ที่คุณมีเพียง Module เดียว จะทำอย่างไรที่จะนำตารางนั้นมาใช้ซ้ำแล้วซ้ำอีก เพื่อหาทางใช้ Module เดียว ชีทเพียงชีทเดียว หรือแฟ้มเพียงแฟ้มเดียวกับงานคำนวณที่คุณคิดไม่ถึงว่าจะใช้ Excel ทำได้โดยไม่ต้องพึ่งการเขียนโปรแกรม Visual Basic มาช่วยแม้แต่น้อย

จุดประสงค์ของการสร้างตารางคำนวณแบบ Module

1. เพื่อลดความซับซ้อนของสูตรคำนวณ จากเดิมที่ต้องสร้างสูตรยาวๆลงไปเซลล์เพียงเซลล์เดียว เปลี่ยนไปเป็นการกระจายสูตรแยกให้ใช้เซลล์แต่ละเซลล์มีสูตรคำนวณที่ละชั้นส่งผลการคำนวณต่อเนื่องกันไป ทำให้สูตรสั้นลง และทำให้คุณเข้าใจลำดับการคำนวณได้ง่ายขึ้น
2. เพื่อให้คุณสามารถใช้ Excel กับการคำนวณงานที่มีหลายขั้นตอนหรือเกี่ยวข้องกับการทำงานที่มีลักษณะต่อเนื่องกัน
3. เพื่อช่วยให้คุณสามารถลดขนาดแฟ้มและจำนวนตารางที่มีอยู่

ลักษณะโครงสร้างตารางคำนวณแบบ Single Module

ตารางคำนวณแบบ Single Module มีลักษณะตรงกับความหมายของคำว่า Single นั่นคือเป็นตารางคำนวณแบบตารางเดียวหรือตารางเดี่ยว ที่สามารถคำนวณให้ผลลัพธ์ที่ต้องการได้ครบถ้วนภายในตารางคำนวณตารางเดียว

ผู้ใช้ Excel ทั่วไปมักใช้ Excel กันแบบผิดๆ จากความง่ายของการใช้ตารางของ Excel เช่น ถ้ารายงานที่ต้องการมีร้อยหน้า เราก็มักจะออกแบบตาราง Excel ให้มีความยาวเหยียดเลียนแบบตามหน้ากระดาษที่ต้องการพิมพ์ หรือเมื่อใดที่ต้องการรายงานตามรายเดือน ก็มักจะสร้างชีทหลายๆชีทแยกเป็นชีทละเดือนหรือสร้างตารางแบ่งเป็นแฟ้มคำนวณตามรายเดือน ซึ่งยังดีที่ในปีหนึ่งมีแค่ 12 เดือน จึงทำให้มีจำนวนชีทหรือแฟ้มเพียง 12 ชีทหรือ 12 แฟ้ม แต่ถ้าเป็นงานที่ต้องใช้ติดต่อกันไปหลายปีหรือแบ่งรายงานตามวัตถุดิบนับร้อยนับพันชนิด เห็นที่ย่อมทำให้มีจำนวนชีทหรือจำนวนแฟ้มที่มีการคำนวณแบบเดียวกันซ้ำมากจนจดจำกันไม่ไหว

ตารางคำนวณแบบ Single Module เป็นตารางคำนวณเพียงตารางเดียวที่คุณสามารถกำหนดให้ตารางคำนวณนั้นเลือกรับตัวแปรต่างไปจากเดิมได้ตามต้องการ ทำให้สามารถใช้ตารางคำนวณ

เพียงตารางเดียวหรือหน้าเดียวเพื่อแสดงผลการคำนวณเปลี่ยนไปเป็นตารางคำนวณรายละเอียดของเดือนที่ต้องการหรือวัตถุดิบที่ต้องการ โดยใช้โครงสร้างตารางแบ่งออกเป็น 3 ส่วน ดังนี้

1. **ส่วนของตาราง Input** ใช้สำหรับเก็บตัวแปรที่ต้องการตามรายเดือนหรือแต่ละชุด ซึ่งตาราง Input นี้ใช้จัดเก็บข้อมูลรายเดือนหรือแต่ละชุดไว้เป็นแนวนอนหรือแนวตั้งติดกันก็ได้
 2. **ส่วนของตารางสูตรตัวกลางใช้ดึงข้อมูล** เพื่อเลือก Input ตามเลขเดือนหรือเลขที่ของชุด ซึ่งมักใช้สูตร Index เพื่อเลือกดึงข้อมูลจากตาราง Input ในชีทเดียวกัน หรือใช้สูตร Indirect เพื่อเลือกดึง Input จากตารางที่แยกชีทกัน
 3. **ส่วนของตารางคำนวณ** เพื่อคำนวณตามค่าที่รับมาจากตารางสูตรตัวกลาง
-

ตัวอย่างโครงสร้างตารางคำนวณแบบ Single Module ในชีทเดียว

	F3												
	A	B	C	D	E	F	G	H	I	J	K	L	
1		SINGLE MODULE				CaseNum							
2		กรณี Link ภายในชีทเดียวกัน				2							
3		Sales	200			200			Sales	100	200	300	
4		Cost1	12			12			Cost1	10	12	13	
5		Cost2	22			22			Cost2	20	22	23	
6		Cost3	32			32			Cost3	30	32	33	
7		Margin	134										
8		30%	Tax	40.2									
9			Profit	93.8									
10													
11		Range Name ที่ใช้											
12		CaseNum											
13		MyData											
14													

ตัวอย่างนี้เป็นการคำนวณงบกำไรขาดทุนจากตัวอย่างประกอบการอบรมหลักสูตรสุดยอดเคล็ดลับและลัดของ Excel ซึ่งกำหนดให้มีตารางที่ต้องการคำนวณตามยอดขายและต้นทุนที่ต่างกัน ไปอยู่ 3 case แต่ละ case กำหนดให้แปรเปลี่ยนยอดขายและต้นทุนต่างกันไป ดังนั้นแทนที่จะต้องสร้างตารางคำนวณขึ้น 3 ตารางเพื่อแยกคำนวณแต่ละ case ขอให้ออกแบบตารางในชีทเดียวกันแบ่งเป็น 3 ส่วน

ส่วนแรกใช้สำหรับบันทึกตัวแปรแต่ละ case ซึ่งในภาพใช้ตารางด้านขวาสุดตั้งแต่เซลล์ I3:K6 (ตั้งชื่อ Range Name ตารางส่วนนี้ว่า MyData) แบ่งแยกให้แต่ละ column บันทึกตัวแปรของแต่ละ case เช่น case1 เป็นพื้นที่ตั้งแต่เซลล์ I3:I6 โดยกำหนดให้เรียงลำดับตัวแปรของยอดขายและต้นทุนต่อกันไปใน column เดียวกัน แล้วใช้ column ถัดไปบันทึกตัวแปรของ case ต่อไปในทำนองเดียวกัน

สาเหตุที่ต้องจัดให้บันทึกตัวแปรของแต่ละ case ไว้ในแนวตั้งเดียวกันนั้น เพื่อให้สามารถใช้สูตร Index ดึงข้อมูลตัวแปรที่เก็บไว้ในแนวตั้งเดียวกันไปพร้อมกัน โดยสร้างสูตร Index ต่อไปในเซลล์ F3:F6 แบบ Array (โดยเลือกเซลล์ F3:F6 แล้วสร้างสูตร จากนั้นกดปุ่ม Ctrl+Shift+Enter พร้อมกันเพื่อบันทึกสูตรลงไปทีเดียวทุกเซลล์ จะพบเครื่องหมายวงเล็บปีกกาปิดหัวท้ายสูตรให้เอง)

=INDEX(MyData, 0, CaseNum)

- MyData เป็นตารางบันทึกตัวแปรตั้งแต่เซลล์ I3:K6
- เลข 0 เพื่อกำหนดให้สูตร Index ดึงข้อมูลทุก Row

- CaseNum คือเซลล์ F2 ใช้สำหรับใส่ตัวเลข 1, 2, หรือ 3 ลงไป เพื่อให้สูตร Index ดึงข้อมูล Input จาก Case1, Case2, หรือ Case3 ตามลำดับ

เมื่อตั้งตัวแปรตามตัวเลขของ CaseNum ที่ต้องการมาแสดงในเซลล์ F3:F6 ได้แล้ว จากนั้นจึงสร้างสูตร link ตามปกติไปใช้คำนวณงบกำไรขาดทุนในส่วนของตารางด้านซ้ายสุดจากเซลล์ D3:D9

ในกรณีที่ต้องการเปรียบเทียบผลของการคำนวณทุก case พร้อมกัน ให้สร้างตาราง Data Table แบบ 1 ตัวแปร ที่ใช้ Row Input Cell เป็นเซลล์ CaseNum ตามรูปต่อไปนี้

H12													=D3	
	A	B	C	D	E	F	G	H	I	J	K	L		
1	SINGLE MODULE					CaseNum								
2	กรณี Link ภายในชีทเดียวกัน					2								
3		Sales	200			200			Case1	Case2	Case3			
4		Cost1	12			12			100	200	300			
5		Cost2	22			22			10	12	13			
6		Cost3	32			32			20	22	23			
7		Margin	134						30	32	33			
8	30%	Tax	40.2			สูตร Array ในเซลล์ F3:F6 {=INDEX(MyData,0,CaseNum)}								
9		Profit	93.8											
10														
11	Range Name ที่ใช้													
12	CaseNum	=AllinOne!\$F\$2												
13	MyData	=AllinOne!\$I\$3:\$K\$6												
14														
15														
16														
17														
18														
19														

Data Table Row Input Cell =CaseNum				
CaseNum >	1	2	3	
Sales	200	100	200	300
Cost1	12	10	12	13
Cost2	22	20	22	23
Cost3	32	30	32	33
Margin	134	40	134	231
Tax	40.2	12	40.2	69.3
Profit	93.8	28	93.8	161.7

เซลล์ H11:K18 เป็นพื้นที่ตารางที่ใช้กับคำสั่ง Data > Table แล้วกำหนดให้เซลล์ F2 หรือ CaseNum เป็น Row Input Cell หรือเซลล์มีรับค่าตัวแปรจากหัวตารางตามแนวนอน ซึ่งกำหนดให้ใช้เซลล์ I11:K11 บันทึกหมายเลข 1, 2, 3 ซึ่งหมายถึงเลขที่ case ใด ส่วนหัวตารางด้านซ้ายสุดตั้งแต่ H12:H18 เป็นสูตรที่ link การคำนวณมาจากตารางคำนวณงบกำไรขาดทุนจากเซลล์ D3:D9

เมื่อตาราง Data Table นี้ทำงาน เราจะได้ผลลัพธ์การคำนวณของแต่ละ case เสร็จในพริบตา แต่ถ้าเข้าใจขั้นตอนการทำงานของ Excel ที่ละเอียดขึ้น จะพบว่า Excel ค่อยๆส่งตัวเลขตัวแปร 1, 2, 3 บนหัวตารางด้านบนไปที่เซลล์ F2 หรือ CaseNum ทีละค่า จากนั้นสูตร Index จะทำหน้าที่ดึงตัวแปรแต่ละตัวตามเลข case ส่งไปคำนวณในงบกำไรขาดทุนด้านซ้ายสุด แล้วจึงส่งผลลัพธ์กลับมาแสดงในพื้นที่ตาราง I12:K18 ตาม case บนหัวตารางด้านบนของพื้นที่ซึ่งส่ง

