

Google Apps Script

เชอร์วิส Spreadsheet

25 มิ.ย. 2020

เรียบเรียงโดย
วสันต์ คุณดิลกเศวต

พิมพ์ครั้งที่ 1 : 3 ธ.ค. 2019

พิมพ์ครั้งที่ 2 : 2 เม.ย. 2020

พิมพ์ครั้งที่ 3 : 25 มิ.ย. 2020

Google Apps Script เชอร์วิส Spreadsheet

เรียบเรียงโดย
วสันต์ คุณดิลกเศวต

wasankds@gmail.com
Line ID : wasankds
08-1459-8343
www.poeclub.org

สารบัญ

สารบัญ.....	4
คำนำ.....	12
บทที่ 1 การบันทึก Macro ใน Google Sheets.....	15
1.1. การบันทึก Macro ใน Google Sheet	16
1.1.ก.) ขั้นตอนการบันทึก Macro (16)	
1.1.ข.) การเรียกใช้ Macro (17)	
1.1.ค.) ดูโค้ดที่ Macro สร้าง (17)	
1.2. Macro แบบ Absolute และ Relative	18
1.2.ก.) Macro จัดรูปแบบเซลล์บันทึกแบบ Absolute และ Relative (18)	
1.2.ข.) Macro บันทึก Alternating Colors (19)	
บทที่ 2 เรียกใช้เซอร์วิส Spreadsheet.....	21
2.1. เรียกใช้เซอร์วิส Spreadsheet	22
2.2. จับไฟล์ที่แอคทีฟ	22
2.2.ก.) getActiveSheet() (22)	
2.2.ข.) getUrl(), getId() และ getName() (23)	
2.2.ค.) getName() (23)	
2.3. flush()	24
บทที่ 3 ไฟล์.....	25
3.1. การเปิดไฟล์	26
3.1.ก.) openById() และ openByUrl() (26)	
3.1.ข.) open() (26)	
3.2. การสร้างไฟล์	27
3.3. หน้าต่างแบบต่างๆ	27
3.3.ก.) toast() (27)	
3.3.ข.) show() (28)	

บทที่ 4 ขี๊	29
4.1. ขี๊ขี๊ต่อจากไฟล์	30
4.2. getActiveSheet()	30
4.3. getSheets()	30
4.4. getSheetByName()	30
4.5. insertSheet()	31
4.6. deleteSheet()	32
4.7. activate()	33
4.8. getParent()	33
4.9. การลั้งขี๊	33
4.9.ก.) clear() (33)	
4.9.ข.) clearContents() และ clearFormats() (34)	
บทที่ 5 การจัดการแถว หรือคอลัมน์	35
5.1. การแทรกแถว	36
5.2. การลบแถว	37
บทที่ 6 เร็นจ์	39
6.1. ขี๊เร็นจ์	40
6.2. getRange()	40
6.2.ก.) getRange(row, column) (40)	
6.2.ข.) getRange(row, column, numRows) (40)	
6.2.ค.) getRange(row, column, numRows, numColumns) (41)	
6.2.ง.) getRange(a1Notation) (41)	
6.3. getLastRow() และ getLastColumn()	41
6.4. getA1Notation()	42
6.5. getCell()	42
6.6. getDataRegion()	43
6.6.ก.) getDataRegion() (43)	
6.6.ข.) getDataRegion(dimension) (43)	
6.7. getDataRange()	44
6.8. getValue() และ getValues()	46
6.9. getDisplayValue() และ getDisplayValues()	47

6.10. จับเร้นจจากชื่อเร้นจ (Named ranges)	47
6.10.ก.) getRangeByName() (47)	
6.10.ข.) getNamedRanges() (48)	
6.10.ค.) setNamedRange() (49)	
6.10.ง.) removeNamedRange() (50)	
6.10.จ.) setRange() และ setName() (50)	
6.10.ฉ.) remove() (51)	
6.11. offset()	52
6.12. setValue() และ setValues()	53
6.13. นำคอลัมน์มาคำนวณ แล้ววางผลลัพธ์ไว้คอลัมน์ข้างๆ	54
6.14. กรอกสูตรลงเซลล์	55
6.14.ก.) setFormula() (55)	
6.14.ข.) setFormulaR1C1() (56)	
6.15. getActiveRange()	57
6.16. getRow(), getColumn(), getNumRows(), getNumColumns()	58
6.17. การล้างเนื้อหา และ การล้างรูปแบบ	59
6.17.ก.) clearContents() (59)	
6.17.ข.) clearFormats() (59)	
6.18. appendRow()	59
6.19. โน้ต	60
6.19.ก.) setNote() และ setNotes() (60)	
6.19.ข.) clearNote() (61)	
บทที่ 7 กลุ่มของเร้นจ.....	63
7.1. getRanges()	64
7.2. getActiveRangeList()	64
7.3. getRangeList()	65
7.4. getSelection() และ getCurrentCell()	65
7.5. setActiveRange()	67
7.6. setCurrentCell()	67
7.7. activate()	68
7.8. getNextDataCell() และ getNextDataRange()	68
7.8.ก.) getNextDataCell() (68)	
7.8.ข.) getNextDataRange() (69)	

บทที่ 8 รูปแบบ.....	71
8.1. setNumberFormat()	72
8.1.ก.) setNumberFormat() (72)	
8.1.ข.) setNumberFormats() (72)	
8.2. เส้นขอบ	73
8.3. แบ็คกราวนด์	74
8.3.ก.) setBackground() และ setBackgrounds() (74)	
8.3.ข.) setBackgroundRGB() (75)	
8.3.ค.) ลิงค์เครื่องมือช่วยเลือกสี (76)	
8.3.ง.) setBackgroundObject() (76)	
8.4. กำหนดคุณลักษณะให้กับเร้นจ์	78
8.4.ก.) setFontFamily() และ setFontFamilies() (78)	
8.4.ข.) setFontSize() และ setFontSizes() (78)	
8.4.ค.) setFontWeight() และ setFontWeights() (79)	
8.4.ง.) setFontStyle(fontStyle) และ setFontStyles(fontStyles) (80)	
8.4.จ.) setFontColor() และ (80)	
8.4.ฉ.) setHorizontalAlignment() (81)	
8.4.ช.) setVerticalAlignment() (81)	
8.4.ซ.) setFontLine(fontLine)(82)	
8.5. ตัวอย่างการกำหนดรูปแบบหลายอย่างให้กับเร้นจ์	83
8.6. Text wrapping	84
8.6.ก.) getWrapStrategy() และ getWrapStrategies()(84)	
8.6.ข.) Enum WrapStrategy (84)	
8.6.ค.) setWrapStrategy() และ setWrapStrategies()(85)	
8.6.ง.) setWrap(isWrapEnabled) และ setWraps(isWrapEnabled) (85)	
8.7. Text Style	86
8.7.ก.) newTextStyle() และ setTextStyle() (86)	
8.7.ข.) เมธอดที่ใช้กำหนดรูปแบบของข้อความ (87)	
8.8. Rich Text Value	88
บทที่ 9 การก๊อปปี้.....	91
9.1. copyTo() คลาส Range	92
9.1.ก.) copyTo(destination) (92)	
9.1.ข.) copyTo(destination, options) (93)	
9.1.ค.) copyTo(destination, copyPasteType, transposed)(94)	
9.2. ก๊อปปี้รูปแบบไปที่เร้นจ์	95
9.2.ก.) copyFormatToRange() (95)	
9.3. ก๊อปปี้ไฟล์ Google Sheets ด้วย copy()	96
9.3.ก.) copy() (96)	

บทที่ 10 Banding หรือ Alternating Colors.....	99
10.1. applyRowBanding()	100
10.2. จับ Banding	101
10.3. การลบ Banding หรือ Alternating Colors	101
บทที่ 11 Data validation.....	103
11.1. สร้าง Data validation ให้กับเซลล์ในเรนจ์	104
11.1.ก.) คลาสและเมธอดที่สำคัญ (104)	
11.1.ข.) ตัวอย่างที่ 1 (105)	
11.1.ค.) ตัวอย่างที่ 2 (105)	
11.2. ดูและอัปเดต Data validation	106
11.2.ก.) getDataValidation() (106)	
11.2.ข.) getDataValidations() - มี s (109)	
11.3. Checkboxes	111
11.3.ก.) insertCheckboxes() (111)	
11.3.ข.) requireCheckbox() (112)	
บทที่ 12 การเรียง (Sorting).....	115
12.1. การเรียงข้อมูลโดยใช้เมธอด sort()	116
12.1.ก.) การเรียงข้อมูลในเรนจ์ (เมธอดในคลาส Range)(116)	
12.1.ข.) การเรียงข้อมูลในชีท (เมธอดในคลาส Sheet) (118)	
บทที่ 13 ฟิลเตอร์ (ตัวกรอง).....	119
13.1. คลาสที่เกี่ยวข้องกับฟิลเตอร์	120
13.2. การจับฟิลเตอร์	120
13.2.ก.) เมธอดที่เกี่ยวข้อง (120)	
13.2.ข.) ตัวอย่างที่ 1 : จับฟิลเตอร์ที่กรองแบบ Filter by values ไว้อย่างเดียว (121)	
13.2.ค.) ตัวอย่างที่ 2 : จับฟิลเตอร์ที่กรองแบบ Filter by values และ Filter by condition (122)	
13.3. การสร้างฟิลเตอร์	123
13.3.ก.) เมธอดที่เกี่ยวข้อง (123)	
13.3.ข.) ตัวอย่างที่ 1 ใส่ฟิลเตอร์ให้กับเรนจ์ (124)	
13.3.ค.) ตัวอย่างที่ 2 ใส่ฟิลเตอร์ให้กับเรนจ์และกรองแบบ Filter by values (125)	
13.4. การแก้ไขกฎการกรอง	125
13.4.ก.) ตัวอย่างที่ 1 (125)	
13.4.ข.) ตัวอย่างที่ 2 ฟิลเตอร์โดยใช้ค่าในเซลล์(Dropdown) (126)	
13.5. การถอดกฎการกรองหรือถอดฟิลเตอร์	127
13.5.ก.) remove() - ถอดกฎการกรอง (127)	
13.5.ข.) removeColumnFilterCriteria() - ถอดคอลัมน์ที่มีการกรอง (127)	

13.6. การเรียงข้อมูลผลการกรอง	127
13.7. withCriteria()	128
13.7.ก.) เมล็ดที่เกี่ยวข้อง (128)	
13.8. กฎการกรองแบบ Filter by condition	129
13.8.ก.) ตัวอย่างที่ 1 – whenTextContains() (129)	
13.8.ข.) ตัวอย่างที่ 2 – whenDateEqualTo() (130)	
บทที่ 14 การสร้าง Chart.....	131
14.1. สร้าง Bar Chart	132
14.1.ก.) ตัวอย่างการสร้าง Bar Chart - คอลัมน์ Cost (132)	
14.1.ข.) ตัวอย่างการสร้าง Bar Chart- คอลัมน์ Total (133)	
14.2. newChart() และ build()	134
14.3. setChartType()	134
14.4. addRange()	135
14.5. setNumHeaders()	135
14.6. setPosition()	135
14.7. setMergeStrategy()	136
14.8. insertChart(), updateChart() และ getChart()	137
14.8.ก.) ตัวอย่างที่ 1 (137)	
14.8.ข.) ตัวอย่างที่ 2 (138)	
14.9. จัปชาร์ตเป็นภาพ	138
14.9.ก.) ส่งเมลล์ (138)	
14.9.ข.) ใส่ไว้ในชีทอื่น (139)	
บทที่ 15 Drawing และ Over Grid Image.....	141
15.1. Drawing(รูปภาพ)	142
15.2. Over Grid Images	143
บทที่ 16 User Interfaces.....	145
16.1. การสร้างเมนู	146
16.1.ก.) คลาสและเมล็ดที่สำคัญในการสร้างเมนู (146)	
16.2. alert()	148
16.3. prompt()	149
16.4. Custom dialogs (Modal Dialog)	151
16.4.ก.) ตัวอย่าง : แสดง Dialog (151)	
16.5. Custom sidebars	152
บทที่ 17 คลาส Browser.....	155
17.1. msgBox()	156
17.2. inputBox()	157

บทที่ 18 การประยุกต์ใช้งาน.....	159
18.1. การกรองแวงซ้ำทั้ง	160
18.2. Data Lookup	161
บทที่ 19 วันและเวลา.....	163
19.1. สร้างวัตถุวันที่และเวลา	164
19.1.ก.) new Date() (164)	
19.1.ข.) +new Date() (165)	
19.2. การทำงานกับวันที่และเวลา	165
19.3. การคำนวณเกี่ยวกับเวลา	167
19.4. setHours()	168
บทที่ 20 ฟังก์ชัน, เมธอด และ คุณสมบัติ ที่น่าสนใจ.....	169
20.1. toString()	170
20.2. String()	170
20.3. toLowerCase() และ toUpperCase()	171
20.4. PI	172
20.5. isNaN()	172
20.6. Number()	173
20.7. split()	173
20.8. parseInt()	174
20.9. charAt()	175
20.10. charCodeAt()	175
20.11. fromCharCode()	176
20.12. trim()	176
20.13. substr() และ substring()	177
20.14. search()	178
20.15. replace()	179
20.16. match()	180
20.17. includes() - Chrome V8	180
20.18. indexOf()	181
20.19. lastIndexOf()	181
20.20. toFixed	182
20.21. toPrecision()	182
20.22. valueOf()	183

บทที่ 21 Script Properties.....	185
21.1. Script Properties คืออะไร ?	186
21.2. การสร้าง Script Properties แบบกรอกตัวเอง	186
21.3. การจับ Script Properties	186
21.4. การสร้าง Script Properties โดยการเขียนสคริปต์	187
21.4.ก.) setProperty() (187)	
21.4.ข.) setProperties() (188)	
21.5. deleteAllProperties() และ deleteProperty()	189
บทที่ 22 Regular Expression.....	191
22.1. Regular Expression คืออะไร ?	192
22.1.ก.) Regular Expression คืออะไร ? (192)	
22.1.ข.) การใช้งาน Regular Expression ใน Apps Script (192)	
22.2. Regular Expression Systax ที่สำคัญ	193
22.2.ก.) Regular Expression Systax ที่สำคัญ (193)	
22.2.ข.) การผสม Regular Expression ที่น่าสนใจ (194)	
22.3. การค้นหาและแทนที่โดยใช้ Regular Expression	197
22.4. RegExp()	197
22.5. exec()	198
22.6. test()	199

คำนำ

หนังสือเล่มนี้ เป็นหนึ่งในชุด **การเขียนโปรแกรม Google Apps Script** โดยในเล่มนี้อธิบายการใช้งาน **เชอร์วิส Spreadsheet** ก็คือ เชอร์วิสสำหรับการใช้งาน Google Sheets นั่นเอง

นอกเหนือจากเนื้อหาดังกล่าว อะไรที่สำคัญที่เกี่ยวข้องก็อธิบายไว้ด้วย เช่น Regular Expression, วันที่และเวลา เป็นต้น

ผู้เขียน เขียนหนังสือเล่มนี้ จุดประสงค์ดั้งเดิม ก็คือ **เก็บไว้อ่านเอง**

เมื่อผู้เขียนศึกษาเรื่องอะไร ก็จะไปเรียนรู้จากสื่อออนไลน์ในอินเทอร์เน็ต ทั้งคอร์สออนไลน์ วิดีโอ หรือ เอกสาร ทั้งในแบบฟรีและเสียเงิน

แต่ยุคนี้ เราต้องเรียนรู้อะไรให้เร็ว โดยเฉพาะเรื่องของ IT ผู้เขียนจึงตั้งใจจะดูวิดีโอเพียงรอบเดียว จึงดูไป พิมพ์สรุปไปด้วย เวลาจำอะไรไม่ได้ มาดูจากที่พิมพ์สรุปไว้ ง่ายกว่าการไปย้อนดูจากวิดีโอ นอกจากนี้ก็นำมาทบทวนได้ง่าย ในอนาคตสามารถเพิ่มเติมเสริมแต่งเนื้อหาได้เรื่อยๆด้วย

หากเนื้อหา ไม่ละเอียด อ่านแล้วไม่เข้าใจ หรือ ไม่ได้แปลเป็นไทยในบางส่วน ต้องขอภัยไว้ ณ ที่นี้ เพราะผู้เขียนเขียนโปรแกรมไป(ตามที่ลูกคำว่าจ้าง) แล้วก็ไปศึกษาส่วนที่เกี่ยวข้อง จากนั้นก็เขียนลงในหนังสือเก็บไว้ บางทีงานก็เร่งมาก จนไม่มีเวลาเขียนอย่างละเอียด

หนังสือเล่มนี้ ความตั้งใจดั้งเดิมของผู้เขียน ก็คือ **ตั้งใจเก็บไว้อ่านเองคนเดียว** อย่างไรก็ตาม อดสำหรับพิมพ์ไว้เป็นหนังสือแล้ว จะเก็บไว้คนเดียวก็รู้สึกเสียดาย ฉะนั้น ผู้เขียนจึงนำออกมาแบ่งปัน

เนื่องด้วย ผู้เขียนให้ความสำคัญกับประเด็นด้านลิขสิทธิ์มาก ฉะนั้นจึงขอแจ้งไว้ ณ ที่นี้ ตั้งแต่ต้นก็คือ

1. เนื้อหาในหนังสือ ผู้เขียนรวบรวมมาจากแหล่งต่างๆในอินเทอร์เน็ต โดยจะพยายามให้มากที่สุด ที่จะบอกลิงค์หรือแหล่งที่มาในแต่ละหัวข้อ เพราะหนังสือเล่มนี้ถูกเขียนไว้นานแล้ว บางเรื่องก็ลืมก็อปปีลิงค์มาแปะไว้
2. ผู้เขียนรวบรวมเนื้อหาจากแหล่งต่างๆ และเขียนเพิ่มเติมไปด้วย
3. หนังสือเล่มนี้แจกฟรี ผู้เขียนไม่มีรายได้จากหนังสือเล่มนี้

หนังสือเล่มนี้ยังไม่จบเสียทีเดียว หากผู้เขียนว่าง จะมาเขียนเพิ่มเติมเรื่อยๆ ให้ดูเวอร์ชันตามวันที่ที่ปล่อยหนังสือ

วสันต์ คุณดิลกเสวต
wasankds@gmail.com
081-459-8343
Line ID : wasankds

บทที่ 1

การบันทึก Macro ใน Google Sheets



ที่มา

คอร์สออนไลน์ Apps Script Blastoff! (เรียนฟรี)

<https://courses.benlcollins.com/courses/435404/lectures/6708124>

1.1. การบันทึก Macro ใน Google Sheet

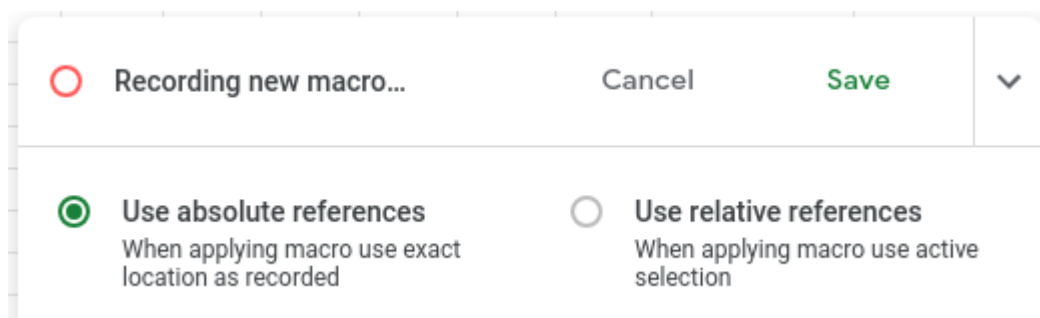
1.1.ก.) ขั้นตอนการบันทึก Macro

Macro เป็นโปรแกรมเล็กๆ ที่สร้างโดยการบันทึกขั้นตอนการทำงาน เช่น พิมพ์ คลิก เป็นต้น จากนั้นระบบจะสร้างเป็นโค้ดของ Google Apps Script ให้อัตโนมัติ

ขั้นตอนการบันทึก Macro จัดรูปแบบเซลล์ - [ใช้ Absolute](#)

1. ไปที่เมนู Tools → Macros → Record จะปรากฏหน้าต่างตามภาพ

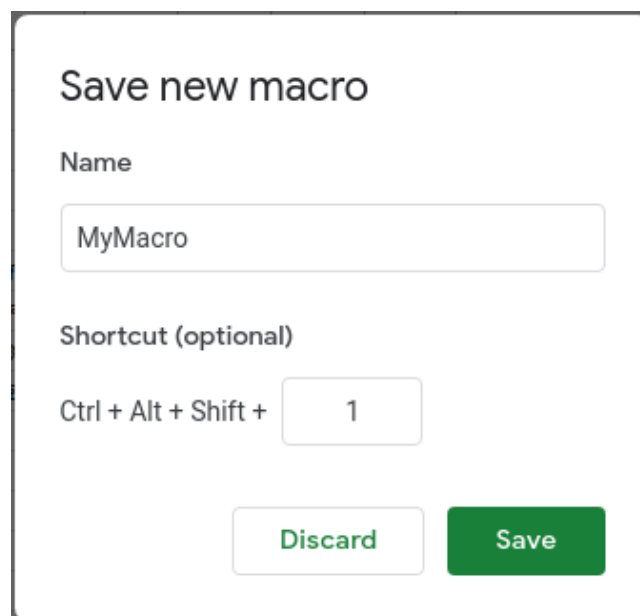
1.1 เลือก Use absolute references



2. ทำงาน - โดยทำคำสั่งต่างๆ เช่น คลิกเมาส์ พิมพ์ เลือกจากเมนู เป็นต้น ****

3. คลิกที่ Save เมื่อจบการทำงาน - จะปรากฏหน้าต่างมาให้บันทึก Macro

4. ที่หน้าต่าง Save new macro - บันทึกมาโคร



4.1 ที่ช่อง Name - ตั้งชื่อ Macro

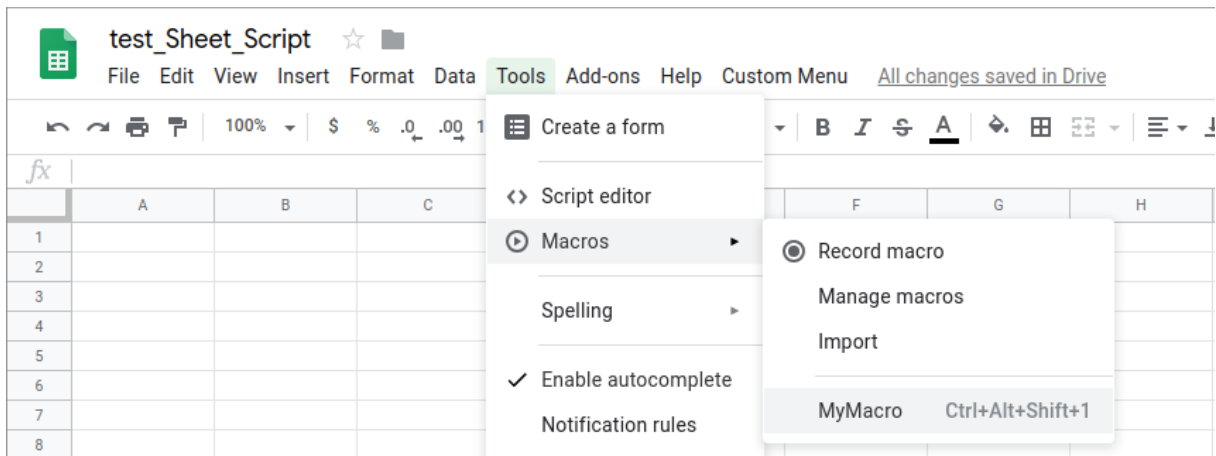
4.2 ที่ช่อง Shortcut กำหนดคีย์ลัดให้กับการเรียกใช้ Macro เช่น <Ctrl><Alt><Shift><1> ถึง <9>

4.3 คลิกปุ่ม Save

1.1.ข.) การเรียกใช้ Macro

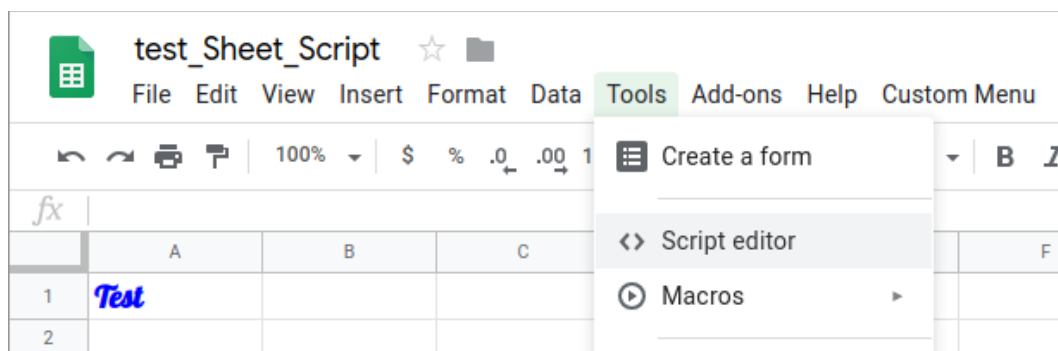
หลังจากบันทึก Macro แล้ว เราสามารถเรียกใช้ Macro ที่บันทึกไว้ได้ที่

เมนู Tools → Macros → (ชื่อ Macro)