Data > Table นั้นเอง (การใช้ Data > Table ร่วมกับสูตร Index นี้จะทำให้ Data > Table สามารถส่งตัวแปรไปใช้คำนวณได้ไม่จำกัดจำนวนตัวแปร ถือเป็นเคล็ดลับการใช้ Excel ที่น้อยคนนักจะใช้เป็น)

ตัวอย่างโครงสร้างตารางคำนวณแบบ Single Module แบบแยก Input ไว้ในชีทอื่น

ในกรณีที่มีการคำนวณมีความสลับซับซ้อนมากขึ้นหรือมีจำนวนตัวแปรมากขึ้น คุณสามารถแยกตารางส่วนของการคำนวณผลลัพธ์ไปไว้ในชีทอื่นต่างหาก เพื่อให้สามารถจัดโครงสร้างตารางได้ยืดหยุ่นและแยกชีทเก็บข้อมูลให้แยกเป็นส่วนๆ ได้ชัดเจนยิ่งขึ้น

ในตัวอย่างต่อไปเป็นการคำนวณทางสถิติของการวัด (Calibration) ก่อนอื่นขอเล่าที่ไปที่มาของตัวอย่างต่อไปนี้ก่อนว่า แต่เดิมนั้นแฟ้มมีขนาดใหญ่มากเพราะมีชีทนับร้อยชีท แต่ละชีทมีสูตรคำนวณทางสถิติยาวเหยียดหลายๆ เซลล์

ทุกครั้งที่มีการผลิต ทางโรงงานต้องสุ่มสินค้าที่ผลิตได้ส่งมาตรวจสอบคุณภาพก่อน โดยวัดผลออกมาเป็นตัวเลขแล้วนำไปคำนวณทางสถิติว่าเป็นตัวเลขที่อยู่ในช่วงที่ยอมรับได้หรือไม่ ทุกครั้งที่จะคำนวณ ก็ต้องเปิดชีทใหม่แล้ว copy ตารางคำนวณจากชีทเก่ามาแก้ตัวเลขตามการวัดที่เกิดขึ้นใหม่ ดังนั้นถ้าต้องคำนวณกันทุกวัน วันละหลายๆ ครั้ง ย่อมทำให้มีชีทใหม่เกิดขึ้นทุกครั้ง

นอกจากแฟ้มจะมีขนาดใหญ่เพราะมีชีทหลายๆ ชีทแล้ว สิ่งที่น่าห่วงคือทุกครั้งที่ต้องสร้างตารางคำนวณในชีทใหม่นั้น อาจสร้างตารางที่มีสูตรคำนวณผิดเพี้ยนไปจากสูตรที่สร้างไว้ในชีทเดิม หรือแม้คำนวณได้ถูกต้อง แต่นานไปอาจพลั้งเผลอย้อนกลับไปแก้ไขบางเซลล์ในบางชีทให้ต่างไปจากที่เคยเป็น และเนื่องจากมีชีทหลายชีท จึงเป็นไปได้ว่าเมื่อทำการแก้ไขไปแล้ว อาจหลงลืมจำไม่ได้ว่าแก้ไขชีทชื่อใด และแก้ไขอะไรที่เซลล์ใดไปบ้าง ยิ่งกว่านั้นเมื่อใดที่ต้องการแก้ไขโครงสร้างตารางคำนวณตามหน้าตาใหม่ที่ผู้บริหารกำหนด ก็ต้องเสียแรงเสียเวลา ย้อนกลับไปแก้ไขชีทเก่าทั้งหมดที่อาจมีถึงนับร้อยนับพันชีทเข้าไปแล้วทีเดียว

แทนที่จะต้องเปิดชีทใหม่ ขอให้ใช้หลักการสร้างตารางคำนวณแบบ Single Module โดยใช้โครงสร้างตารางแบ่งออกเป็น 3 ส่วน ดังนี้

1. **ส่วนของตาราง Input** ใช้สำหรับเก็บตัวแปรที่ต้องการตามการวัดคุณภาพสินค้าแต่ละครั้ง ซึ่งตาราง Input นี้ใช้จัดเก็บข้อมูลการวัดแต่ละครั้งไว้เป็นแนวตั้งติดกัน
2. **ส่วนของตารางสูตรตัวกลางใช้ดึงข้อมูล** เพื่อเลือก Input ตามเลขที่ของเลขครั้งในการวัด โดยใช้สูตร Index เพื่อเลือกดึงข้อมูลจากตาราง Input ในชีทเดียวกัน
3. **ส่วนของตารางคำนวณ** เพื่อคำนวณตามค่าที่รับมาจากตารางสูตรตัวกลาง แต่เนื่องจากการคำนวณที่ซับซ้อนและเป็นรายงานที่ต้องใช้ตารางขนาดใหญ่ จึงให้ใช้ชีทอีกชีทหนึ่งเป็นส่วนของการคำนวณโดยเฉพาะ

จากเดิมซึ่งต้องมีหลายชีท จึงกลายเป็นแฟ้มใหม่ซึ่งมีชีทเพียง 2 ชีท ชีทหนึ่งเป็นชีทที่ใช้บันทึกผลการวัดคุณภาพแต่ละครั้ง แล้วใช้อีกชีทหนึ่งเป็นชีทแสดงการคำนวณทางสถิติ

โครงสร้างภายในชีทสำหรับบันทึกตัวเลขจากการวัด

Certificate_No.			=INDEX(DataRange,D2,CaseNo)										
A	B	C	D	E	F	G	H	I	J	K	L	M	N
2	Certificate No.	A005	1	A001	A002	A003	A004	A005	A006	A007	A008	A009	A010
3	Uncertainty Budget for	Oscilloscope	2	Oscilloscope	Oscilloscope	Oscilloscope	Oscilloscope	Oscilloscope	Oscilloscope	Oscilloscope	Oscilloscope	Oscilloscope	Oscilloscope
4	For model	xBFc5	3	xBFc1	xBFc2	xBFc3	xBFc4	xBFc5	xBFc6	xBFc7	xBFc8	xBFc9	xBFc10
5	Unit	mS	4	mV	mV	mV	mV	mS	mS	s	s		
6	Range	0.2	5	1	2	5	10	0.2	0.5	1	2		
7	Calculated at	0.2	6	1	2	5	10	0.2	0.5	1	2		
8	u ₀ 1	0.0154	7					0.0154	0.0385	0.077	0.154		
9	%rdg.	0	8	0.25	0.25	0.25	0.25						
10	Prob. Dist. U ₀ 1	R	9	R	R	R	R	R	R	R	R	R	R
11	u ₀ 2	0.04	10	0.2	0.4	1	2	0.04	0.1	0.2	0.4		
12	Prob. Dist. U ₀ 2	R	11	R	R	R	R	R	R	R	R	R	R
13	u _s	0	12	0	0	0	0	0	0	0	0		
14	No. of Samples	5	13	5	5	5	5	5	5	5	5		
15	Prob. Dist. U _s	N	14	N	N	N	N	N	N	N	N		
16	K	2	15	2	2	2	2	2	2	2	2		
17	Note 1	xxxx	16	xxxx		xxxx		xxxx		xxxx			
18	Calibrated by	Sherlock	17	James Bond	Robin Hood	Tom	Peter	Sherlock	Sun	SFK	Kamrai		
19	Cal Date	23/8/2008	18	3/7/2001	3/7/2001	23/8/2008	23/8/2008	23/8/2008	23/8/2008	23/8/2008	23/8/2008		
20	Certified by	Holm	19	007	Sherwood	Jerry	Pann	Holm	Moon	PS	Thakorn		
21	Ct Date	23/8/2008	20	3/7/2001	3/7/2001	23/8/2008	23/8/2008	23/8/2008	23/8/2008	23/8/2008	23/8/2008		
22													

ตารางตั้งแต่เซลล์ E2:E21 ใช้สำหรับบันทึกข้อมูลจากการวัดครั้งแรกแล้วใช้ column ถัดไปบันทึกข้อมูลจากการวัดในครั้งถัดไปติดต่อกันไปตามแนว column โดยในตัวอย่างนี้ตั้งชื่อพื้นที่ตารางส่วนที่ใช้บันทึกข้อมูลจากการวัดนี้ว่า DataRange

ตารางด้านซ้ายตั้งแต่เซลล์ C2:C21 เป็นสูตร Index ใช้สำหรับดึงข้อมูลที่บันทึกไว้ตามเลขที่ของการวัด ซึ่งใช้ตัวเลข 1-20 ใน Column D ช่วยในการชี้ตำแหน่งเลขที่ row

=INDEX(DataRange, D2, CaseNo)

- DataRange คือพื้นที่ตาราง E2:N21 ใช้สำหรับบันทึกข้อมูลจากการวัด
- D2 เป็นเลขที่ row เนื่องจากเป็นค่าแรกจึงใช้ดึงค่าจาก row ที่ 1
- CaseNo เป็นเซลล์รับเลขที่ครั้งของการวัด ซึ่ง link มาจากชีทตารางคำนวณทางสถิติ

โครงสร้างภายในชีทสำหรับคำนวณทางสถิติของการวัด

CaseNo

5

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

Case No.

5

Uncertainty Budget for Oscilloscope

For model xBFc5

Range: 0.2 mS Calculated at 0.2 mS

Certificate No. A005

Symbol	Source of Uncertainty	Value mS	Prob. Dist.	Divisor	c_i	$U_i ()$ mS	V_{eff}
u_{b1}	Accuracy of STD	0.0154	R	$\sqrt{3}$	1	0.008 891	∞
u_{b2}	Resolution of UU 0.04 mS	0.02	R	$\sqrt{3}$	1	0.011 547	∞
u_g	Repeatability 0.00	0	N	1	1	0.000 000	4
$U_c ()$	Combined Uncertainty		normal			0.014 573	∞
U	Expanded Uncertainty		k = 2.00			0.029 147	∞
						\pm	0.029 147 mS

xxxx

23/08/2008

เซลล์ B4 ถือเป็นหัวใจของชีทนี้ เมื่อคุณพิมพ์เลขที่ CaseNo ลงไป จะทำให้สูตร Index ใน column C ของอีกชีทหนึ่งตามภาพก่อนหน้านี้ ดึงข้อมูลส่งต่อมาให้แสดงหรือคำนวณในตารางคำนวณ Uncertainty Budget อีกทั้งยังใช้เซลล์ B4 ร่วมกับคำสั่ง Data > Table เพื่อสรุปผลเปรียบเทียบการวัด หรือใช้ร่วมกับคำสั่งพิมพ์อัตโนมัติที่เขียนด้วย VBA ตามนี้ได้อีกด้วย

```

For i = StartNum To StopNum
    Range("CaseNo") = i
    Calculate
    ActiveWindow.SelectedSheets.PrintPreview
Next i

```

- StartNum เป็นตัวแปรกำหนดตัวเลขเริ่มของเลขที่ Case ที่ต้องการพิมพ์เป็นรายการแรก
- StopNum เป็นตัวแปรกำหนดตัวเลขสุดท้ายของเลขที่ Case ที่ต้องการพิมพ์เป็นรายการสุดท้าย
- For Next Loop จะส่งค่า i ไปยังเซลล์ชื่อ CaseNo เพื่อดูผลก่อนพิมพ์บนจอ (Print Preview) ซึ่งถ้าเปลี่ยนเป็นคำสั่ง Print ก็จะมีพรายงานตามรายการที่ต้องการโดยอัตโนมัติ

การออกแบบโครงสร้างตารางคำนวณ ภาค 3

ตารางคำนวณแบบ Single Module ในภาคนี้เป็นตารางคำนวณในกรณีที่แยกชีทหรือแยกแฟ้มเพื่อใช้กับงานที่มีการคำนวณสลับซับซ้อนมากกว่าเดิม แทนที่จะมีตารางคำนวณเพียงตารางเดียวแล้วใช้ข้อมูลตัวแปรส่งค่าใหม่ไปคำนวณในตารางนั้นๆในชีทเดียวกัน คราวนี้ผมจะนำเสนอวิธีการสร้างตารางคำนวณพร้อมกันในหลายชีทหรือหลายแฟ้ม จากนั้นจึงเลือกนำผลการคำนวณจากชีทหรือแฟ้มที่ต้องการไปนำเสนอผลงานต่อไป

ผู้ใช้ Excel ต้องคิดไว้เสมอว่า แม้ววันนี้ข้อมูลที่เก็บไว้ยังคงมีไม่มากและสามารถใช้ชีทเดียวเก็บข้อมูลทั้งหมดไว้ด้วยกัน แต่ในอนาคตข้อมูลที่ต้องเก็บไว้จะต้องมีปริมาณของข้อมูลมากขึ้นไปเรื่อยๆ จากเดิมที่เคยเก็บไว้ในชีทเดียว จะกลายเป็นต้องเก็บไว้ในชีทหลายชีท และในที่สุดเมื่อแฟ้มมีขนาดใหญ่มาแล้วใช้เวลาคำนวณช้าลง ก็ต้องหาทางโยกย้ายข้อมูลแยกออกไปเก็บไว้ในแฟ้มใหม่ในที่สุด พฤติกรรมเช่นนี้ถือเป็นวิวัฒนาการของการใช้ Excel ที่เราทุกคนหลีกเลี่ยงไม่ได้

ดังนั้นการออกแบบโครงสร้างตารางคำนวณที่ดี จึงต้องหาทางออกแบบตารางตั้งแต่แรกให้ใช้พื้นที่ตารางแยกเป็นส่วนๆ เพื่อสามารถโยกย้ายตารางจากเดิมที่เก็บอยู่ในชีทเดียว ไปแยกเก็บในชีทใหม่หรือแฟ้มใหม่ได้ทันทีที่ต้องการ

ลักษณะของตารางคำนวณ Single Module แบบแยกชีทหรือแยกแฟ้ม

1. ตารางคำนวณที่แยกชีทนี้ มิได้หมายความว่าให้แยกแต่ละลำดับของการคำนวณให้ใช้พื้นที่ต่างชีทแล้วส่งผลการคำนวณไปคำนวณต่อในชีทอื่นต่อกันไป แต่เป็นการรวมลำดับการคำนวณทั้งหมดไว้ในชีทเดียวกัน จากนั้นจึงสร้างชีทใหม่ให้มีตารางคำนวณซ้ำกับตารางคำนวณที่มีอยู่ในชีทเดิม เพียงแต่ว่าการคำนวณที่เกิดขึ้นในแต่ละชีทจะเกิดจากตัวแปรที่ต่างกัน เช่น ชีทเดิมใช้คำนวณตัวเลขในอดีต ส่วนชีทใหม่ให้ใช้คำนวณตัวเลขในปัจจุบัน จากนั้นจึงนำผลที่คำนวณได้นามาเปรียบเทียบหาความแตกต่างระหว่างอดีตและปัจจุบัน
2. เมื่อใดที่ใช้ชีทหลายๆชีทแยกตารางคำนวณ ย่อมแสดงว่าชีททั้งหมดยังคงอยู่ในแฟ้มเดียวกัน เมื่อนั้นย่อมทำให้แฟ้ม Excel มีขนาดใหญ่ขึ้น ซึ่งหากในการใช้งานไม่ได้ต้องการให้ทุกตารางทำการคำนวณพร้อมกัน ก็ไม่จำเป็นต้องเก็บชีททั้งหมดไว้ในแฟ้มเดียวกัน ก็หาทางแยกชีทไปแยกเก็บเป็นแฟ้มได้ทันที จากนั้นเมื่อใดที่ต้องการให้แฟ้มใดคำนวณก็จัดการเลือกเปิดเฉพาะแฟ้มที่ต้องการขึ้นมาสั่งคำนวณ
3. แต่ละชีทหรือแฟ้มล้วนมีตารางคำนวณที่มีโครงสร้างและลำดับคำนวณเดียวกัน ซึ่งไม่ใช่เรื่องยากที่จะสร้างชีทหรือแฟ้มขึ้นใหม่ให้มีตารางคำนวณแบบเดิมเพราะเพียงแค่ copy ตารางทั้งตารางไปทับก็ใช้งานได้แล้ว แต่ถ้ามีการแก้ไขสูตรคำนวณหรือย้ายตำแหน่ง

เซลล์ในชีทใด จึงเป็นข้อควรระวังว่าต้องติดตามไปแก้ไขทุกชีทและทุกแฟ้มให้มีสูตรคำนวณและตำแหน่งเซลล์ตรงกันด้วยในแต่ละชีทหรือแต่ละแฟ้ม

4. แทนที่จะสร้างตารางคำนวณแยกชีทหรือแยกแฟ้มซึ่งทำให้เกิดข้อควรระวังในการติดตามไปแก้ไขสูตรในแต่ละตารางคำนวณดังกล่าวข้างต้นแล้ว ยังมีวิธีที่ดีกว่าโดยการใช้ชีทเดียวเป็นตารางคำนวณ ส่วนชีทอื่นๆให้ใช้สำหรับเก็บค่าตัวแปร จากนั้นจึงใช้สูตรดึงข้อมูลตัวแปรจากชีทที่ต้องการมาคำนวณในชีทเดียว และเมื่อใดที่ต้องการเปรียบเทียบผลของการคำนวณก็สามารถใช้ Data Table สรุปผลในลักษณะเช่นเดียวกับโครงสร้างตารางคำนวณ Single Module ในชีทเดียวตามที่ได้นแนะนำไว้ในภาคก่อน

ตัวอย่างโครงสร้างตารางคำนวณ Single Module

แบบแยกชีท (อย่างง่าย)

	E	F	G
1			
2		Case1	
3	Sales	100	
4	Cost1	10	
5	Cost2	20	
6	Cost3	30	
7			

	E	F	G
1			
2		Case2	
3	Sales	200	
4	Cost1	12	
5	Cost2	22	
6	Cost3	32	
7			

	E	F	G
1			
2		Case3	
3	Sales	300	
4	Cost1	13	
5	Cost2	23	
6	Cost3	33	
7			

ตัวอย่างนี้ดัดแปลงมาจากตัวอย่างที่ใช้ในการอบรมหลักสูตรสุดยอดเคล็ดลับและลัดของ Excel จากเดิมซึ่งรวมตัวแปรของงบการเงินทั้งสาม case ไว้ติดกันในตารางเดียวกันในชีทเดียวกัน เปลี่ยนมาเป็นการแยกตัวแปรของแต่ละ case ไว้ในชีทต่างหากของมันเอง

โปรดสังเกตว่านอกเหนือจากตำแหน่งของตารางในชีท Case1, Case2, และ Case3 ต้องใช้ตำแหน่งเซลล์ตรงกันทั้งสามชีทคือใช้เซลล์ F3:F6 เก็บตัวเลขของยอดขายและต้นทุนแล้ว การตั้งชื่อชีทให้มีค่านำหน้าว่า Case เหมือนกันแล้วตามด้วยตัวเลข 1, 2, หรือ 3 ถือเป็นส่วนสำคัญที่จะช่วยให้สูตรที่ใช้ดึงข้อมูลจากชีทที่ต้องการสามารถทำได้ง่ายขึ้นด้วย ดังรูปต่อไปนี้ซึ่งเป็นชีทชื่อ SheetModule เป็นตารางคำนวณที่เลือกดึงข้อมูลจากชีทชื่อที่ต้องการมาคำนวณ

F3		=INDIRECT("Case"&\$F\$2&"!"&ADDRESS(ROW(),COLUMN()))													
SingleModule.xls [Compatibility Mode]															
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	
1	SINGLE MODULE				CaseNum										
2	กรณี Link ข้ามชีทภายในแฟ้มเดียวกัน					2									
3		Sales	200			200									
4		Cost1	12			12									
5		Cost2	22			22									
6		Cost3	32			32									
7		Margin	134			สูตรในเซลล์ F3:F6									
8	30%	Tax	40.2			=INDIRECT("Case"&\$F\$2&"!"&ADDRESS(ROW(),COLUMN()))									
9		Profit	93.8												
10															
SheetModule Case1 Case2 Case3															

- ประเด็นสำคัญคือตำแหน่งเซลล์ F3:F6 เป็นตำแหน่งเดียวกับข้อมูลที่แยกเก็บไว้ในชีท Case1, Case2, และ Case3
- เซลล์ F2 เป็นเซลล์เดียวเท่านั้นที่มีการแก้ไขตัวเลข คุณสามารถบันทึกตัวเลข 1, 2, หรือ 3 ลงไปเพื่อแสดงถึงเลขที่ของ CaseNum ตาม case ที่ถูกแยกไว้ในชีทอื่น ดังนั้นเมื่อนำตัวเลขในเซลล์ F2 ไปต่อท้ายกับคำว่า Case โดยใช้สูตร ="Case"&\$F\$2 ย่อมได้ชื่อชีทที่ต้องการ (ตามภาพนี้ F2 มีเลข 2 บันทึกอยู่ ดังนั้นสูตร ="Case"&\$F\$2 จึงคืนค่าเป็นคำว่า Case2)
- สูตรในเซลล์ F3:F6 คือ
=INDIRECT("Case"&\$F\$2&"!"&ADDRESS(ROW(),COLUMN()))
 - สูตรส่วนตั้งแต่ "Case"&\$F\$2&"!" จะคืนค่าออกมาเป็นคำว่า Case2!
 - สูตรส่วนของ ADDRESS(ROW(),COLUMN()) คืนค่าออกมาเป็นตำแหน่งเซลล์ของ เลขที่ row และเลขที่ column ของตำแหน่งเซลล์ที่มีสูตรนี้อยู่ เช่น ในเซลล์ F3 ส่งผลให้สูตร Row() มีค่าเป็น 3 และสูตร Column() มีค่าเป็น 6 เพราะ Column F เป็น Column ที่ 6 จึงส่งค่าไปยังสูตร ADDRESS(3,6) ให้คืนค่าเป็น \$F\$3
 - เมื่อนำผลลัพธ์คำว่า Case2! มาเชื่อมกับคำว่า \$F\$3 ด้วยสูตร "Case"&\$F\$2&"!"&ADDRESS(ROW(),COLUMN()) จึงได้ชื่อชีทกับตำแหน่งเซลล์ที่ต้องการออกมาเป็นคำว่า "Case2!\$F\$3"
 - สูตร Indirect จึงหาค่าจากชื่อชีทและตำแหน่งเซลล์ "Case2!\$F\$3" ได้ตัวเลข 200 ตามต้องการ
- เมื่อหาค่าด้วยสูตร Indirect ในเซลล์ F3:F6 ได้แล้ว จึงส่งค่าต่อไปคำนวณในตารางคำนวณงบการเงินในพื้นที่ตารางด้านซ้ายสุด D3:D6 เพื่อใช้คำนวณหา Margin