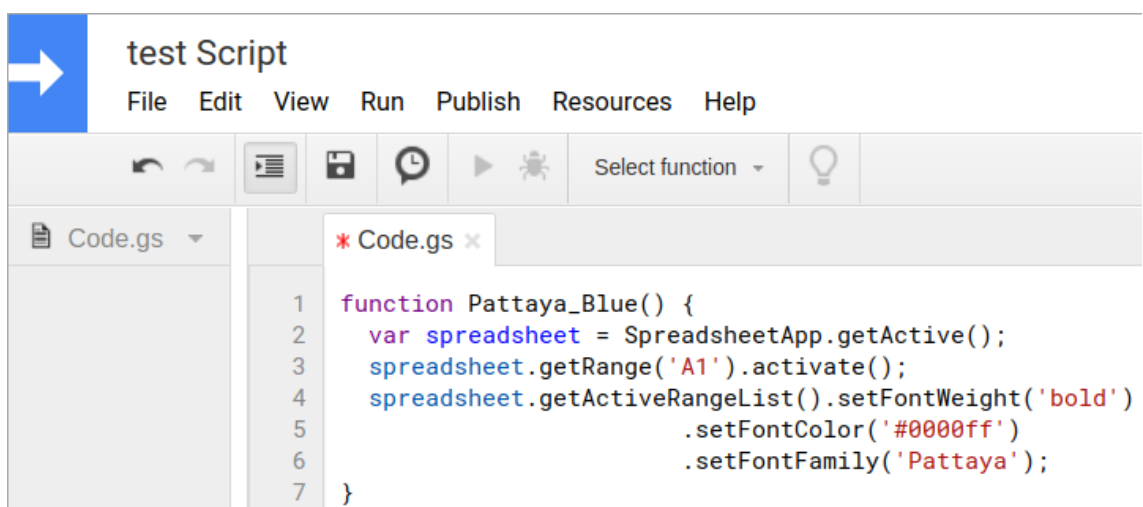


1.1.ค.) ดูโค้ดที่ Macro สร้าง

ที่หน้าแรก Google Sheet ไปที่เมนู Tools → Script Editor ตามภาพ



จะปรากฏหน้าโปรเจ็ค Google Apps Script แสดงโค้ดต่างๆที่ถูกสร้างขึ้น ตามภาพ



1.2. Macro แบบ Absolute และ Relative

(หากยังไม่เข้าใจเรื่องโค้ด ให้ข้ามไปอ่านบทที่ 2 ก่อน)

1.2.ก.) Macro จัดรูปแบบเซลล์บันทึกแบบ Absolute และ Relative

2 โค้ดต่อไปนี้เป็นขั้นตอนการบันทึก Macro เหมือนกัน แต่ต่างกันที่เลือก **Absolute** และ **Relative** จึงได้โค้ดและผลการรันต่างกัน

โค้ดต่อไปนี้เป็นได้จากการบันทึก Macro แบบ Absolute Reference - เมื่อรัน Macro จะจัดรูปแบบเซลล์ A1 เท่านั้น

```
// เมื่อรัน Macro จะจัดรูปแบบเซลล์ A1 เท่านั้น  
  
function Pattaya_Blue() {  
  var spreadsheet = SpreadsheetApp.getActive();  
  spreadsheet.getRange('A1').activate();  
  spreadsheet.getActiveRangeList().setFontWeight('bold')  
                                     .setFontColor('#0000ff')  
                                     .setFontFamily('Pattaya');  
}
```

โค้ดต่อไปนี้เป็นได้จากการบันทึก Macro แบบ Relative Reference - เมื่อรัน Macro จะจัดรูปแบบเซลล์ที่เลือก

```
// เมื่อรัน Macro จะจัดรูปแบบเซลล์ที่เลือก  
  
function Pattaya_RED_18() {  
  var spreadsheet = SpreadsheetApp.getActive();  
  spreadsheet.getActiveRangeList().setFontColor('#ff0000')  
                                     .setFontSize(18)  
                                     .setFontStyle('italic')  
}
```

1.2.ข.) Macro บันทึก Alternating Colors

บันทึก การจัดรูปแบบตารางข้อมูลด้วย Alternating Colors โดยบันทึกแบบ Relative Reference

```
// การบันทึก เราเลือกอาณาเขตของเซลล์ก่อน แล้วจึงใส่ Alternating Color
// *** แบบ Relative Reference ***

function TableAlterColor() {
  var spreadsheet = SpreadsheetApp.getActive();
  spreadsheet.getCurrentCell().offset(0, 0, 14, 7).activate(); // กำหนดขอบเขตเซลล์ชัดเจน
  spreadsheet.getActiveRange().applyRowBanding(SpreadsheetApp.BandingTheme.LIGHT_GREY);

  var banding = spreadsheet.getActiveRange().getBandings()[0];
  banding.setHeaderRowColor('#8bc34a')
    .setFirstRowColor('#ffffff')
    .setSecondRowColor('#eef7e3')
    .setFooterRowColor(null);
}
```

Macro ข้างต้น เวลาบันทึก ผลจาก Alternating Colors จะทำกับตารางข้อมูลขนาด 14x7 เสมอ สังเกตโค้ดที่บรรทัด `spreadsheet.getCurrentCell().offset(0, 0, 14, 7).activate();` ก็คือ 14 แถว 7 คอลัมน์ ฉะนั้น จึงไม่สามารถทำกับเร้นจ์ตามขนาดที่เราเลือกได้

มาลองดูการบันทึกแบบ Absolute Reference กันดูบ้าง

```
// บันทึก การจัดรูปแบบตารางข้อมูล โดยใช้ Alternating Color
// *** แบบ Absolute Reference ***

function TableAlterColor2ABS() {
  var spreadsheet = SpreadsheetApp.getActive();
  spreadsheet.getRange('A1:G14').activate();
  spreadsheet.getRange('A1:G14').applyRowBanding(SpreadsheetApp.BandingTheme.LIGHT_GREY);

  var banding = spreadsheet.getRange('A1:G14').getBandings()[0];
  banding.setHeaderRowColor('#e91d63')
    .setFirstRowColor('#ffffff')
    .setSecondRowColor('#fddce8')
    .setFooterRowColor(null);
}
```

Macro ข้างต้น เวลาบันทึก ผลจาก Alternating Color จะใส่ให้กับเร้นจ์ A1:G14 เสมอ ใช้งานไม่ได้อีกเหมือนกัน ต้องเข้าไปแก้ไขโค้ดเอง

การแก้ไขโค้ด

ไปที่เมนู Tools → Script Editor จากนั้นแก้ไขโค้ดเป็นดังนี้

โค้ดใส่ Alternating Colors เขียนสั้นๆตามด้านล่าง แค่นี้ก็ใช้ได้

```
function TableAlterColor2ABS() { // ลดบรรทัดโค้ด แต่ได้ผลเหมือนเดิม
  var spreadsheet = SpreadsheetApp.getActive();
  spreadsheet.getRange('A1:G14').activate();
  spreadsheet.getRange('A1:G14').applyRowBanding(SpreadsheetApp.BandingTheme.ORANGE);
}
```

อย่างไรก็ดี โค้ดข้างบน ใช้ได้เฉพาะกับเซลล์ที่กำหนดแบบตายตัวเท่านั้น ก็คือ **A1:G14** เท่านั้น

ฉะนั้นต้องแก้ ดังต่อไปนี้

```
// แบบที่ 1
// ใส่ Alternating Color กับเรนจ์ – เลือกเซลล์เดียวในตารางข้อมูล ใส่ให้กับทุกเซลล์ที่ติดกัน

function TableAlterColor2ABS-2() {
  var spreadsheet = SpreadsheetApp.getActive() ;
  spreadsheet.getDataRange().applyRowBanding(SpreadsheetApp.BandingTheme.ORANGE) ;
}

// แบบที่ 2
// ใส่ Alternating Color เฉพาะเซลล์ที่เลือก ***

function TableAlterColor2ABS-3() {
  var spreadsheet = SpreadsheetApp.getActive() ;
  spreadsheet.getActiveRange().applyRowBanding(SpreadsheetApp.BandingTheme.ORANGE) ;
}
```

บทที่ 2
เรียกใช้เซอร์วิส
Spreadsheet



2.1. เรียกใช้เซอร์วิส Spreadsheet

Spreadsheet Service

<https://developers.google.com/apps-script/reference/spreadsheet>

Extending Google Sheets

<https://developers.google.com/apps-script/guides/sheets>

เซอร์วิส Spreadsheet ใช้สร้าง เข้าถึงไฟล์ และแก้ไขข้อมูลต่างๆในไฟล์ Google Sheets

คลาส SpreadsheetApp

การเรียกใช้งานเซอร์วิส Spreadsheet เริ่มต้นจากคลาส SpreadsheetApp จากนั้นจึงจับคลาสย่อยต่อไปเรื่อยๆ จนกว่าจะถึงคลาสที่เราต้องการจะควบคุม

ตัวอย่าง - โค้ดต่อไปนี้ สร้างไฟล์ Google Sheets ขึ้นมา จากนั้นจับ Url และ Id ของไฟล์ออกมา Logs

ดู

```
// สร้างไฟล์ Google Sheets ชื่อ Finances ไว้ที่ Google Drive
var ssNew = SpreadsheetApp.create("Finances") ;

Logger.log(ssNew.getUrl()) ;
// พิมพ์ : URL ของไฟล์ - ผลลัพธ์เช่น
// https://docs.google.com/spreadsheets/d/1TjYroQfJs2QNfz9wqhwnSFZ4rHChK9y8Toh22b1_Do

Logger.log(ssNew.getId()) ;
// พิมพ์ ID ของไฟล์ - ผลลัพธ์เช่น
// 1TjYroQfJs2QNfz9wqhwnSFZ4rHChK9y8Toh22b1_Do
```

2.2. จับไฟล์ที่แอ็คทีฟ

2.2.ก.) getActiveSheet()

คลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet>

getActiveSheet() - เมธอดในคลาส SpreadsheetApp

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet-app#getactivespreadsheet>

getActiveSheet() ใช้จับไฟล์ Google Sheets โดยจะคืนค่ากลับมาเป็นคลาส Spreadsheet

getActiveSheet() ใช้ได้กับโปรเจ็ค Google Apps Script ที่เขียนฝังอยู่ในไฟล์ Google Sheets

หากเป็น Google Apps Script แบบ Stand alone จะคืนค่าเป็น **null**

ตัวอย่างการใช้งาน

```
// พิมพ์ URL ของไฟล์ Google Sheets ที่โปรเจ็ค Google Script ฝังอยู่
//
Logger.log(SpreadsheetApp.getActiveSheet().getUrl()) ;
```

2.2.ข.) getUrl(), getId() และ getName()

getId() - เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#getId>

ใช้จับ Id ของไฟล์ Google Apps โดยจะคืนค่ากลับมาเป็น String(ข้อความ)

getUrl() - เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#getUrl>

ใช้จับ Url ของไฟล์ Google Apps โดยจะคืนค่ากลับมาเป็น String(ข้อความ)

getName() - เมธอดในคลาส Spreadsheet

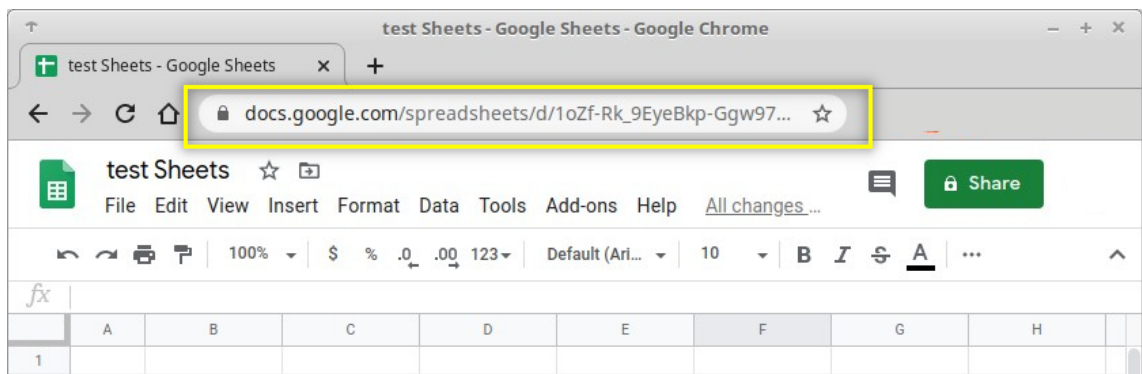
<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#getName>

ใช้จับ Url ชื่อไฟล์ Google Apps โดยจะคืนค่ากลับมาเป็น String(ข้อความ)

Url และ Id ของไฟล์

Url ของไฟล์ Google Sheets คือ ข้อความในช่องที่เราพิมพ์ชื่อเว็บลงไป ตัวอย่างตามภาพ

Id คือส่วนข้อความในบล็อคดีังต่อไปนี้ https://docs.google.com/spreadsheets/d/1oZf-Rk_9EyeBkp-Ggw97... /edit



2.2.ค.) getName()

getName()

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#getName>

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#getName>

getName() เป็นเมธอดในหลายคลาส เช่น คลาส Spreadsheet, คลาส Sheet เป็นต้น ใช้จับชื่อของวัตถุ เช่น ชื่อไฟล์ , ชื่อชีท เป็นต้น

```
var ss = SpreadsheetApp.getActive() ;  
var name = ss.getName() ;           // จับชื่อชีท  
  
Logger.log(typeof ss) ;             // พิมพ์ : Object  
Logger.log(typeof name) ;           // พิมพ์ : String  
Logger.log(name) ;                  // พิมพ์ : (ชื่อ Spreadsheet)
```

2.3. flush()

flush() - เมธอดในคลาส SpreadsheetApp

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet-app#flush>

โค้ดบางอย่าง ที่เราต้องการให้ปรากฏผลใน Google Sheets ขณะรัน แต่ว่า พอรันโค้ดจนจบแล้ว กลับไม่ปรากฏผลให้เห็นขณะรัน แต่เห็นผลตอนจบเลย

flush() ใช้สั่งให้ความเปลี่ยนแปลงขณะรันโค้ด ปรากฏใน Google Sheets

flush() มีข้อเสีย ที่ทำให้การประมวลผลทั้งหมดช้าลง แต่ก็มีข้อดีที่ เราสามารถเห็นความเปลี่ยนแปลงที่ Google Sheets ทำให้สัมผัสได้ว่าโค้ดกำลังทำงานอยู่

ผู้เขียนมักใช้ flush() กับอะไรที่ต้องรันเพียงครั้งเดียว เช่น การรัน Setup เพื่อติดตั้งระบบงาน เป็นต้น

ตัวอย่าง - โค้ดต่อไปนี้จะไม่มีบรรทัด SpreadsheetApp.flush() จะไม่เห็นความเปลี่ยนแปลงระหว่างรัน จะเห็นผลสุดท้ายเลย

```
var sheet = SpreadsheetApp.getActiveSheet() ;  
for (var i = 0 ; i < 20; i++) {  
  if ((i % 2) == 0) {  
    sheet.getRange('A1').setBackground('green') ;  
    sheet.getRange('B1').setBackground('red') ;  
  } else {  
    sheet.getRange('A1').setBackground('red') ;  
    sheet.getRange('B1').setBackground('green') ;  
  }  
  SpreadsheetApp.flush() ;  
}
```

ผล - เห็นสีพื้นหลังของเซลล์ A1 และ B1 สลับสีกัน

	A	B
1		

บทที่ 3

ไฟล์



3.1. การเปิดไฟล์

3.1.ก.) openById() และ openByUrl()

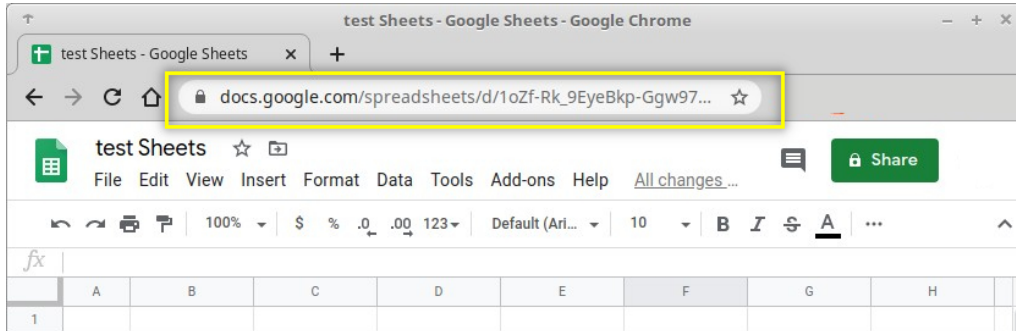
openById() - เมธอดในคลาส SpreadsheetApp

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet-app#openbyidid>

openByUrl() - เมธอดในคลาส SpreadsheetApp

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet-app#openbyurlurl>

openById() และ **openByUrl()** ใช้เปิดไฟล์โดยระบุ Id หรือ Url ของไฟล์ตามลำดับ



ตัวอย่าง โค้ดต่อไปนี้ เปิดไฟล์ Google Sheets ตาม URL ของไฟล์ที่ระบุ จากนั้น Logs ชื่อไฟล์ออกมาดู

```
var ss =  
SpreadsheetApp.openByUrl('https://docs.google.com/spreadsheets/d/abc1234567/edit') ;  
Logger.log(ss.getName()) ;
```

โค้ดถัดไป ให้ผลลัพธ์ เช่นเดียวกับโค้ดก่อนหน้า

```
var ss = SpreadsheetApp.openById("abc1234567") ;  
Logger.log(ss.getName()) ;
```

3.1.ข.) open()

คลาส File

<https://developers.google.com/apps-script/reference/drive/file.html>

ไฟล์ใน Google Drive ซึ่งสามารถเข้าถึงและถูกสร้างได้โดยใช้เซอร์วิส Drive

open(file)

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet-app#openfile>

เปิดไฟล์ Google Sheets ที่เป็นวัตถุไฟล์ โดยพารามิเตอร์ **file** เป็นวัตถุไฟล์

ตัวอย่าง

```
// จับไฟล์ที่ถูกมาร์คดาวใน Google Drive และเป็นไฟล์ Google Sheets
// จากนั้นเปิดไฟล์ Spreadsheet และ Logs ชื่อชี้แรกของไฟล์ออกมาดู
var files = DriveApp.searchFiles(
  'starred = true and mimeType = "' + MimeType.GOOGLE_SHEETS + '"');
while (files.hasNext()) {
  var spreadsheet = SpreadsheetApp.open(files.next());
  var sheet = spreadsheet.getSheets()[0];
  Logger.log(sheet.getName());
}
```

3.2. การสร้างไฟล์

create(name) – เมธอดในคลาส SpreadsheetApp

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet-app#createName>

create(name, rows, columns)

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet-app#createName,-rows,-columns>

create() ใช้สร้างไฟล์ Spreadsheet

ตัวอย่าง – เมื่อรันโค้ดจะสร้างไฟล์ Google Sheets ไว้ที่ My Drive (ที่ Root folder)

```
var ssNew = SpreadsheetApp.create("Finances");
Logger.log(ssNew.getUrl()); // Url ของไฟล์

var ssNew2 = SpreadsheetApp.create("Finances2",50,5);
Logger.log(ssNew2.getUrl()); // Url ของไฟล์
```

เมื่อสร้างไฟล์แล้ว จากนั้น สามารถสร้างชีตต่อได้ หากต้องการสร้างไฟล์ไว้ยังโฟลเดอร์แบบเจาะจง ต้องใช้เซอร์วิส Drive ร่วมด้วย

3.3. หน้าต่างแบบต่างๆ

3.3.ก.) toast()

toast(msg, title, timeoutSeconds) – เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#toastMsg,-title,-timeoutSeconds>

แสดงหน้าต่าง Popup ที่มุมขวาล่าง โดยกำหนดเวลาการเปิดไว้ได้

พารามิเตอร์

Name	Type	Description
msg	String	ข้อความที่จะแสดง
title	String	ไตเติ้ลของกรอบ Popup
timeoutSeconds	Number	เวลาที่จะเปิดทิ้งไว้ หน่วยเป็นวินาที ถ้าเว้นไว้ จะหมายถึง 5 วินาที ถ้าเป็นค่าลบ จะเปิดค้างไว้จนกว่าเราจะปิดเอง

ตัวอย่าง

```
// แสดง วินาที  
SpreadsheetApp.getActiveSpreadsheet().toast('Task started', 'Status', 3) ;
```

ผลลัพธ์ - ปรากฏหน้าต่าง Popup ที่มุมขวาล่างของ Spreadsheet ตามภาพ



3.3.ข.) show()

show(userInterface) – เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#showuserinterface>

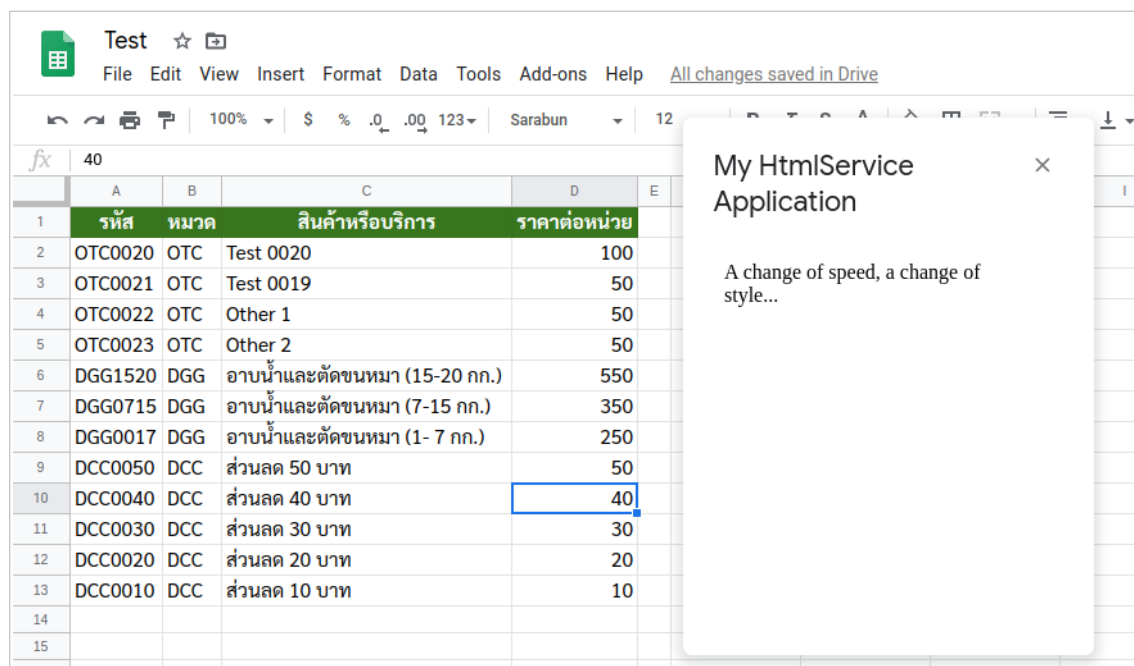
show() แสดงหน้าต่าง ที่เราสามารถสร้างองค์ประกอบเองได้ ใช้สร้างหน้าต่างรับค่าจากยูสเซอร์ได้ (ถ้า

รับหลายค่าสร้าง Sidebar ดีกว่า) พารามิเตอร์ **userInterface** ก็คือ วัตถุ HtmlOutput (โค้ด Html)

ตัวอย่าง – โค้ดต่อไปนี้ใช้เซอร์วิส HtmlService สร้างวัตถุ HtmlOutput (โค้ด HTML) สำหรับใช้เป็น
เนื้อหาให้กับเมธอด show()

```
var htmlApp = HtmlService  
    .createHtmlOutput('<p>A change of speed, a change of style...</p>')  
    .setTitle('My HtmlService Application')  
    .setWidth(250)  
    .setHeight(300) ;  
  
SpreadsheetApp.getActiveSpreadsheet().show(htmlApp) ;
```

ผล



บทที่ 4 ชี้ท



4.1. จับชี้ต่อจากไฟล์

เมื่อจับไปที่ Spreadsheet หรือ ไฟล์ Google Sheets แล้ว เช่น
SpreadsheetApp.getActiveSpreadsheet() หรือ SpreadsheetApp.openById(id) เป็นต้น จากนั้น เรา
สามารถจับต่อไปยังชี้ที่ต่างๆในไฟล์ต่อได้ โดยใช้หลายเมธอด

4.2. getActiveSheet()

getActiveSheet() - เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#getactivesheet>

ใช้จับชี้ที่เปิดอยู่ โดยคืนค่ากลับมาเป็นวัตถุ Sheet

ตัวอย่าง

```
var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
```

4.3. getSheets()

getSheet() - เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#getsheets>

ใช้จับชี้ทั้งหมดใน Spreadsheet โดยคืนค่ากลับมาเป็นคลาส **Sheet[]** หรือ หลายชี้ที่เก็บอยู่ในอาร์เรย์
เช่น **[Sheet, Sheet, Sheet, ...]**

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet();  
var sheets = ss.getSheets();  
Logger.log(sheets.length); // นับจำนวนชี้ทั้งหมด  
sheets.forEach(function(sheet){  
    Logger.log(sheet.getName()); // วนพิมพ์ชื่อชี้ทั้งหมด  
});
```

หมายเหตุ : โค้ดต่อไปนี้ จับชี้ในลำดับที่ 1 เพียงชี้เดียว

```
getSheets()[0]  
Logger.log(getSheets()[0].getName())
```

4.4. getSheetByName()

getSheetByName() - เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#getsheetbyname>

ใช้จับชี้ตามชื่อชี้ โดยคืนค่ากลับมาเป็นคลาส Sheet

ตัวอย่าง