เคล็ดลับที่ใช้ร่วมกับโครงสร้างตารางคำนวณ Single Module แบบแยกชีท

- กรณีต้องการสร้างชีทหลายชีทให้มีหน้าตาแบบเดียวกันโดยการสร้างงานพร้อมกันเพียงครั้งเดียว ให้เริ่มจากกดปุ่ม Ctrl พร้อมกับคลิกที่ Sheet Tab เพื่อเลือกชีทที่ต้องการ หรือหากต้องการเลือกทุกชีทที่ติดกัน ให้เริ่มจากอยู่ในชีทแรกแล้วกดปุ่ม Shift พร้อมกับคลิกเลือกชีทสุดท้าย ซึ่งจะพบคำว่า [Group] แสดงไว้ต่อท้ายชื่อแฟ้มที่ปรากฏอยู่ด้านบนสุดของหน้าจอ จากนั้นทุกอย่างที่คุณสร้างลงไปในชีทจะถูกสร้างพร้อมกันลงไปทุกชีท

- เลือกไว้ใน [Group] เดียวกัน และเมื่อต้องการเลิกงานให้คลิกเลือกชีทอื่นที่ไม่ได้เลือกไว้ หรือชีทอื่นในกรณีที่เดิมเลือกไว้ทุกชีท ซึ่งจะพบว่าคำว่า [Group] หายไป
2. กรณีต้องการสร้างชีทหลายชีทให้มีหน้าตาแบบเดียวกันโดยไม่ต้องใช้วิธี Copy จากชีทต้นแบบไปยังชีทอื่น ให้เลือกชีทแบบ [Group] โดยกดปุ่ม Ctrl หรือ Shift แล้วคลิกเลือกชีทเช่นเดียวกับวิธีในข้อแรก จากนั้นให้คลิกเลือกพื้นที่ตารางที่ต้องการ แล้วสั่ง Home > Fill > Across Worksheets
 3. กรณีต้องการคำนวณหาผลรวมของทุกชีท ให้เริ่มจากพิมพ์สูตร =sum(แล้วคลิกเซลล์ในชีทแรกที่ต้องการ จากนั้นกดปุ่ม Shift แล้วคลิกเลือกชีทสุดท้ายแล้วกดปุ่ม Enter จะได้สูตร =SUM(Case1:Case3!F3) โดยสูตรนี้จะหาผลรวมของเซลล์ F3 จากทุกชีทตั้งแต่ชีทชื่อ Case1 ถึง Case3 ให้ทันที โดยไม่ต้องเสียเวลาสร้างสูตรบวกต่อกันทีละเซลล์ทีละชีท ตามสูตร =Case1!F3+Case2!F3+Case3!F3 อีกต่อไป
 4. กรณีต้องการคำนวณหาผลรวมของทุกชีท "เว้น" เฉพาะค่าจากชีทที่มีสูตรยอดรวม ให้พิมพ์สูตร =sum('*!f3) โดยขอสมมติว่าในแฟ้มมี 3 ชีทติดกันตามลำดับ คือชื่อ Sheet1, Sheet2, และ Sheet3 ถ้าสร้างสูตรนี้ลงไปใน Sheet1 จะได้สูตร =SUM(Sheet2:Sheet3!F3) แต่ถ้าสร้างสูตรลงไปใน Sheet2 จะได้สูตร =SUM(Sheet1!F3,Sheet3!F3) หรือถ้าสร้างสูตรลงไปใน Sheet3 จะได้สูตร =SUM(Sheet1:Sheet2!F3)
 5. กรณีต้องการจัดการแสดงโครงสร้างแต่ละชีทให้ดูต่างกันไปหลายๆแบบโดยอัตโนมัติ ให้จัดโครงสร้างตารางโดยอาจสั่ง Hide Row/Column ซ่อนสิ่งที่ไม่ต้องการแสดง เพื่อทำให้เห็นเฉพาะพื้นที่ตารางเท่าที่ต้องการก่อน จากนั้นให้สั่ง View > Custom Views > Adds แล้วตั้งชื่อ View ให้กับโครงสร้างตารางที่เห็นอยู่บนจอในขณะนั้น จากนั้นให้จัดหน้าตาตารางใหม่ในชีทเดิมซ้ำหรือในชีทอื่นตามต้องการแล้วตั้งชื่อ Custom View เพิ่มเดิมจนครบ แล้วเมื่อใดที่ต้องการไปที่ชีทใดก็ไม่จำเป็นต้องใช้วิธีไล่คลิกไปที่ชีทอีก แต่ให้สั่ง View > Custom View แล้วคลิกเลือกชื่อ View แทน ซึ่งวิธีนี้นอกจากช่วยเลือกตารางในชีทที่ต้องการให้แล้ว ในแต่ละ View ยังบันทึก Print Settings และ Filter Settings ตามให้อีกด้วย
 6. กรณีต้องการใช้สูตรเพื่อค้นหาชื่อชีททุกชีทมาแสดงในเซลล์ ให้ใช้สูตร =GetSheetName(เลขที่ชีท) โดยสร้างรหัส Function VBA ต่อไปนี้ไว้ในแฟ้ม


```
Function GetSheetName(x)
    GetSheetName = Sheets(x).Name
End Function
```

ขั้นตอนการปรับโครงสร้างตารางคำนวณ Single Module จากแบบแยกชีทให้เป็นแบบแยกแฟ้ม

1. แก่สูตร Indirect ในเซลล์ F3:F6 เป็นสูตร =Case1!F3, =Case1!F4, =Case1!F5, และ =Case1!F6 ตามลำดับเพื่อ link ข้อมูลมาจากชีทชื่อ Case1 ในตำแหน่งเซลล์เดียวกัน

F3

=Case1!F3

SingleModuleLinkFiles.xls [Compatibility Mode]

A

B

C

D

E

F

G

H

1

SINGLE MODULE

2

กรณี Link ข้ามแฟ้ม

3

Sales

100

100

4

Cost1

10

10

5

Cost2

20

20

6

Cost3

30

30

7

Margin

40

สูตรในเซลล์ F3

8

30%

Tax

12

=Case1!F3

9

Profit

28

10

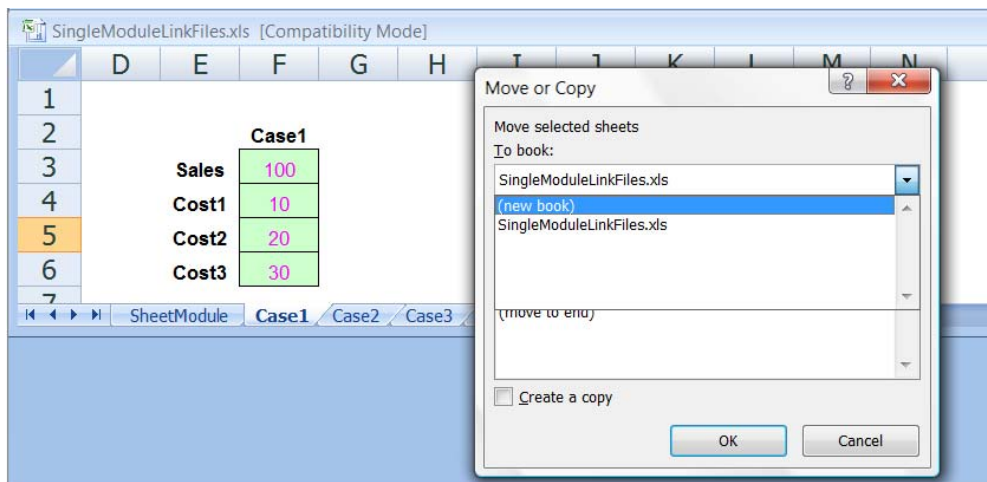
SheetModule

Case1

Case2

Case3

2. Move ชีทชื่อ Case1, Case2, Case3 ไปเป็นแฟ้มใหม่ โดยใช้วิธีคลิกขวาที่ชื่อชีท Case1 แล้วเลือก Move or Copy จากนั้นคลิกเลือก (new book) แล้วกด OK จะพบว่าชีท Case1 ถูกย้ายทั้งชีทไปเป็นแฟ้มใหม่ที่มีชีทเพียงชีทเดียว แล้วให้ทำซ้ำกับชีท Case2 และ Case3 ให้เป็นแฟ้มใหม่แยกจากกันโดยใช้วิธีเดียวกันนี้อีก



3. จัดเก็บแฟ้มที่เกิดขึ้นใหม่จากการย้ายชีท Case1, Case2, และ Case3 โดยตั้งชื่อแฟ้มเป็น SingleModuleLinkFilesCase1.xls, SingleModuleLinkFilesCase2.xls, และ SingleModuleLinkFilesCase3.xls ตามลำดับ แล้วเมื่อกลับไปคลิกดูสูตรในเซลล์ F3:F6 ในชีท SheetModule จะพบว่าสูตรเดิมที่เคยเป็นสูตร link ข้ามชีทถูกแก้ไขเองเป็นสูตร link ข้ามแฟ้ม =[SingleModuleLinkFilesCase1.xls]Case1!F3, = [SingleModuleLinkFilesCase1.xls]Case1!F4, = [SingleModuleLinkFilesCase1.xls]Case1!F5, และ =

[SingleModuleLinkFilesCase1.xls]Case1!F6 ตามลำดับให้เอง

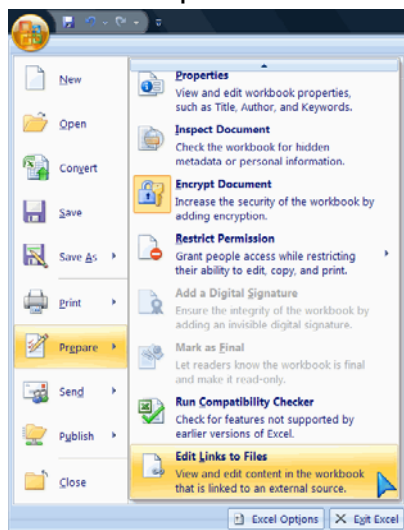
SINGLE MODULE						
3	Sales	100	100			
4	Cost1	10	10			
5	Cost2	20	20			
6	Cost3	30	30			
7	Margin	40				
8	30% Tax	12				
9	Profit	28				