```
var sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("Sheet1") ;  
if (sheet != null) { // ถ้า sheet ไม่ใช่ null หรือ ไม่มีชีตใดๆ  
    Logger.log(sheet.getIndex()) ; // พิมพ์ : 1 (ลำดับของชีต)  
}
```

4.5. insertSheet()

insertSheet() - เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#insertsheet>

insertSheet(sheetIndex) - เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#insertsheetsheetindex>

insertSheet(sheetIndex, options) - เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#insertsheetsheetindex,-options>

insertSheet(options) - เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#insertsheetoptions>

insertSheet(sheetName) - เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#insertsheetsheetname>

insertSheet(sheetName, sheetIndex) - เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#insertsheetsheetname,-sheetindex>

insertSheet(sheetName, sheetIndex, options) - เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#insertsheetsheetname,-sheetindex,-options>

insertSheet(sheetName, options) - เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#insertsheetsheetname,-options>

insertSheetWithDataSourceTable(spec) - เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#insertsheetwithdatasourcetablespec>

insertSheet ใช้แทรกชีตใหม่ให้กับ Spreadsheet

พารามิเตอร์

Name	Type	Description
sheetName	String	ชื่อชีต
sheetIndex	Integer	ลำดับของชีตใหม่ที่จะแทรก (เริ่มที่ 0)
option	Object	ระบุเป็นอ็อบเจกต์ของไฟล์ที่จะใช้เป็นเทมเพลต

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;  
var templateSheet = ss.getSheetByName('Sales') ;  
ss.insertSheet('My New Sheet', 1, {template: templateSheet}) ;
```

ตัวอย่าง – โค้ดต่อไปนี้จะตรวจสอบชื่อชีตก่อนว่ามีหรือไม่ ถ้ามีก็ลบก่อน ถ้าไม่มีก็แทรกชีตใหม่ จากนั้นจะทำอะไรในชีตใหม่ก็ว่ากันไป

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;

var newSheet ;
var isSheet = ss.getSheetByName('ชีตใหม่') ;

// ถ้าไม่มีชีตชื่อ 'ชีตใหม่'
if (isSheet == null)
{
    newSheet = ss.insertSheet('ชีตใหม่') ;
}

// ถ้ามีชีตชื่อ 'ชีตใหม่'
else
{
    ss.deleteSheet(isSheet) ;
    newSheet = ss.insertSheet('ชีตใหม่') ;
    // ไม่ต้องลบชีตออก แค่ล้างข้อมูลและรูปแบบก็ได้ แล้วจับใส่ตัวแปรใหม่ ดังนี้
    // newSheet = isSheet.clear() ;
}

// กรอกข้อความลงในชีตใหม่เซลล์ A1
newSheet.getRange('A1').setValue('Something') ;
```

4.6. deleteSheet()

deleteSheet(sheet) - เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#deletesheetsheet>

ใช้ลบชีตที่ระบุ โดยพารามิเตอร์ก็คือ วัตถุ Sheet ที่ต้องการลบ

ตัวอย่าง

```
var ss = SpreadsheetApp.getActive() ;
var sheet3 = ss.getSheetByName('Sheet3') ;

if ( sheet3 != null)
{
    ss.deleteSheet(sheet3) ;
}
```


4.7. activate()

activate() - เมธอดในคลาส Sheet

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#activate>

ใช้กำหนดชีทให้เป็นตัวแอ็คทีฟ คล้ายกับการคลิกเมาส์เลือกชีทดังกล่าว

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var first = ss.getSheetByName("first") ;
first.activate() ; // เลือกชีทชื่อ first
```

4.8. getParent()

getParent() - เมธอดในคลาส Sheet

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#getparent>

คืนค่าเป็นวัตถุ Spreadsheet ที่ชีทนั้นอยู่ **getParent()** เป็นเมธอดในหลายคลาส และหลายเซอร์วิส

ใช้จับวัตถุพ่อแม่

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;

// parent ก็คือ Spreadsheet ที่ชีทนั้นอยู่
var parent = sheet.getParent() ;
```

4.9. การล้างชีท

4.9.ก.) clear()

clear(options) - เมธอดในคลาส Sheet

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#clearoptions>

ใช้ล้างชีทโดย สามารถกำหนดได้ว่าจะล้างเนื้อหาหรือรูปแบบ ซึ่งกำหนดได้ที่พารามิเตอร์ **options** โดย

ต้องกำหนดแบบวัตถุ หรือ ค่าใน { } (ดูตัวอย่าง เพื่อทำความเข้าใจวิธีการกำหนดค่าของ **options**)

พารามิเตอร์

Name	Type	Description
options	Object	อ็อบเจ็ค Javascript ที่ประกอบไปด้วยพารามิเตอร์ขั้นสูง (ดูตาราง พารามิเตอร์ขั้นสูง)

พารามิเตอร์ขั้นสูง

Name	Type	Description
contentsOnly	Boolean	จะล้างเนื้อหาหรือไม่
formatOnly	Boolean	จะล้างรูปแบบหรือไม่

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;  
var sheet = ss.getSheets()[0] ;  
  
sheet.clear({ formatOnly: true, contentsOnly: true }) ;
```

4.9.ข.) clearContents() และ clearFormats()

Clear spreadsheet content while preserving any formatting

<https://developers.google.com/google-ads/scripts/docs/examples/spreadsheetapp#clear-spreadsheet-content-while-preserving-any-formatting>

clearContents() - เมธอดในคลาส Sheet

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#clearcontents>

ใช้ลบเนื้อหาทั้งหมดในชีต แต่ไม่ลบรูปแบบของเซลล์ต่างๆในชีต โดยจะคืนค่ากลับมาเป็นคลาส Sheet

Clear spreadsheet formatting while preserving any data

<https://developers.google.com/google-ads/scripts/docs/examples/spreadsheetapp#clear-spreadsheet-formatting-while-preserving-any-data>

clearFormats() - เมธอดในคลาส Sheet

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#clearformats>

ใช้ล้างรูปแบบทั้งหมดในชีต แต่ไม่ลบเนื้อหาต่างๆในเซลล์ โดยจะคืนค่ากลับมาเป็นคลาส Sheet

ตัวอย่าง

```
var ssID = 'sheet_ID';  
var s1Name = 'Sheet1' ;  
var s2Name = 'Sheet2' ;  
  
var ss = SpreadsheetApp.openById(ssID) ;  
  
var sheet1 = ss.getSheetByName(s1Name) ;  
var sheet2 = ss.getSheetByName(s2Name) ;  
  
sheet1.clearContents() ;  
sheet2.clearFormats() ;
```

เมื่อรันโค้ด เนื้อหาในชีตชื่อ Sheet1 จะถูกลบออกไป แต่รูปแบบของเซลล์ต่างๆ จะยังคงอยู่ ส่วน Sheet2 เนื้อหายังคงอยู่ แต่รูปแบบของเซลล์ถูกล้างออกทั้งชีต

บทที่ 5
การจัดการแถว
หรือคอลัมน์



5.1. การแทรกแถว

insertRowAfter(afterPosition) – เมธอดในคลาส Sheet

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#insertrowafterafterposition>

ใช้แทรก 1 แถว หลังตำแหน่งแถวที่ระบุในพารามิเตอร์ **afterPosition**

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;  
var sheet = ss.getSheets()[0] ;  
  
// แทรกแถวไว้ ด้านล่างของแถวที่ 1  
sheet.insertRowAfter(1) ;
```

insertRowBefore(beforePosition) – เมธอดในคลาส Sheet

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#insertrowbeforebeforeposition>

ใช้แทรก 1 แถว ก่อนตำแหน่งแถวที่ระบุในพารามิเตอร์ **beforePosition**

insertRows(rowIndex) – เมธอดในคลาส Sheet

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#insertrowsrowindex>

ใช้แทรกแถวว่าง 1 แถว ณ ตำแหน่งที่ระบุ (แถวด้านล่างถูกเลื่อนลง)

insertRows(rowIndex, numRows) – เมธอดในคลาส Sheet

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#insertrowsrowindex,-numrows>

ใช้แทรกแถวว่าง 1 หรือหลายแถว ณ ตำแหน่งที่ระบุ (แถวด้านล่างถูกเลื่อนลง)

Name	Type	Description
rowIndex	Integer	ตำแหน่งของแถว ที่จะแทรกแถว
numRows	Integer	จำนวนแถวที่จะแทรก

insertRowsAfter(afterPosition, howMany) – เมธอดในคลาส Sheet

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#insertrowsafterafterposition,-howmany>

แทรกแถว 1 หรือหลายแถว หลังจากตำแหน่งที่ระบุ

Name	Type	Description
afterPosition	Integer	The row after which the new rows should be added.
howMany	Integer	The number of rows to insert.

insertRowsBefore(beforePosition, howMany) – เมธอดในคลาส Sheet

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#insertrowsbeforebeforeposition,-howmany>

แทรกแถว 1 หรือหลายแถว ก่อนตำแหน่งที่ระบุ

Name	Type	Description
beforePosition	Integer	The row before which the new rows should be added.
howMany	Integer	The number of rows to insert.

5.2. การลบแถว

`deleteRow(rowPosition)` – เมธอดในคลาส Sheet

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#deleterowrowposition>

ใช้ลบแถว ณ ตำแหน่งที่ระบุในพารามิเตอร์ `rowPosition`

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;

// ลบแถวที่ 1
sheet.deleteRow(1) ;
```

`deleteRows(rowPosition, howMany)` – เมธอดในคลาส Sheet

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#deleterowsrowposition,-howmany>

ใช้ลบแถว 1 หรือหลายแถว ตามจำนวนที่ระบุในพารามิเตอร์ `howMany` โดยเริ่มจากตำแหน่งที่ระบุใน

พารามิเตอร์ `rowPosition`

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;

// ลบแถวที่ 1 ยาวไป 2 แถว
sheet.deleteRows(1, 2) ;
```


บทที่ 6

เร้นจ์



6.1. จับเร้นจ์

เมื่อจับไปที่ชี้ที่ต้องการได้แล้ว หากต้องการจับต่อไปที่ เร้นจ์ หรือ เซลล์ต่างๆ สามารถทำได้โดยใช้หลาย เมธอดในคลาส Range

6.2. getRange()

คลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range>

getRange() ใช้จับเร้นจ์ โดยจะคืนค่ากลับมาเป็นวัตถุ Range

6.2.ก.) getRange(row, column)

getRange(row, column) - เมธอดในคลาส Sheet

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#getrangerow,-column>

ใช้จับเร้นจ์เพียง 1 เซลล์ โดยระบุเร้นจ์เป็นตัวเลขตำแหน่งแถวและคอลัมน์ เช่น getRange(2, 3) ก็คือ แถวที่ 2 คอลัมน์ที่ 3 (เซลล์ C2) เป็นต้น

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;    // จับชีทแรก

var range = sheet.getRange(1, 1) ; // จับเซลล์ A1
var values = range.getValues() ;   // จับค่าในเซลล์ A1 - แบบอาเรย์ 2 มิติ
// var value = range.getValue() ;  // จับค่าในเซลล์เดียวใช้ getValue ง่ายกว่า

Logger.log(values[0][0]) ;         // พิมพ์ : (ค่าในเซลล์ A1)
```

6.2.ข.) getRange(row, column, numRows)

getRange(row, column, numRows) - เมธอดในคลาส Sheet

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#getrangerow,-column,-numrows>

ใช้จับเร้นจ์ 1 คอลัมน์ โดยระบุตัวเลขตำแหน่งเป็น จุดเริ่มแถว จุดเริ่มคอลัมน์ และจำนวนแถว

ตัวอย่างการใช้งาน

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;

var range = sheet.getRange(1, 1, 3) ; // จับเริ่มต้นที่ A1 ยาวไป 3 แถว
var values = range.getValues() ;      // จับค่าในเร้นจ์ - ได้มาเป็นอาเรย์ 2 มิติ

for (var row in values) {
  for (var col in values[row]) {
    Logger.log(values[row][col]) ;    // วนลูป Log ค่าในเซลล์ A1 ถึง A3 - 3 ค่า
  } // จบ for ใน
} // จบ for นอก
```

หมายเหตุ : เมธอดตระกูลวนลูปของอาเรย์ เช่น **forEach** นิยมใช้เข้าถึงข้อมูลในอาเรย์มากกว่าลูป for

6.2.ค.) `getRange(row, column, numRows, numColumns)`

`getRange(row, column, numRows, numColumns)` - เมธอดในคลาส `Sheet`

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#getrangerow,-column,-numrows,-numcolumns>

ใช้จับเร้นจ์แบบตารางข้อมูล โดยระบุตัวเลขตำแหน่งเป็น จุดเริ่มแถว จุดเริ่มคอลัมน์ จำนวนแถว และจำนวนคอลัมน์ ตามลำดับ

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;
var range = sheet.getRange(1, 1, 3, 3) ; // จับเริ่มต้นที่ A1 ยาวไป 3 แถว 3 คอลัมน์
var values = range.getValues() ; // จับค่าในเร้นจ์ - ได้มาเป็นอาเรย์ 2 มิติ

for (var row in values) {
  for (var col in values[row]) {
    Logger.log(values[row][col]) ; // วนลูป Log ค่าในเซลล์ A1 ถึง C3 - ได้ 9 ค่า
  } // จบ for ใน
} // จบ for นอก
```

6.2.ง.) `getRange(a1Notation)`

`getRange(a1Notation)` - เมธอดในคลาส `Sheet`

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#getrangea1notation>

ใช้จับเร้นจ์ โดยระบุพารามิเตอร์ `a1Notation` เป็นสัญลักษณ์ A1 Notation เช่น A5 หรือ A5:B10 เป็นต้น หรือสัญลักษณ์ R1C1 เช่น R[1]C[2]

ตัวอย่าง

```
// จับเร้นจ์ A1:D4 ที่อยู่บนชีท "Invoices"
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var range = ss.getRange("Invoices!A1:D4") ;

// จับเร้นจ์ A1 ในชีทที่ 1
var sheet = ss.getSheets()[0] ;
var cell = sheet.getRange("A1") ;
```

6.3. `getLastRow()` และ `getLastColumn()`

How to Get The Last Row - Google Sheets Apps Scripts - Array Methods Part 10

https://youtu.be/zWcNzLbJpb0?list=PLv9Pf9aNgemvD9NFa86_uDt-NWh37efmD

`getLastRow()` - เมธอดในคลาส `Range`

<https://developers.google.com/apps-script/reference/spreadsheet/range#getlastrow>

`getLastColumn()` - เมธอดในคลาส `Range`

<https://developers.google.com/apps-script/reference/spreadsheet/range#getlastcolumn>

`getLastRow()` และ `getLastColumn()` ใช้จับแถวสุดท้าย หรือ คอลัมน์สุดท้ายของเร้นจ์ที่เราจับไว้ ตามลำดับ โดยจะคืนค่ากลับมาเป็น เลขจำนวนเต็ม แถวที่ หรือ คอลัมน์ที่

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;
var range = sheet.getRange("B3:D10") ;

var lr = range.getLastRow() ;
var lc = range.getLastColumn() ;

Logger.log(lr + " " + lc) ; // พิมพ์ : 10 4
```

6.4. getA1Notation()

getA1Notation() - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#geta1notation>

getA1Notation() ใช้จับตำแหน่งของเร้นต์ที่เลือกหรือระบุ แล้วคืนค่ากลับมาเป็นตำแหน่งของเร้นจ์ในรูปแบบ

แบบ A1 Notation เช่น A2:G20 เป็นต้น

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;

var range = sheet.getRange(1, 1, 2, 5) ;

Logger.log(range.getA1Notation()) ; // พิมพ์ : A1:E2
```

6.5. getCell()

getCell(row, column) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#getcellrow,-column>

ใช้จับ 1 เร้นจ์ที่ระบุ โดยจะคืนค่ากลับมาเป็นวัตถุ Range ของ 1 เซลล์

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;
var range = sheet.getRange("B2:D4") ;

// จับเซลล์ตำแหน่ง 1,1 ในเร้นจ์ B2:D4 ก็คือ จับไปที่เซลล์ B2
var cell = range.getCell(1, 1) ;
Logger.log(cell.getValue()) ;
```

หมายเหตุ

getCell() ใช้จับเซลล์ในเร้นจ์ต่อได้ด้วย เช่น

```
sheet.getRange("A1:D1").getCell(1,1) ; // จับเร้นจ์ A1 เซลล์เดียว
sheet.getRange("A1:D1").getCell(1,2) ; // จับเร้นจ์ A2 เซลล์เดียว
```

6.6. getDataRegion()

6.6.ก.) getDataRegion()

getDataRegion() - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#getDataRegion>

getDataRegion() ใช้จับเรนจ์ที่อยู่ติดกับเซลล์ที่เราเลือก คล้ายกับเลือกบางเซลล์ไว้ เช่นเลือก B2 จากนั้นกด **<Ctrl><A>** อาณาเขตการเลือกจะขยายออกไปครอบคลุมเซลล์ที่มีข้อมูลที่อยู่ติดกับ B2 ด้วย

ตัวอย่างข้อมูล

	A	B	C	D	E	F	G	H	I	J
1	กขค5	กขค6	กขค7	กขค8						
2	จฉช5	จฉช6	จฉช7	จฉช8						
3	ตบล5	ตบล6	ตบล7	ตบล8						
4	อพย5	อพย6	อพย7	อพย8						
5										
6						กขค5	กขค6	กขค7	กขค8	
7						จฉช5	จฉช6	จฉช7	จฉช8	
8						ตบล5	ตบล6	ตบล7	ตบล8	
9						อพย5	อพย6	อพย7	อพย8	

ตัวอย่างการใช้งาน

```
var sheet = SpreadsheetApp.getActiveSpreadsheet().getSheets()[0] ;  
var Region_1 = sheet.getRange("B2").getDataRegion().getA1Notation() ;  
var Region_2 = sheet.getRange("H8").getDataRegion().getA1Notation() ;  
  
Logger.log(Region_1) ; // พิมพ์ : A1:D4  
Logger.log(Region_2) ; // พิมพ์ : F6:I9
```

6.6.ข.) getDataRegion(dimension)

getDataRegion(dimension)

<https://developers.google.com/apps-script/reference/spreadsheet/range#getDataRegionDimension>

getDataRegion(dimension) ใช้จับเรนจ์ที่อยู่ติดกับเซลล์ที่เราเลือก โดยสามารถเลือกทิศทางได้ การขยายเรนจ์ได้ getDataRegion(dimension) คืนค่ากลับมาเป็นคลาส Range โดยพารามิเตอร์ dimension ก็คือ Enum Direction(ใช้ไม่ได้) หรือ Enum Dimension

Enum Direction – การเรียกใช้งานเช่น SpreadsheetApp.Direction.DOWN เป็นต้น

<https://developers.google.com/apps-script/reference/spreadsheet/direction#UP>

Property	Type	Description
UP	Enum	The direction of decreasing row indices.
DOWN	Enum	The direction of increasing row indices.
PREVIOUS	Enum	The direction of decreasing column indices.
NEXT	Enum	The direction of increasing column indices.

Enum Dimension – การเรียกใช้งานเช่น `SpreadsheetApp.Dimension.COLUMNS` เป็นต้น
<https://developers.google.com/apps-script/reference/spreadsheet/dimension>

Property	Type	Description
<code>COLUMNS</code>	Enum	The column (vertical) dimension.
<code>ROWS</code>	Enum	The row (horizontal) dimension.

ตัวอย่าง

```
// แรกเริ่มก่อนรันโค้ด ชีทไม่มีข้อมูลใดๆ
var ss = SpreadsheetApp.getActiveSpreadsheet();
var sheet = ss.getSheets()[0];

// กรอกข้อมูลลงในเซลล์ต่างๆ 4 เซลล์
sheet.getRange("C2").setValue(100);
sheet.getRange("B3").setValue(100);
sheet.getRange("D3").setValue(100);
sheet.getRange("C4").setValue(100);

Logger.log( // Logs "C2:C4"
  sheet.getRange("C3").getDataRegion(SpreadsheetApp.Dimension.ROWS).getA1Notation()
);
Logger.log( // Logs "B3:D3"
  sheet.getRange("C3").getDataRegion(SpreadsheetApp.Dimension.COLUMNS).getA1Notation()
);
```

	A	B	C	D	E
1					
2			100		
3		100		100	
4			100		
5					

6.7. getDataRange()

`getDataRange()` - เมธอดในคลาส Sheet

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#getdatarange>

ใช้จับข้อมูลที่มีในชีท ตั้งแต่มุมซ้ายบนที่มีข้อมูล ไปจนถึงมุมขวาล่างที่มีข้อมูลตัวสุดท้าย

`getDataRange` ให้ผลเหมือนกับโค้ดนี้ `getRange(1,1,sheet.getLastRow(),sheet.getLastColumn())`

ตัวอย่าง

ตารางข้อมูลใน Google Sheet มีดังต่อไปนี้

	A	B	C
2	Apple	15	200
3	Bananas	50	520
4	Mangoes	20	1025
5	Durian	150	50
6	Longan	60	450

โค้ดต่อไปนี้จะ Logs ข้อมูลทั้งหมดที่มีในชีตออกมาดู

```
var ss = SpreadsheetApp.getActive() ; // ไฟล์ปัจจุบัน
var sheet = ss.getActiveSheet() ; // ชีตที่เปิด
var data = sheet.getDataRange() ; // เรนจ์ข้อมูล
var values = data.getValues() ; // ข้อมูลในชีตทั้งหมด

Logger.log(values) ; // ดูผลที่ Logs ----- > [ 01 ]
```

ผล

Logs

```
[ 01 ] [ [ Fruit,      Cost,    Quantity ],
        [ Apple,     15,      200     ],
        [ Bananas,   50.0,    520.0    ],
        [ Mangoes,   20,      1025    ],
        [ Durian,    150,      50      ],
        [ Longan,    60,      450     ] ]
```

อีกตัวอย่าง ต่อเนื่องจากตัวอย่างก่อนหน้านี้ - ย้ายข้อมูลออกมา ให้มีแถวว่างและคอลัมน์ว่างที่ด้านซ้ายและด้านบน

	A	B	C	D
1				
2				
3		Fruit	Cost	Quantity
4		Apple	15	200
5		Bananas	50	520
6		Mangoes	20	1025
7		Durian	150	50
8		Longan	60	450

Logs ข้อมูลในรูปแบบ CSV ขึ้นด้วย ", " (คอมม่า)

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;

var range = sheet.getDataRange() ;
var values = range.getValues() ;

for (var i = 0; i < values.length; i++) {
  var row = "" ;
  for (var j = 0; j < values[i].length; j++) {
    if (values[i][j]) {
      row = row + values[i][j] ;
    }
    row = row + ", " ;
  }
  Logger.log(row) ; // Log ข้อมูลในรูปแบบ CSV ขึ้นด้วย , (คอมม่า)
}
```

Logs

```
[01] ,,,
[02] ,,,
[03] ,Fruit,Cost,Quantity,
[04] ,Apple,15,200,
[05] ,Bananas,50,520,
[06] ,Mangoes,20,1025,
[07] ,Durian,150,50,
[08] ,Longan,60,450,
```

หมายเหตุ :

เมื่อใช้ `getDataRange()` จับข้อมูลในชีทแล้ว มักจะใช้อาร์เรย์เมธอด เช่น `shift()`, `splice()` หรือ `slice()` ตัดข้อมูลที่ไม่ต้องการออกไป เช่น ตัดหัวตาราง จากนั้นจะเหลือเฉพาะส่วนที่ต้องการ

6.8. `getValue()` และ `getValues()`

`getValue()` - เมธอดในคลาส `Range`

<https://developers.google.com/apps-script/reference/spreadsheet/range#getValue>

ใช้จับค่าในเรนจ์ที่มุมซ้ายบน(จับมาค่าเดียว) ค่าในเรนจ์ที่จับสามารถเป็น ตัวเลข, บูลีน, ข้อความ เป็นต้น หากเซลล์ว่างสิ่งที่จับมาได้จะเป็น ข้อความว่างๆ

ตัวอย่าง

```
var valInA1 = SpreadsheetApp.getActiveSheet().getRange(1, 1).getValue() ;
```

`getValues()` - เมธอดในคลาส `Range`

<https://developers.google.com/apps-script/reference/spreadsheet/range#getValues>

ใช้จับค่าในเรนจ์ที่ระบุ ค่าที่คืนกลับมาเป็นอาร์เรย์ 2 มิติ (แม้จะจับไปที่เซลล์เดียวก็ตาม) สามารถระบุตำแหน่งภายในอาร์เรย์ได้ดังนี้ `getValues()[i][j]`

ตัวอย่าง

```
// จับค่าในเรนจ์ C2:G8
var values = SpreadsheetApp.getActiveSheet().getRange(2, 3, 6, 4).getValues();
Logger.log(values[0][0]) ; // ดูค่าในอาร์เรย์ตัวแรกตัวเดียว
```

6.9. getDisplayValue() และ getDisplayValues()

getDisplayValue() - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#getdisplayvalue>

ใช้จับค่าตามที่แสดงให้เห็นในเรนจ์ โดยจับที่มุมซ้ายบน(จับมาค่าเดียว) โดยจะคืนค่ากลับมาเป็น **String**

getDisplayValues() - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#getdisplayvalues>

ใช้จับค่าตามที่แสดงให้เห็นในเรนจ์ โดยจะคืนค่ากลับมาเป็น **String[][]**-อาเรย์ 2 มิติที่มีข้อมูลเป็น String

ตัวอย่าง - ตารางข้อมูลมีดังต่อไปนี้

	A
1	฿10.00
2	฿10.50

โค้ด

```
var sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName('Sheet1') ;  
var values = sheet.getRange('C1:C2').getValues() ;  
Logger.log(values) ; // พิมพ์ : [[10.0], [10.5]]  
var displayValues = sheet.getRange('C1:C2').getDisplayValues() ;  
Logger.log(displayValues) ; // พิมพ์ : [[฿10.00], [฿10.50]]
```

6.10. จับเรนจ์จากชื่อเรนจ์ (Named ranges)

Retrieve a named range from a spreadsheet

<https://developers.google.com/google-ads/scripts/docs/examples/spreadsheetapp#retrieve-a-named-range-from-a-spreadsheet>

6.10.ก.) getRangeByName()

getRangeByName(name) - เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#getrangebyname>

ใช้จับเรนจ์จาก ชื่อเรนจ์(Named Range) พารามิเตอร์ **name** เป็นชื่อของเรนจ์

ชื่อเรนจ์ หรือ **Named Range** เมื่อเริ่มต้นใช้งาน Google Sheets จะยังไม่มี แต่เราสามารถสร้าง **Named Range** หรือ ตั้งชื่อให้กับเรนจ์ ได้ที่เมนู Data → Named Ranges (ตัวอย่างตามภาพถัดไป)

การตั้งชื่อเรนจ์ ทำให้เราสามารถเข้าถึงเรนจ์ได้ง่ายจากชื่อของเรนจ์ เช่น `sum(NamedRange)` เป็นต้น

The screenshot shows a Google Sheets spreadsheet with a table of fruit prices. The table has columns for Fruit, Cost, Quantity, and Total. The data is as follows:

Fruit	Cost	Quantity	Total
Apple	15	200	3,000
Bananas	50	520	26,000
Mangoes	20	1,025	20,500
Mangoes	20	123	20,500
Durian	150	50	7,500
Durian	150	50	7,500
Longan	60	450	27,000
Total			132,500

On the right, the 'Named ranges' dialog box is open, showing a range named 'Test' with the formula 'Sheet1!B4:E12'.