Case2			
3	Sales	200	
4	Cost1	12	
5	Cost2	22	
6	Cost3	32	

Case1			
3	Sales	100	
4	Cost1	10	
5	Cost2	20	
6	Cost3	30	

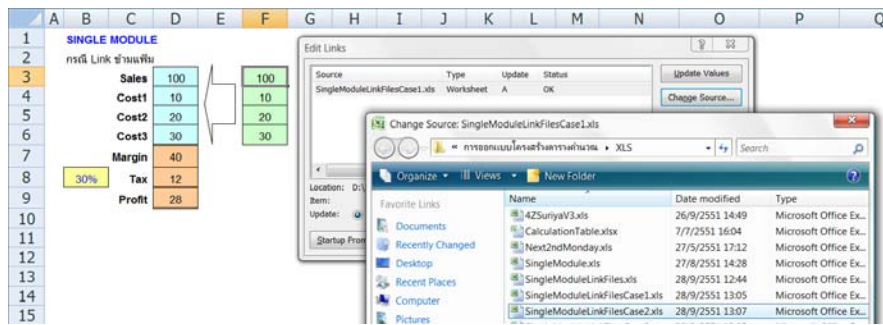
Case3			
3	Sales	300	
4	Cost1	13	
5	Cost2	23	
6	Cost3	33	

4. เมื่อต้องการดึงข้อมูลจาก Case2 หรือ Case อื่น ให้เปิดแฟ้มปลายทางที่มีสูตร link ข้ามแฟ้มแสดงอยู่ขึ้นมาใช้งานเพียงแฟ้มเดียว (โดยไม่ต้องเปิดแฟ้มต้นทาง) จากนั้นสั่ง Office > Prepare > Edit Links to Files



5. ให้คลิกเลือกชื่อแฟ้มต้นทางเดิม แล้วกดปุ่ม Change Source จากนั้นไล่คลิกหาแฟ้มต้นทางใหม่ที่ต้องการ ซึ่งเมื่อเลือกแฟ้มต้นทางใหม่ได้เรียบร้อยแล้วและกดปุ่ม OK > Close กลับมาแฟ้มปลายทาง จะพบว่าสูตรเก่าถูกแก้ไขให้ดึงข้อมูลมาจากแฟ้มต้นทางใหม่ให้

เองทันที



ข้อควรระวังในการใช้ข้อมูลที่ link ข้ามแฟ้ม

1. โครงสร้างภายในของแฟ้มต้นทางต้องเหมือนกันทุกแฟ้ม โดยตำแหน่งเซลล์ใดเป็นข้อมูลอะไรก็ต้องกำหนดให้ทุกแฟ้มใช้ตำแหน่งเซลล์เดียวกันทั้งหมด
2. หากแฟ้มต้นทางมีชีทเดียวแต่ใช้ชื่อชีทต่างกันเช่นในตัวอย่างนี้ใช้ชื่อชีทว่า Case1, Case2, และ Case3 เมื่อสั่ง Change Source จะพบว่า Excel สามารถปรับสูตร link ข้ามแฟ้มให้ถูกต้อง แต่ถ้าแฟ้มต้นทางมีหลายชีท ต้องตั้งชื่อชีทให้ตรงกันทุกแฟ้มด้วย มิฉะนั้น Excel จะมีขั้นตอนเปิดจ่อขึ้นมาถามว่าให้จับชื่อชีทใดแทนด้วยชื่อชีทใด
3. ถ้าต้องการแก้ไขโครงสร้างตารางโดยเฉพาะตำแหน่งเซลล์ในแฟ้มต้นทาง ก่อนจะแก้ไขใดๆ ให้เปิดทุกแฟ้มที่ link กัน ทั้งแฟ้มต้นทาง แฟ้มปลายทาง รวมทั้งแฟ้มต้นทางใหม่ที่ยังไม่ได้ link ขึ้นมาบนจอให้ครบทั้งหมดก่อน จากนั้นจึงค่อยทำการแก้ไข เพื่อให้ Excel ไล่แก้ตำแหน่งอ้างอิงในสูตรที่อ้างอิงถึงแฟ้มที่เกี่ยวข้องให้เอง (ถ้าไม่ได้เปิดแฟ้ม Excel จะละเลยไม่สนใจตามไปแก้ไขให้แม่แต่น้อย)
4. ถ้าต้องการ Save เพื่อตั้งชื่อแฟ้มใหม่หรือย้าย Folder ที่เก็บแฟ้ม ให้สั่ง Save แฟ้มต้นทางก่อนแล้วจึง Save แฟ้มปลายทางทีหลัง (แต่ถ้าไม่ได้แก้ไขชื่อแฟ้มหรือย้าย Folder จะ Save แฟ้มใดก่อนก็ได้ แต่ต้อง Save ให้ครบทุกแฟ้ม)

การออกแบบโครงสร้างตารางคำนวณ

แบบ Compound Module

เคยมีคำกล่าวว่า Excel ไม่สามารถนำมาใช้กับการวางแผนความต้องการวัสดุได้หรอก เราต้องหันไปซื้อโปรแกรมสำเร็จรูปราคาแพงเป็นแสนเป็นล้านบาทมาใช้ในการวางแผนการผลิต ซึ่งโรงงานผลิตสินค้าหลายๆแห่งอาจคิดว่าไม่แพง เพราะเมื่อนำราคาโปรแกรมสำเร็จรูปไปเทียบกับมูลค่าของโรงงานและสินค้าที่ผลิตแล้วถือว่าเป็นสัดส่วนที่น้อยมาก แล้วเมื่อนำโปรแกรมสำเร็จรูปเหล่านี้มาใช้ กลับพบว่าไม่สามารถนำมาใช้กับการวางแผนการผลิตได้อย่างสมบูรณ์ เสียอีก ครั้นจะขอให้ผู้ขายโปรแกรมสำเร็จรูปช่วยแก้ไขดัดแปลงโปรแกรมให้ใหม่ก็ถูกคิดค่าใช้จ่ายเป็นเงินอีกมากมาย ที่แย่ที่สุดก็คือ พอซื้อโปรแกรมมาใช้ได้สักพัก เจ้าตัวบริษัทที่ขายโปรแกรมให้กลับปิดตัวเลิกกิจการไปแล้ว เป็นเหตุให้โปรแกรมสำเร็จรูปที่ได้มาถูกนำมาใช้งานแค่ครึ่งๆกลางๆหรือถึงกับเลิกใช้ไปเลยก็มี

คนที่เก่งคอมพิวเตอร์ไม่ว่าจะเป็นโปรแกรมเมอร์หรือผู้ที่คลุกคลีอยู่ในแวดวง IT มักเข้าใจว่าปัญหาการคำนวณยากๆ ต้องหาทางเขียนโปรแกรมหรือหาซื้อโปรแกรมสำเร็จรูปที่ถูกสร้างขึ้นมามาเพื่อใช้แก้ปัญหายากๆเหล่านั้นโดยเฉพาะ แม้แต่คนที่เก่ง Excel ก็ตาม มักชอบคิดว่าต้องใช้โปรแกรม VBA (Visual Basic for Applications) เป็นทางออกสุดท้ายเมื่อไม่สามารถใช้สูตรหรือคำสั่งบนเมนูของ Excel ช่วยในการแก้ปัญหาที่ว่ายากนั้นได้ ซึ่งถ้าว่าไปแล้วก็ไม่ใช่ว่าการคิดที่ไม่มีเหตุผล เพราะปัญหาการคำนวณยากๆ ยกตัวอย่างเช่นการวางแผนการผลิต มีสิ่งที่ต้องคำนึงถึงหลายอย่างที่เกี่ยวข้องกับการวางแผน เช่น เงื่อนไขที่เกี่ยวข้องกับยอดสินค้าคงเหลือ ต้นงวดปลายงวด ยอดวัตถุดิบที่ต้องสั่งซื้อซึ่งยังสัมพันธ์กับโครงสร้างผลิตภัณฑ์หรืออัตราส่วนผสม หากวัตถุดิบขาดมือก็ต้องสั่งซื้อมาสำรองไว้ให้เพียงพอซึ่งเกี่ยวข้องกับเงื่อนไขการขายของผู้ขายเข้าไปอีก และทั้งหมดนี้ยังต้องหาทางกำหนดเวลาที่จะต้องสั่งผลิตเพื่อให้ได้จำนวนสินค้าตามต้องการ ณ กำหนดเวลาที่ต้องการ แค่นี้ปัญหาที่เกี่ยวข้องเหล่านี้ก็แทบไม่อยากที่จะคิดด้วยสมองแล้วใช่ไหม หลายต่อหลายคนจึงพากันตอบว่า หันไปซื้อโปรแกรมสำเร็จรูปมาใช้กันดีกว่า

จากประสบการณ์ที่ผมคลุกคลีอยู่กับ Excel มานานกว่ายี่สิบปีพบว่า คนที่เก่ง Excel มากๆมักขาดความรู้เรื่องการวางแผนการผลิต ส่วนคนที่เก่งในการวางแผนการผลิตก็มักจะไมเก่ง Excel จึงทำให้ไม่มีใครที่สามารถนำความรู้ทั้ง Excel และการวางแผนการผลิตมาใช้งานร่วมกันเสียที่ทางสมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น) ได้กรุณาให้ผมเข้าอบรมหลักสูตรการวางแผนการผลิตที่สมาคมจัดขึ้น ทำให้เกิดแนวทางใช้ Excel โดยออกแบบตารางคำนวณแบบ Compound

Module นี้ขึ้นมา จึงใคร่ขอใช้โอกาสนี้ ขอบคุนสมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น) เป็นอย่างยิ่ง ซึ่งคงไม่เป็นการกล่าวที่เกินไปว่า ในโลกนี้มีแต่ประเทศไทยของเราเท่านั้นที่มีเทคโนโลยีการออกแบบตารางคำนวณแบบ Compound Module หรือหากประเทศอื่นทำได้ก็น่าจะเรียนรู้มาจากเราก็คงต่อหนึ่ง

ลักษณะโครงสร้างตารางคำนวณแบบ Compound Module

โครงสร้างตารางคำนวณแบบ Compound Module เกิดจากแนวความคิดที่จะใช้ตาราง Excel หลายๆ เซลล์ต่อกันสำหรับการคำนวณที่ต่อเนื่องกันหลายๆ ชั้น พอเซลล์หนึ่งคำนวณชั้นหนึ่งเสร็จ ก็ให้ส่งค่าไปคำนวณต่อในอีกเซลล์หนึ่ง แล้วส่งค่าต่อกันไปยังเซลล์ที่อยู่ในพื้นที่ตารางคำนวณชุดเดียวกัน ซึ่งในตารางคำนวณชุดเดียวกันนี้อาจต้องใช้จำนวนเซลล์ตั้งแต่ 2 เซลล์ขึ้นไปจนถึงนับร้อยนับพันเซลล์ก็เป็นได้เพื่อให้ได้คำนวณตั้งแต่ต้นจนจบ