โค้ดต่อไปนี้จะดึงข้อมูลของเรนจ์ จากชื่อของเรนจ์(Named Range) ออกมา Log ดู

```
var ssID = 'Your Google Sheets ID' ;
var ss = SpreadsheetApp.openById(ssID) ;

var range = ss.getRangeByName('Test') ;           // จับเรนจ์ชื่อ Test

if (range != null) {
  Logger.log(range.getNumColumns()) ;              // พิมพ์ : 4.0
  Logger.log(range.getNumRows()) ;                 // พิมพ์ : 9.0
  Logger.log(range.getA1Notation()) ;               // พิมพ์ : B4:E12
  Logger.log(range.getSheet().getName()) ;         // พิมพ์ : ชื่อชีท
} // Close – if
```

6.10.ข.) `getNamedRanges()`

`getNamedRanges()` - เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#getnamedranges>

ใช้จับ Named Ranges ในไฟล์ Google Sheets โดยคืนค่ากลับมาเป็น `NamedRange[]`-วัตถุ

`NamedRange` ในอาเรย์

`getNamedRanges()` - เมธอดในคลาส Sheet

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#getnamedranges>

ใช้จับ Named Ranges ในชีทใดๆ โดยคืนค่ากลับมาเป็น `NamedRange[]`-วัตถุ `NamedRange` ในอา

เรย์

ตัวอย่างที่ 1 – จับ Named ranges ในไฟล์ Google Sheets มา Logs ดูข้อมูล

```
var ssID = 'Your Google Sheets ID' ;
var ss = SpreadsheetApp.openById(ssID) ;

var nrange = ss.getNamedRanges() ;
Logger.log(nrange) ; // พิมพ์ : [NamedRange]

nrange.forEach(function(nr){
  Logger.log(nr.getName()) ; // พิมพ์ : Test – ชื่อ Name range
  Logger.log(nr.getRange().getA1Notation()) ; // พิมพ์ : B4:E12 – เรนจ์ของ Name range
}) ;
```

ตัวอย่างที่ 2 – จับ Named ranges ในชีทที่ระบุ มา Logs ดูชื่อ Named range ตัวแรก

```
// The code below logs the name of the first named range.

var namedRanges = SpreadsheetApp.getActiveSheet().getNamedRanges() ;

if (namedRanges.length > 1)
{
  Logger.log(namedRanges[0].getName()) ;
}
```

6.10.ค.) setNamedRange()

setNamedRange(name, range) – เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#setnamedrangename,-range>

ใช้สร้าง Named Range

พารามิเตอร์

Name	Type	Description
name	String	ชื่อที่จะใช้ตั้งเป็นชื่อของ Named Range
range	Range	วัตถุเรนจ์

ตัวอย่างที่ 1 – สร้าง Named range จากเรนจ์ที่ใช้เมาส์เลือก

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
ss.setNamedRange("foo", ss.getActiveRange()) ;
ss.removeNamedRange("foo") ;
```

ตัวอย่างที่ 2 – สร้าง Named range จากเรนจ์ D4:D10 ชีท Sheet2

```
var ss = SpreadsheetApp.getActiveSpreadsheet();
ss.setNamedRange("NameRangeX",ss.getSheetByName('Sheet2').getRange('D4:D10')) ;
```

6.10.ง.) removeNamedRange()

removeNamedRange(name) - เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#removenamedrangename>

ใช้ลบ Named Range พารามิเตอร์ **name** ก็คือชื่อ Name Range ที่จะลบ

ตัวอย่าง

```
ss.removeNamedRange('Test') ;
```

6.10.จ.) setRange() และ setName()

setRange(range) - เมธอดในคลาส NamedRange

<https://developers.google.com/apps-script/reference/spreadsheet/named-range#setrangerange>

ใช้เซตหรืออัปเดตเรนจ์ให้กับ Named range ที่ระบุ โดยคืนค่ากลับมาเป็น Named range ตัวเดิม

setName(name) - เมธอดในคลาส NamedRange

<https://developers.google.com/apps-script/reference/spreadsheet/named-range#setnamenname>

ใช้เซตหรืออัปเดตชื่อใหม่ให้กับ Named range ที่ระบุ โดยคืนค่ากลับมาเป็น Named range ตัวเดิม

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheetByName('Sheet1') ;

var range1 = sheet.getRange('A1:D1') ;
var range2 = sheet.getRange('A1:D13') ;

// สร้าง Named range
ss.setNamedRange("DataHeader", range1) ;

var NameRangesAll = ss.getNamedRanges() ;

NameRangesAll.forEach(function(nr){
    Logger.log(nr.getName()) ; // ดูผลที่ Logs ----- > (1.)
    Logger.log(nr.getRange().getA1Notation()) ; // ดูผลที่ Logs ----- > (2.)
}) ;

NameRangesAll.forEach(function(nr){
    // แก่เรนจ์ให้กับ Named range ชื่อ DataHeader
    // และเปลี่ยนชื่อ Named range เป็น DataTable
    if(nr.getName() == 'DataHeader')
    {
        nr.setRange(range2) ; // เซตเรนจ์ใหม่
        Logger.log(nr.setName('DataTable').getName()) ; // ดูผลที่ Logs ----- > (3.)
        Logger.log(nr.getRange().getA1Notation()) ; // ดูผลที่ Logs ----- > (4.)
    }
})
```

Logs

```
[ 1. ] DataHeader
[ 2. ] A1:D1
[ 3. ] DataTable
[ 4. ] A1:D13
```

6.10.จ.) remove()

remove() - เมธอดในคลาส NamedRange

<https://developers.google.com/apps-script/reference/spreadsheet/named-range#remove>

ใช้ลบ Named ranges

ตัวอย่าง - ลบทุก Named ranges ในไฟล์ Google Sheets

```
// ลบ Named ranges ทุกตัวใน Spreadsheet
var namedRanges = SpreadsheetApp.getActive().getNamedRanges();
for (var i = 0 ; i < namedRanges.length ; i++) {
    namedRanges[i].remove() ;
}
```

ตัวอย่าง - ลบทุก Named ranges ในชีทที่จับมา

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var nrInSheet = ss.getSheetByName('sheetname').getNamedRanges() ;
nrInSheet.forEach(function(nr){
    nr.remove() ;
}) ; // foreach
```

6.11. offset()

`offset(rowOffset, columnOffset)` – เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#offsetrowoffset,-columnoffset>

`offset(rowOffset, columnOffset, numRows)` – เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#offsetrowoffset,-columnoffset,-numrows>

`offset(rowOffset, columnOffset, numRows, numColumns)` – เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#offsetrowoffset,-columnoffset,-numrows,-numcolumns>

`offset` ใช้เลื่อน หด หรือขยายเรนจ์ออกไป โดยคืนค่ากลับมาเป็นวัตถุ Range เหมือนเดิม

พารามิเตอร์

Name	Type	Description
<code>rowOffset</code>	Integer	ตำแหน่งแถว นับลงล่าง จากตำแหน่งมุมซ้ายบน ของเรนจ์อ้างอิง เลขลบ หมายถึง ทิศขึ้น
<code>columnOffset</code>	Integer	ตำแหน่งคอลัมน์ นับไปทางขวา จากตำแหน่งมุมซ้ายบน ของเรนจ์อ้างอิง เลขลบ หมายถึง นับไปทางซ้าย
<code>numRows</code>	Integer	ความสูงของแถว ของเรนจ์ใหม่
<code>numColumns</code>	Integer	ความกว้างของคอลัมน์ ของเรนจ์ใหม่

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;
var range = sheet.getRange("A1:C2") ;

var newRange1 = range.offset(1, 1) ;           // ขยับเรนจ์ทั้งก้อนไป 1 แถว 1 คอลัมน์
Logger.log(newRange1.getA1Notation()) ;       // B2:D3

var newRange2 = range.offset(3, 3, 3) ;        // ขยับเรนจ์ทั้งก้อนไป 3 แถว 3 คอลัมน์ และ
Logger.log(newRange2.getA1Notation()) ;       // เลือกเรนจ์ 3 แถว นับจากมุมซ้ายบน ของเรนจ์ใหม่
                                              // D4:F6

var newRange3 = range.offset(5, 4, 2, 2) ;     // ขยับเรนจ์ทั้งก้อนไป 5 แถว 4 คอลัมน์ และ
Logger.log(newRange3.getA1Notation()) ;       // เลือกเรนจ์ 2 แถว 2 คอลัมน์
                                              // นับจากมุมซ้ายบน ของเรนจ์ใหม่
                                              // E6:F7
```

ผล

	A	B	C	D	E	F
1	range					
2		newRange1				
3						
4				newRange2		
5						
6					newRange3	
7						

6.12. setValue() และ setValues()

setValue(**value**) – เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setValuevalue>

setValues(**values**) – เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setValuesvalues>

Set a range's values

<https://developers.google.com/google-ads/scripts/docs/examples/spreadsheetapp#set-a-ranges-values>

setValue() และ setValues() ใช้กรอกข้อมูลลงไปในพื้นที่ โดย setValue() กรอกลงในเซลล์เดียว ส่วน setValues() กรอกลงในพื้นที่(หลายเซลล์)

พารามิเตอร์ **value** เป็นค่าใดๆ String หรือ Number เป็นต้น ส่วน พารามิเตอร์ **values** เป็นอาร์เรย์ 2 มิติ

ตัวอย่างการใช้งาน setValue()

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;
var cell = sheet.getRange("B2") ;

cell.setValue(100) ;
```

ตัวอย่างการใช้ setValues()

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;

var values = [ [ "2.000", "1,000,000", "$2.99" ] ] ; // มิติ 1x3

var range = sheet.getRange("B2:D2") ; // มิติ 1x3

range.setValues(values) ; // กรอกอาร์เรย์ลงในเซลล์ B2:D2
```

6.13. นำคอลัมน์มาคำนวณ แล้ววางผลลัพธ์ไว้คอลัมน์ข้างๆ

ตารางข้อมูลเริ่มต้น มีดังต่อไปนี้

	A	B	C	D
1	Fruit	Cost	Quantity	Total
2	Apple	15	200	
3	Bananas	50	520	
4	Mangoes	20	1,025	
5	Durian	150	50	
6	Longan	60	450	

โค้ดต่อไปนี้คำนวณค่าในคอลัมน์ Total

```
function addTotal(){
  var ss = SpreadsheetApp.getActive() ;
  var sheet1 = ss.getSheetByName("Sheet1") ;
  var lastRow = sheet1.getLastRow() ;

  // จับข้อมูล ตั้งแต่แถวที่ 2 จนถึงแถวสุดท้าย คอลัมน์ A-C
  var data = sheet1.getRange(2, 1, lastRow-1, 3).getValues() ;

  var totalsArray = [] ;           // เตรียมอาเรย์ว่างไว้เก็บผลลัพธ์

  data.forEach(function(row) {    // ส่งแถวข้อมูลไปประมวลผล
    var name = row[0] ;           // จับข้อมูลตัวที่ 0 (คอลัมน์ Fruit) เช่น Apple – ไม่ได้ใช้
    var cost = row[1] ;           // จับข้อมูลตัวที่ 1 (คอลัมน์ Cost) เช่น 15
    var quantity = row[2] ;       // จับข้อมูลตัวที่ 2 (คอลัมน์ Quantity) เช่น 200
    var total = cost * quantity ; // คำนวณ เช่น ผลลัพธ์ 3,000

    row.push(total) ;             // ยัดผลการคำนวณ ไปต่อท้ายแถว
                                  // ได้ผลลัพธ์ เช่น [ Apple , 15 , 200 , 3000 ]
    totalsArray.push([total]) ;   // เอาผลการคำนวณไปใส่ในอาเรย์ว่างๆ ที่เตรียมไว้
                                  // มี [ ] ครอบ Total หมายถึงใส่อาเรย์ลงไป จึงเป็น อาเรย์ 2 มิติ
  }) ; // End – forEach

  sheet1.getRange(2,4, lastRow-1,1).setValues(totalsArray) ; // วางข้อมูลในอาเรย์ผลลัพธ์ ลงในเรนจ์
}
```