แนวความคิดดังกล่าวนี้ฟังดูแล้วก็ไม่เห็นเป็นเรื่องแปลกใช้ใหม่ คุณเองก็ใช้ตาราง Excel ในแบบที่กล่าวนี้เป็นกันอยู่แล้ว เพียงแต่ว่าแทนที่จะใช้ตารางคำนวณเพียงชุดเดียวเพื่อคำนวณให้ผลลัพธ์ที่ต้องการแล้วเสร็จ เรายังต้องหาทางใช้ตารางคำนวณชุดถัดไป ที่มีหน้าตาตารางเหมือนกับตารางคำนวณชุดแรก เพื่อใช้ในการคำนวณซ้ำแบบเดียวกับการคำนวณในตารางคำนวณชุดแรกซ้ำต่อไปอีกให้ได้ โดยหาทางสร้างสูตรในตารางคำนวณชุดถัดไปให้สามารถรับค่าจากตารางคำนวณชุดก่อน (Simple Compound Module) หรือเลือกรับค่าตารางคำนวณชุดใดๆก็ได้มาใช้คำนวณซ้ำอีก (Dynamic Compound Module)

องค์ประกอบสำคัญที่เป็นส่วนช่วยให้เราสามารถสร้างตารางคำนวณขึ้นเพียงชุดเดียว จากนั้นสามารถสั่ง Copy แล้วนำไป Paste ให้เกิดตารางคำนวณชุดต่อไปซึ่งมีสูตรซ้ำกับตารางคำนวณชุดแรกทุกเซลล์ก็คือ ขอให้หลีกเลี่ยงการกำหนดตำแหน่งอ้างอิงใดๆในสูตรแบบ Absolute (เช่น \$A\$1) แต่ให้ใช้การกำหนดตำแหน่งอ้างอิงแบบ Relative (เช่น A1) แทน หรือถ้าหลีกเลี่ยงไม่ได้ ก็ขอให้ใช้ตำแหน่งอ้างอิงแบบ Mixed ในแบบควบคุมตำแหน่ง Column ให้คงที่ (เช่น \$A1) โดยให้ใช้ให้น้อยที่สุด เพราะถ้ากำหนดตำแหน่งอ้างอิง \$A1 ไว้ จะต้อง Paste ตารางคำนวณชุดถัดไปในแนว Column เดิมเสมอ มิฉะนั้นการอ้างอิงจะผิดเพี้ยนจากตำแหน่งที่ต้องการ

ผู้ที่คุ้นเคยกับการสร้างตารางคำนวณต้องทราบถึงประโยชน์ของการใส่เครื่องหมาย \$ เพื่อช่วยควบคุมตำแหน่งอ้างอิงในสูตรว่ามีความสำคัญเพียงใด แม้เครื่องหมาย \$ จะช่วยให้เราสร้างสูตรแล้ว Copy ไปใช้ที่เซลล์อื่นได้ทันที แต่ในโครงสร้างตารางคำนวณแบบ Compound Module นี้

เครื่องหมาย \$ กลับเป็นสิ่งต้องห้าม ดังนั้นแม้ในระหว่างการสร้าง Module แรกขึ้นมาใช้งาน ซึ่งเราจำเป็นต้องอาศัยเครื่องหมาย \$ แต่เมื่อสร้างงานเสร็จแล้วก็ต้องลบเฉพาะเครื่องหมาย \$ ที่ใช้ในเซลล์ทิ้งไป โดยใน Excel 2007 ใช้คำสั่ง Home > Find & Select > Replace

ตัวอย่างตารางคำนวณ Compound Module อย่างง่าย

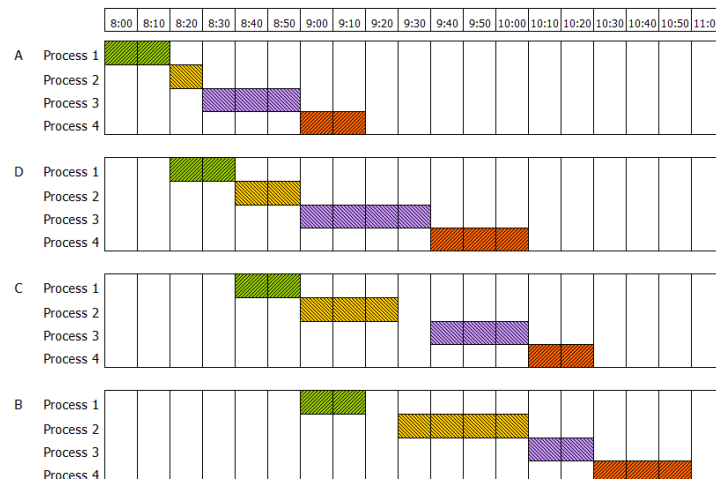
	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2		Time Consumed Table (Unit : h:mm)											
3		Product Name	Process 1	Process 2	Process 3	Process 4		All Processes Conclusion :					
4		A	0:20	0:10	0:30	0:20		Start Time		8:00			
5		B	0:20	0:30	0:20	0:30		Stop Time		11:00			
6		C	0:20	0:30	0:20	0:30		Process Time		3:00			
7		D	0:20	0:10	0:40	0:20							

ตัวอย่างนี้อยู่ในชีทชื่อ SimpleCompound เป็นตารางคำนวณหาระยะเวลาที่ใช้ในการผลิต กำหนดให้ผลิตสินค้า A, B, C, D (ProductName เซลล์ B4:B7) โดยสินค้าแต่ละชนิดต้องใช้เวลาในการผลิตต่างกันไป เช่น ในการผลิตสินค้า A เมื่อผ่านขั้นตอนการผลิต 4 ขั้นตอน (เซลล์ C4:F4) ต้องใช้เวลา 20 นาทีในการผลิตขั้นตอนที่ 1 จากนั้นต้องใช้เวลา 10 นาที ตามด้วย 30 นาที และตามด้วย 20 นาที ในขั้นตอนที่ 2, 3, 4 ตามลำดับจึงจะแล้วเสร็จ

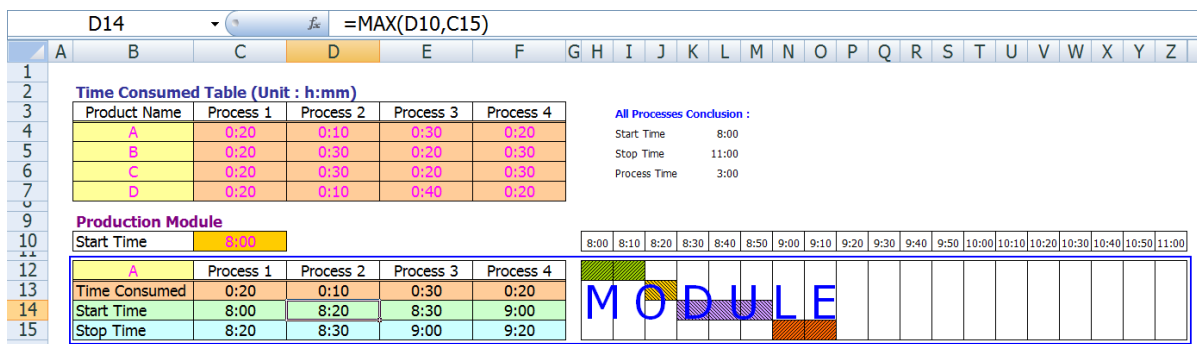
ในกระบวนการผลิตสินค้า A, B, C, และ D นี้ หากกำหนดให้เริ่มต้นผลิตในเวลา 8 นาฬิกา จะผลิตเสร็จในเวลาใด ทั้งนี้กำหนดให้เลือกผลิตสินค้า A, D, C, B ก่อนหลังตามลำดับ (หรือในขนาดอาจจัดลำดับการผลิตก่อนหลังต่างไปก็ได้) โดยหาทางให้ใช้ระยะเวลาผลิตที่น้อยที่สุด เมื่อผลิตสินค้าใดเสร็จให้ผลิตสินค้าชนิดต่อไปได้ทันที

ตารางด้านขวาที่เซลล์ L4:L6 ของรูปข้างต้นเป็นคำตอบที่ได้จากการคำนวณ กล่าวคือ เมื่อเริ่มผลิตในเวลา 8 นาฬิกา จะผลิตสินค้าทั้งหมดเสร็จในเวลา 11 นาฬิกา ถือเป็นระยะเวลาผลิตทั้งสิ้น 3 ชั่วโมง

ปัญหาที่ต้องคำนึงถึงในการกำหนดขั้นตอนการผลิตในตัวอย่างนี้ก็คือ เราต้องหาทางสร้างตารางคำนวณที่สามารถจัดลำดับการผลิตก่อนหลังแบบใดก็ได้ อีกทั้งต้องหาทางสร้างสูตรให้กำหนดเวลาเริ่มผลิตในแต่ละขั้นตอนของสินค้าแต่ละตัว ต่อเนื่องกับเวลาผลิตเสร็จในขั้นตอนก่อนหน้า หรืออีกนัยหนึ่งห้ามสั่งผลิตสินค้าชนิดถัดไป หากสินค้าชนิดแรกยังอยู่ในกระบวนการผลิตขั้นตอนนั้นๆ ดังรูปต่อไปนี้



ขอให้พิจารณาภาพ Gantt Chart ในลำดับการผลิตของ Process 2 จะพบว่าเมื่อสินค้า A ผลิตเสร็จในเวลา 8:30 นาฬิกา แต่ยังไม่สามารถนำสินค้า D มาผลิตต่อใน Process 2 ได้ทันที เพราะในขณะนั้นสินค้า D ยังอยู่ในกระบวนการผลิตของ Process 1 อยู่ ต้องรอจนถึงเวลา 8:40 นาฬิกา จึงเริ่มผลิตสินค้า D ใน Process 2 ต่อไปได้ แล้วต่อเมื่อสินค้า D ผลิตเสร็จใน Process 2 ก็สามารถนำสินค้า C และ B มาผลิตต่อใน Process 2 ได้ต่อเนื่องทันที

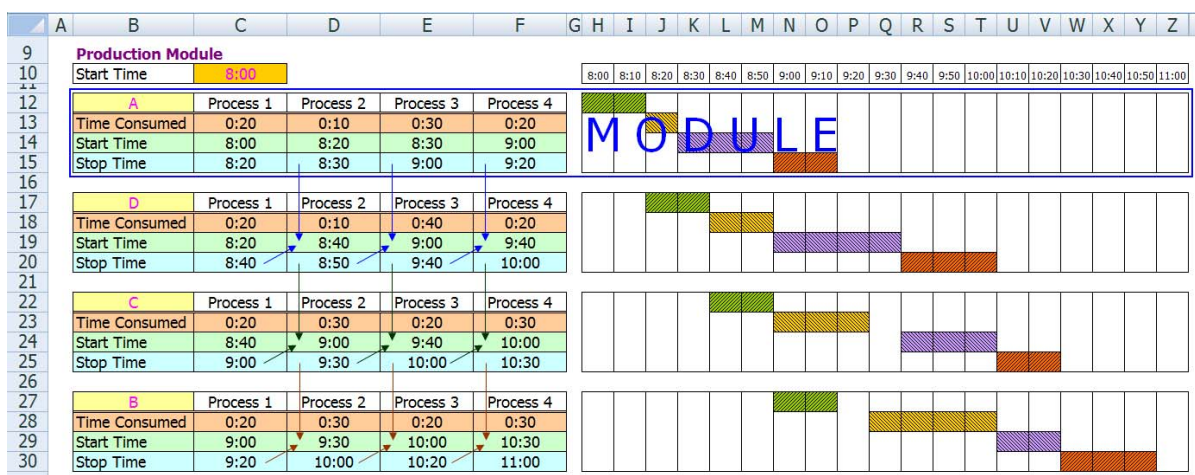


Compound Module ในตัวอย่างนี้คือพื้นที่ตารางตั้งแต่เซลล์ B12:Z15 โดยกำหนดให้ใช้เซลล์ B12 ตรงหัวมุมซ้ายบนสุดสำหรับบันทึกชื่อสินค้าที่ต้องการลงไป จากนั้นเซลล์ C13:F13 จะตั้งระยะเวลาที่ใช้ในการผลิต (Time Consumed) จากตาราง Time Consumed Table ด้านบนของชีทลงมาใช้งาน โดยใช้สูตร =VLOOKUP(\$B12,TimeConsumedTbl,C12+1,0) จากนั้นเมื่อนำเวลาเริ่มผลิต 8:00 ในเซลล์ C10 มาบวกเพิ่มด้วยระยะเวลาที่ใช้ในการผลิต ทำให้คำนวณหา Start Time และ Stop Time ในพื้นที่เซลล์ C14:F15

เซลล์สำคัญที่สุดที่ทำให้ Compound Module สามารถนำไปใช้งานต่อไปได้ในตารางชุดอื่นคือเซลล์ D14 มีสูตร =MAX(D10,C15) โดยขอให้สังเกตว่าตำแหน่งเซลล์ D10 ที่กำหนดไว้ในสูตร MAX นี้เป็นตำแหน่งเซลล์ที่อยู่เหนือขอบเขตของตาราง Module (B12:Z15) เพื่อให้

ตำแหน่งเซลล์ D10 ซึ่งไม่ได้กำหนดเครื่องหมาย \$ ไว้หรืออีกนัยหนึ่งคือเซลล์ที่อยู่เหนือพื้นที่ตาราง Module ขึ้นไปอีก 2 เซลล์ สำหรับนำเวลา Stop Time ของ Module ก่อนหน้ามาเทียบกับเวลา Stop Time ของ Module ตัวเอง (D10 vs C15) ว่าให้เลือกนำเวลาที่มากที่สุดมาใช้เป็นกำหนดการเริ่มกระบวนการผลิตในขั้นตอนต่อไป

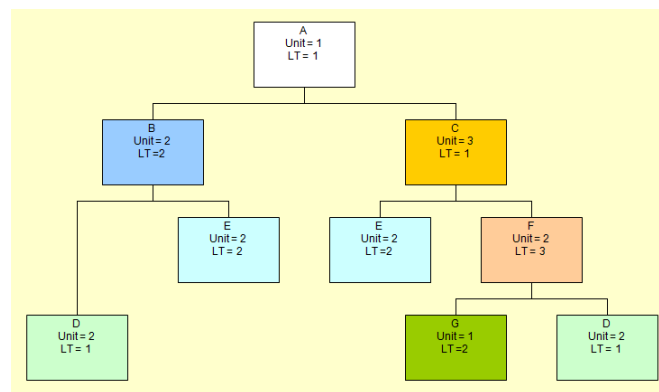
หลังจากสร้าง Module แรกในเซลล์ B12:Z15 เสร็จแล้ว เมื่อต้องการคำนวณหา กำหนดการผลิตของสินค้าในลำดับถัดไป ก็เพียงแค่ Copy เซลล์ B12:Z15 ไป Paste ต่อกันลงไปแล้ว เปลี่ยนชื่อสินค้าที่เซลล์หัวมุมซ้ายของแต่ละ Module ตามชื่อสินค้าที่ผลิตในแต่ละลำดับ ก็จะพบว่า Module ทุกชุดคำนวณร่วมกันได้ระยะเวลาและกำหนดการผลิตแต่ละขั้นตอนตามต้องการในทันที



โปรดสังเกตว่า แต่ละ Module ถูกวางห่างกันโดยเว้นระยะ 1 row ระหว่างกันเสมอ ทั้งนี้เพื่อให้สูตร =MAX(D10,C15) จาก Module แรกเป็นสูตรดึงค่าจาก Stop Time ซึ่งเป็น row ล่างสุดของ Module ด้านบนมาใช้ต่อไปนั่นเอง

ตัวอย่างตารางคำนวณ Compound Module แบบ Dynamic

ตัวอย่างที่ผ่านมาเป็นการกำหนดตำแหน่งอ้างอิงที่เชื่อมค่าจาก Module ที่วางไว้ด้านบนติดกัน เหมาะกับการคำนวณที่มีขั้นตอนต่อเนื่องกันไปแบบเส้นตรง แต่หากการวางแผนการผลิตต้องเกี่ยวข้องกับโครงสร้างผลิตภัณฑ์ (Bill of Materials : BOM) ในโครงสร้างที่ซับซ้อนตามภาพต่อไปนี่ ก็ต้องหาทางพัฒนาโครงสร้างตารางคำนวณแบบ Dynamic



จากภาพแสดงถึงการผลิตสินค้า A ซึ่งหากผลิตเพียง 1 หน่วย (Unit = 1) ต้องอาศัยวัสดุ B จำนวน 2 หน่วยและ C จำนวน 3 หน่วย โดยในการผลิตวัสดุ B ต้องใช้วัสดุ D จำนวน 2 หน่วย และวัสดุ E จำนวน 2 หน่วย ส่วนวัสดุ C ต้องใช้วัสดุ E จำนวน 2 หน่วยและวัสดุ F จำนวน 2 หน่วย ซึ่งในการผลิตวัสดุ F เองก็ต้องอาศัยวัสดุ G จำนวน 1 หน่วยและวัสดุ D จำนวน 2 หน่วย

ส่วนคำว่า LT ย่อมาจากคำว่า Lead Time หมายถึงช่วงเวลานาน ซึ่งเป็นระยะเวลาที่ใช้ในการผลิตหรือระยะเวลาที่ต้องรอคอยในการได้มาของสินค้า จากภาพแสดงว่าสินค้า A ต้องใช้ระยะเวลาผลิต 1 วัน (หรือหน่วยของระยะเวลาอื่นๆก็ได้ ในตัวอย่างนี้ขอใช้ระยะเวลาเป็นหน่วย วัน) วัสดุ B, C, D, E, F, G ต้องใช้ระยะเวลาผลิตหรือระยะเวลารอคอยเท่ากับ 2 วัน, 1 วัน, 1 วัน, 2 วันและ 3 วัน ตามลำดับ (โปรดศึกษารายละเอียดเพิ่มเติมจากหนังสือ ระบบการวางแผนและควบคุมการผลิต โดยรศ.พิภพ ลลิตาภรณ์ สำนักพิมพ์ของสมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น))

ในการออกแบบตารางคำนวณเพื่อวางแผนความต้องการวัสดุ (Material Requirements Planning - MRP) ต้องหาทางสร้างตารางคำนวณที่สามารถคำนวณตามโครงสร้างผลิตภัณฑ์ที่อาจเปลี่ยนแปลงไปจากเดิม เช่น เดิมสินค้า A มีวัสดุ B และ C เป็นโครงสร้างในการผลิต หากในภายหลังมีวัสดุ G เพิ่มเติม หรือเลิกใช้วัสดุ B หรือใช้วัสดุอื่นมาแทนที่ และหากตัวเลขจำนวน

การใช้วัสดุและ LT เปลี่ยนแปลงไปจากเดิม ก็ต้องหาทางประยุกต์ใช้ Excel กับงาน MRP ได้ โดยไม่จำเป็นต้องแก้ไขสูตรคำนวณใหม่แม้แต่บ่อย

อนึ่งในแง่ของโครงสร้างผลิตภัณฑ์ยังมีศัพท์คำว่า Parent กับ Child ที่ต้องทำความเข้าใจ เพราะเป็นหลักการสำคัญที่นำไปใช้ในการสร้างตารางคำนวณ เช่น หากกำหนดให้สินค้า A เป็น Parent ก็จะพบว่าวัสดุ B และ C เป็น Child แล้วหากกำหนดให้วัสดุ B เป็น Parent ก็จะพบว่า วัสดุ D และ E เป็น Child ซึ่งจะเห็นว่าการพิจารณาความสัมพันธ์ระหว่างการเป็น Parent-Child เป็นการพิจารณาความสัมพันธ์ระหว่างการผลิตสินค้าหรือวัสดุในลำดับหนึ่งนั่นเอง

ตัวอย่างต่อไปนี้มาจากชีตชื่อ DynamicCompound ในแฟ้มชื่อ CompoundModule.xls

D14 =MATCH(C14,B\$1:B\$1001,0)+1

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1																
2																
3																
4																
5																
6																
7																
8																
9																
10																
11																
12																
13																
14																
15																
16																
17																
18																
19																
20																
21																
22																

Material Requirements Planning - Dynamic

Item	Parent	Last Row of Module	Quantity	Lead Time	Period	1	2	3	4	5	6	7	8	9
A			1	1	Required Quantity									50
					Planned Order Releases								50	
B	A	5	2	2	Required Quantity								100	
					Planned Order Releases						100			
C	A	5	3	1	Required Quantity								150	
					Planned Order Releases							150		
D	B	7	2	1	Required Quantity									
					Planned Order Releases									
E	B	7	2	2	Required Quantity									
					Planned Order Releases									
E	C	9	2	2	Required Quantity									
					Planned Order Releases									
F	C	9	2	3	Required Quantity									
					Planned Order Releases									
G	F	17	1	2	Required Quantity									
					Planned Order Releases									
D	F	17	2	1	Required Quantity									
					Planned Order Releases									

โครงสร้างตารางคำนวณแบบ Compound Module คือ พื้นที่ตารางตั้งแต่เซลล์ B6:P7 มีเซลล์ที่ใช้รับค่าตัวแปรใน Column B, C, E, และ F โดยใช้เซลล์ B6 สำหรับบันทึกชื่อ Item ในระดับ Child และใช้เซลล์ C6 บันทึกชื่อ Parent โดยใช้เซลล์ E6 และ F6 บันทึกตัวเลขสัดส่วนการผลิตและ Lead Time ตามลำดับ

วิธีการนำ Module ไปใช้งาน ให้เริ่มจาก Copy Module จากเซลล์ B6:P7 ไป Paste ต่อกันไปให้ครบตามจำนวนความสัมพันธ์ระหว่าง Child->Parent ที่มีอยู่ (B->A, C->A, D->B, E->B, E->C, F->C, G->F, D->F) จากนั้นให้บันทึกค่าใหม่ลงไปในเซลล์รับตัวแปรใน Column B, C, E, และ F ของแต่ละ Module ให้ครบถ้วนถูกต้องตามที่กำหนดไว้ในโครงสร้างผลิตภัณฑ์ จะพบว่าพื้นที่ตารางใน Column H:P จะคำนวณจำนวนวัสดุที่ต้องการและกำหนดเวลาที่ต้องเริ่มสั่งผลิตเพื่อให้ได้สินค้าหรือวัสดุเสร็จตามกำหนดให้ทันที

หลักการคำนวณที่ใช้ เกิดจากการหาตำแหน่งเลขที่ row สุดท้ายของแต่ละ Module (Last Row of Module) ใน Column D ให้ได้ก่อน จากนั้นจึงใช้เลขเลขที่ row สุดท้ายของแต่ละ Module สำหรับดึงตัวเลขการสั่งผลิตจาก Module ของ Parent ไปใช้ในการสั่งผลิตของ Module ที่เป็น Child

เพื่อให้ให้เห็นลำดับการคำนวณได้ชัดเจนยิ่งขึ้น ขอให้ดู Row 14 ซึ่งเป็นการสั่งผลิตวัสดุ E ให้กับวัสดุ C พอคุณบันทึกชื่อวัสดุ C ลงไปในเซลล์ C14 จะพบว่าในเซลล์ D14 คำนวณหาเลข 9 มาให้ ซึ่งเกิดจากสูตร =MATCH(C14,B\$1:B\$1001,0)+1 โดยสูตร Match จะนำชื่อ Parent C ไปเทียบหาตำแหน่งจากพื้นที่ B1:B1001 พบว่าชื่อ C อยู่ในลำดับที่ 8 จากนั้นนำไปบวก 1 ให้กลายเป็นเลขที่ Row 9 ซึ่งเป็นตำแหน่งเลขที่ row สุดท้ายของ Module (Last Row of Module) ที่มี Child ชื่อ C

พอได้เลข 9 ในเซลล์ D14 แล้ว จากนั้นจึงใช้สูตรดึงตัวเลข 150 จากเซลล์ N9 ส่งต่อมาใช้ที่เซลล์ N14 โดยใช้สูตร =INDIRECT(ADDRESS(\$D14,COLUMN()))*\$E14 โดยในการผลิตวัสดุ C จำนวน 1 หน่วย ต้องอาศัยวัสดุ E จำนวน 2 หน่วย (เซลล์ E14 = 2) จึงต้องสั่งผลิต E จำนวน 300 หน่วย (=150x2) ส่วนวัสดุ E นั้นกว่าจะผลิตเสร็จก็ต้องใช้ระยะเวลา (LT) 2 วัน ทำให้ต้องสั่งผลิตหรือจัดหาวัสดุ E ล่วงหน้าตั้งแต่วันที่ 5 จึงจะพร้อมต่อการนำไปใช้ผลิตต่อไปในวันที่ 7

ส่วนของสูตร =INDIRECT(ADDRESS(\$D14,COLUMN())) ในเซลล์ N14 ช่วยดึงค่ามาจากเซลล์ที่ทราบตำแหน่ง Row และ Column โดยที่ D14 เป็นเลขที่ Row ส่วนสูตร Column() จะหาเลขที่ Column N ณ ตำแหน่งของเซลล์สูตร จึงทำให้เกิดสูตร Address(9,14) คืนค่าออกมาเป็นตำแหน่งเซลล์ \$N\$9 จากนั้นสูตร Indirect("\$N\$9") จึงดึงค่าจากเซลล์ N9 มาให้ในที่สุด

ตัวอย่างจากชีทชื่อ DynamicCompound นี้เป็นการคำนวณอย่างง่ายเพื่อแสดงหัวใจของการเชื่อมโยงตัวเลขการสั่งผลิต ณ กำหนดเวลาที่ต้องการของแต่ละ Module เข้าด้วยกัน แต่ในการผลิตจริงยังต้องคำนึงถึงตัวเลขสินค้าต้นงวดปลายงวดและตัวเลขอีกมากมายที่เป็นเงื่อนไขเกี่ยวข้องกับการผลิตแต่ละครั้ง จึงขอให้ใช้ Module ที่สร้างไว้ในชีทชื่อ MRPModule แทน

	E	C	D	E	F	G	H	I	J	K	L	M	N	O	P
3															
5															
6															
7															
8															
9															
10															
11															
12															
13															
14															
15															
16															
17															
18															
19															
20															
21															
22															
23															
24															
25															
26															
27															
28															
29															
30															
31															

วิธีนำ Module ไปใช้ก็ใช้วิธี Copy ไป Paste เข้าตามโครงสร้างผลิตภัณฑ์เช่นเดียวกัน โดยโครงสร้างตารางคำนวณที่สร้างขึ้นแบบสมบูรณ์นี้ Child แต่ละตัว สามารถสั่งผลิตเป็นวัสดุให้ Parent ได้ 3 ตัว (เซลล์ D20:D22) ทำให้ไม่จำเป็นต้องเตรียม Module ของ Child ซ้ำกันเช่นตัวอย่างที่ผ่านมา

จากนั้นเมื่อนำตัวอย่างการทำ MRP จากหนังสือ ระบบการวางแผนและควบคุมการผลิต โดยรศ. พิกพ ลลิตาภรณ์ สำนักพิมพ์ของสมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น) หน้า 136 มาวางแผนความต้องการวัสดุ จะเกิดเป็น Compound Module ที่คำนวณร่วมกัน

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
3	MRP Module Complete																				
4	Item	Parent	Last Row	Quantity	Period																
5	A				Required Quantity of A	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
6										100			150								
7																					
8																					
9																					
10																					
11					Gross Requirements of A					100			150								
12					On Hand : Beginning of A					0	0	0	0	0	0	0	0	0	0	0	0
13					Scheduled Receipts of A					0	0	0	0	0	0	0	0	0	0	0	0
14					Available of A					0	0	0	0	0	0	0	0	0	0	0	0
15					On Hand : Ending of A					0	0	0	0	0	0	0	0	0	0	0	0
16					Net Requirements of A					100			150								
17					Planned Order Receipts of A					100			150								
18					Planned Order Releases of A					100			150								
19																					
20	Item	Parent	Last Row	Quantity	Period																
21	B	A	17	1	Required Quantity of B	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
22	F	87	1		Required Quantity of B					100			150								
23					Required Quantity of B					130			125								
24					Ordinary Required Quantity of B																
25					Gross Requirements of B					230			275								
26					On Hand : Beginning of B					150	150	280	50	50	10	10	10	10	10	10	10
27					Scheduled Receipts of B					0	130	0	0	0	0	0	0	0	0	0	0
28					Available of B					140	270	270	40	40	0	0	0	0	0	0	0
29					On Hand : Ending of B					150	280	50	50	10	10	10	10	10	10	10	10
30					Net Requirements of B								235								
31					Planned Order Receipts of B								235								
32					Planned Order Releases of B								235								
33																					
34	Item	Parent	Last Row	Quantity	Period																
35	C	A	17	2	Required Quantity of C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
36	B	31	2		Required Quantity of C					200			300								
37					Required Quantity of C					470											
38					Ordinary Required Quantity of C					10	10	10	10	10	10	10	10	10	10	10	10
39					Gross Requirements of C					10	10	10	10	10	10	10	10	10	10	10	10
40					On Hand : Beginning of C					20	260	250	40	30	20	20	20	20	20	20	20
41					Scheduled Receipts of C					250	0	0	0	0	0	0	0	0	0	0	0
42					Available of C					250	240	230	10	0	0	0	0	0	0	0	0
43					On Hand : Ending of C					260	250	40	30	20	20	20	20	20	20	20	20
44					Net Requirements of C								470	310	10	10	10	10	10	10	10
45					Planned Order Receipts of C								470	310	10	10	10	10	10	10	10
46					Planned Order Releases of C								10	10	10	10	10	10	10	10	10
47																					
48	Item	Parent	Last Row	Quantity	Period																
49	D	A	27	2	Required Quantity of D	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
50	C	45	3		Required Quantity of D					200			300								
51	F	87	1		Required Quantity of D					1410	930	30	30	30	30	30	30	30	30	30	30
52					Ordinary Required Quantity of D					130			125								
53					Gross Requirements of D					1740	930	30	455	30	30	30	30	30	30	30	30
54					On Hand : Beginning of D					700	700	700	60	60	60	60	60	60	60	60	60
55					Scheduled Receipts of D					0	0	0	0	0	0	0	0	0	0	0	0
56					Available of D					640	640	640	0	0	0	0	0	0	0	0	0
57					On Hand : Ending of D					700	700	60	60	60	60	60	60	60	60	60	60
58					Net Requirements of D								1100	930	30	455	30	30	30	30	30
59					Planned Order Receipts of D					1100	930	30	455	30	30	30	30	30	30	30	30
60					Planned Order Releases of D								30	30	30	30	30	30	30	30	30
61																					
62	Item	Parent	Last Row	Quantity	Period																
63	E	B	31	3	Required Quantity of E	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
64	F	45	1		Required Quantity of E					470	310	10	10	10	10	10	10	10	10	10	10
65					Required Quantity of E					130			125								
66					Ordinary Required Quantity of E																
67					Gross Requirements of E					600	210	715	135	10	10	10	10	10	10	10	10
68					On Hand : Beginning of E					60	60	0	0	0	0	0	0	0	0	0	0

ตัวอย่างในแฟ้ม CompoundModuleMRP(Ans).xls นอกเหนือจากแสดงการใช้ Compound Module แบบซับซ้อนโดยนำโครงสร้างผลิตภัณฑ์ 2 โครงสร้างมาใช้ร่วมกันแล้ว ยังแสดงวิธีการสร้างตารางฐานข้อมูลเพื่อเก็บค่าตัวแปร แล้วใช้สูตรดึงค่าจากตารางตัวแปรไปยังเซลล์รับค่าใน Module ให้เองอีกด้วยเพื่อช่วยอำนวยความสะดวกในการกรอกข้อมูลและลดโอกาสผิดพลาดจากการกรอกข้อมูลผิดช่อง จากนั้นเมื่อได้ผลลัพธ์ที่คำนวณใน Compound Module เสร็จเรียบร้อยก็ยังแสดงวิธีใช้สูตรดึงผลลัพธ์ไปสรุปเพื่อเปรียบเทียบกับจำนวนการผลิตที่เกิดขึ้นจริงที่อาจแตกต่างจากแผนการผลิตที่วางไว้

บทส่งท้าย

การออกแบบตารางคำนวณที่นำมาอธิบายนี้ น่าจะเป็นการจุดประกายความคิดให้ผู้ใช้ Excel สามารถสร้างสรรค์งานได้หลายหลายรูปแบบมากขึ้น ผมเชื่อว่าหลักการหรือแนวความคิดที่ได้ให้ไปแล้วย่อมเป็นสิ่งที่สำคัญมากกว่าตัวอย่างที่มีประโยชน์จำกัดแค่ใช้กับการแก้ไขปัญหาหนึ่งเสร็จแล้วก็แล้วกันไป ขอเพียงสามารถนำหลักการหรือแนวความคิดไปพัฒนาต่อ ย่อมเป็นแนวทางให้สามารถใช้ Excel แบบคอมพิวเตอร์ สามารถนำ Excel ไปประยุกต์ใช้กับงานที่หลากหลาย คนคิดว่าทำไมไม่ได้มาก่อน หากทางสร้างตารางคำนวณที่สะดวกต่อการนำมาใช้งานซ้ำแล้วซ้ำอีกง่ายต่อการแก้ไข และที่สำคัญคือเป็นหลักการที่ง่ายต่อความเข้าใจ