ผล – สิ่งที่ได้ที่คอลัมน์ D

	A	B	C	D
1	Fruit	Cost	Quantity	Total
2	Apple	15	200	3,000
3	Bananas	50	520	26,000
4	Mangoes	20	1,025	20,500
5	Durian	150	50	7,500
6	Longan	60	450	27,000

6.14. กรอกสูตรลงเซลล์

6.14.ก.) setFormula()

setFormula(formula) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setformulaformula>

setFormulas(formulas) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setformulasformulas>

Set cell formula

<https://developers.google.com/google-ads/scripts/docs/examples/spreadsheetapp#set-cell-formula>

setFormula() ใช้กรอกสูตรของ Google Sheets ลงในเร้นจ์ เป็นสูตรที่เราใช้กันอย่างปกติในการคำนวณทั่วไป เช่น =SUM(A2:A5) เป็นต้น

ตัวอย่างที่ 1

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;
var cell = sheet.getRange("B5") ;
cell.setFormula("=SUM(B3:B4)") ; // กรอกสูตร =SUM(B3:B4) ลงในเซลล์ B5
```

ตัวอย่างที่ 2 - กรอกสูตรที่แตกต่างกันลงในเร้นจ์

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;

var formulas = [
    ["=SUM(B2:B4)", "=SUM(C2:C4)", "=SUM(D2:D4)" ] ,
    ["=AVERAGE(B2:B4)", "=AVERAGE(C2:C4)", "=AVERAGE(D2:D4)" ]
] ; // มิติ 2x3
var cell = sheet.getRange("B5:D6") ; // มิติ 2x3
cell.setFormulas(formulas) ;
```

ตัวอย่างที่ 3

```
var ss = SpreadsheetApp.getActive() ;
var sheet1 = ss.getSheetByName('Sheet1') ;
var lastRow = sheet1.getLastRow() ;
var lastColumn = sheet1.getLastColumn() ;

// จับแถวสุดท้าย+1 ในคอลัมน์ A (เซลล์ A7) -> จากนั้นใส่ข้อความ Total ลงไป
sheet1.getRange(lastRow+1, 1).setValue("Total") ;

// จับแถวสุดท้าย+1 ในคอลัมน์สุดท้าย (เซลล์ D7) -> จากนั้นใส่สูตร SUM
sheet1.getRange(lastRow+1, lastColumn).setFormula("SUM(D2:D"+lastRow+")") ;
```

ผล

	A	B	C	D
1	Fruit	Cost	Quantity	Total
2	Apple	15	200	3,000
3	Bananas	50	520	26,000
4	Mangoes	20	1,025	20,500
5	Durian	150	50	7,500
6	Longan	60	450	27,000
7	Total			84,000

6.14.ข.) setFormulaR1C1()

setFormulaR1C1(**formula**) – เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setformulaR1C1formula>

setFormulasR1C1(**formula**) – เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setformulasR1C1formulas>

setFormulaR1C1() ใช้กรอกสูตรลงเร้นจ์ โดยสูตรที่กรอกลงไป เป็นสูตรที่ใช้ระบบการอ้างอิงเซลล์แบบ R1C1 หรือ เป็นระยะออฟเซตแถวและคอลัมน์จากเร้นจ์อ้างอิง

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheetByName('sheet1') ;
var cell = sheet.getRange("B5") ;

// กรอกสูตรลงในเซลล์ B5 เป็นการ SUM ของ 3 แถว เหนือ B5
//
cell.setFormulaR1C1("=SUM(R[-3]C[0]:R[-1]C[0])") ;
```

ผล

fx	=SUM(B2:B4)	
	A	B
1		
2		2
3		3
4		4
5		9
6		

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;

var sumOfRowsAbove = "=SUM(R[-3]C[0]:R[-1]C[0])" ;
var averageOfRowsAbove = "=AVERAGE(R[-4]C[0]:R[-2]C[0])" ;

var formulas = [ [sumOfRowsAbove, sumOfRowsAbove, sumOfRowsAbove] , // มิติ 2x3
                 [averageOfRowsAbove, averageOfRowsAbove, averageOfRowsAbove] ] ;

var cell = sheet.getRange("B5:D6") ; // มิติ 2x3

// กรอกสูตรลงในเซลล์ B5:D6
// แถวที่ 5(คอลัมน์ B-D) เป็นค่า SUM ของ 3 แถวเหนือแถวที่ 5 (แถว 2-4)
// แถวที่ 6(คอลัมน์ B-D) เป็นค่า AVG ของ 3 แถวเหนือแถวที่ 5 (แถว 2-4)
cell.setFormulasR1C1(formulas) ;
```

ผล

fx		=SUM(B2:B4)			
		A	B	C	D
1					
2			1	10	123
3			2	20	456
4			3	30	789
5			6	60	1368
6			2	20	456

6.15. getActiveRange()

getActiveRange() - เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#getactiverange>

ใช้จับเร้นต์ในชีทที่ถูกเลือกโดยเมาส์ โดยจะคืนค่ากลับมาเป็นวัตถุ Range ซึ่งเราสามารถใช้เมธอดของคลาส Range จักรายละเอียดหรือเซตสิ่งต่างๆลงไปเ็นเร้นจ้ต่อไปได้

ตัวอย่าง - ก่อนรันโค้ดให้เลือกรั้นจ้ไว้ก่อน

	A	B	C	D
1				
2				
3		First name	Last name	email
4		Wasan	Khunnadiloksawet	wasankds@gmail.com
5		Napabhorn	kds	napa@gmail.com
6				

```

var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
var activeRange = sheet.getActiveRange();

Logger.log(activeRange); // พิมพ์ : Range
Logger.log(activeRange.getA1Notation()); // พิมพ์ : B3:D5
Logger.log(activeRange.getSheet().getName()); // พิมพ์ : Temp

```

6.16. getRow(), getColumn(), getNumRows(), getNumColumns()

getRow() - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#getrow>

ใช้จับตำแหน่งของแถวในเรนจ์ คืนค่าเป็น Integer (เลขจำนวนเต็ม)

getColumn() - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#getcolumn>

ใช้จับตำแหน่งของคอลัมน์ในเรนจ์ คืนค่าเป็น Integer (เลขจำนวนเต็ม)

getNumRows() - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#getnumrows>

ใช้จับจำนวนแถวในเรนจ์ คืนค่าเป็น Integer (เลขจำนวนเต็ม)

getNumColumns() - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#getnumcolumns>

ใช้จับจำนวนคอลัมน์ในเรนจ์ คืนค่าเป็น Integer (เลขจำนวนเต็ม)

getRowIndex() - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#getrowindex>

ใช้จับตำแหน่งของแถวในเรนจ์ คืนค่าเป็น Integer (เลขจำนวนเต็ม) เหมือนกับ getRow()

ตัวอย่างการใช้งาน

```

var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
var range = sheet.getRange("C2:E7"); // 6 แถว 3 คอลัมน์

Logger.log(range.getRow()); // พิมพ์ : 2 (เซลล์แรก C2 คี่น 2)
Logger.log(range.getColumn()); // พิมพ์ : 3 (เซลล์แรก C2 คี่น 3 ก็คือ C)

Logger.log(range.getNumRows()); // พิมพ์ : 6 (6 แถว)
Logger.log(range.getNumColumns()); // พิมพ์ : 3 (3 คอลัมน์)

Logger.log(range.getRowIndex()); // พิมพ์ : 2

```

6.17. การล้างเนื้อหา และ การล้างรูปแบบ

6.17.ก.) clearContents()

clearContent() - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#clearcontent>

ใช้ลบเนื้อหาทั้งหมดในเรนจ์ แต่ไม่ลบรูปแบบของเซลล์ต่างๆในเรนจ์ โดยจะคืนค่ากลับมาเป็นคลาส

Range ตัวเดิม

ตัวอย่าง – ล้างเนื้อหาในเรนจ์ A1:D10

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;  
var sheet = ss.getSheets()[0] ;  
var range = sheet.getRange("A1:D10") ;  
range.clearContent() ;
```

6.17.ข.) clearFormats()

clearFormat() - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#clearformat>

ใช้ล้างรูปแบบทั้งหมดในเรนจ์ แต่ไม่ลบเนื้อหาต่างๆในเรนจ์ โดยจะคืนค่ากลับมาเป็นคลาส Range

ตัวอย่าง – ล้างรูปแบบในเรนจ์ A1:D10

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;  
var sheet = ss.getSheets()[0] ;  
var range = sheet.getRange("A1:D10") ;  
range.clearFormat() ;
```

6.18. appendRow()

Append rows to a spreadsheet

<https://developers.google.com/google-ads/scripts/docs/examples/spreadsheetapp#append-rows-to-a-spreadsheet>

appendRow(rowContents) - เมธอดในคลาส Sheet

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#appendrowrowcontents>

ใช้แนบแถวข้อมูล ต่อจากแถวสุดท้ายในชีต พารามิเตอร์ rowContents เป็นอาร์เรย์เส้น 1 แถว

ตัวอย่าง

```
var ssID = 'Your Spreadsheet Id' ;  
var sName = 'Sheet1' ;  
var ss = SpreadsheetApp.openById(ssID) ;  
var sheet = ss.getSheetByName(sName) ;  
sheet.appendRow([ 'a man' , 'a plan' , 'panama' ]) ;
```

ผล

	A	B	C
1	1	Wasan	456
2	2	Napabhorn	589
3	3	Tualek	123
4	4	Kaopod	568
5	a man	a plan	panama

6.19. โน้ต

6.19.ก.) setNote() และ setNotes()

setNote(note) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setnotenote>

setNote() ใช้กรอกโน้ตให้กับทุกเซลล์ในเรนจ์

setNotes(notes) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setnotesnotes>

setNotes() ใช้กรอกโน้ตให้กับแต่ละเซลล์ในเรนจ์ โดยพารามิเตอร์ **notes** เป็นข้อความโน้ต ในอาเรย์

2 มิติ ซึ่งจะต้องมีมิติเท่ากับเรนจ์

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;  
var sheet = ss.getSheets()[0] ;  
var cell = sheet.getRange("B2") ;  
cell.setNote("This is a note") ;
```

ผล

	A	B	C	D	E
1					
2					
3					
4					
5					
6					
7					

This is a note

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;

// The size of the two-dimensional array must match the size of the range.
var notes = [
    ["it goes" ,           "like this", "the fourth, the fifth" ] ,
    ["the minor fall" ,    "and the",   "major lift"           ]
] ; // มิติ 2x3
var cell = sheet.getRange("B2:D3") ;      // มิติ 2x3
cell.setNotes(notes) ;
```

ผล

	A	B	C	D	E	F	G
1							
2							
3				major lift			
4							
5							
6							
7							
8							

6.19.ข.) clearNote()

clearNote() - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#clearnote>

ใช้ล้างโน้ตออกจากเรนจ์

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;

var range = sheet.getRange("A1:D10") ;
range.clearNote() ;
```


บทที่ 7

กลุ่มของเร้นจ์



7.1. getRanges()

`getRanges()` - เมธอดในคลาส `RangeList`

<https://developers.google.com/apps-script/reference/spreadsheet/range-list#getranges>

ใช้จับเรนจ์ที่ถูกเลือก มักใช้กับการเลือกไว้หลายเรนจ์ โดยการกด **<Ctrl>+เลือกเรนจ์** โดยจะคืนค่ากลับมาเป็น `Range[]`-วัตถุ `Range` ที่เก็บอยู่ในอาร์เรย์ โดยมีจำนวนสมาชิกตามจำนวนเรนจ์ที่ถูกเลือก เช่น เลือกไว้ 3 เรนจ์ ก็จะคืนค่ากลับมาเป็น `[Range, Range, Range]` เป็นต้น

ดูตัวอย่างในข้อ 7.2 เรื่อง `getActiveRangeList()` หน้า 64

7.2. getActiveRangeList()

`getActiveRangeList()` - เมธอดในคลาส `Spreadsheet`

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#getactiverangelist>

ใช้จับเรนจ์ทั้งหมดที่เลือก โดยคืนค่ากลับมาเป็นคลาส `RangeList` ถ้าไม่มีเรนจ์ที่ถูกเลือก จะคืนค่า `null` ถ้าเลือกไว้เรนจ์เดียว `getActiveRangeList()` ใช้งานเหมือน `getActiveRange()` ฉะนั้น `getActiveRangeList()` มักใช้กับการเลือกไว้หลายเรนจ์

ตัวอย่าง

ก่อนรันโค้ดต่อไปนี้ ให้เลือกเรนจ์ในชีทไว้ก่อน โดยให้เลือกมากกว่า 1 เรนจ์ โดยกด **<Ctrl>+เลือกเรนจ์** จะเป็นการเลือกเพิ่มทีละบล็อก(หรือทีละเรนจ์)

	A	B	C	D	E
1		111	111		
2		111	111		333
3		111	111		333
4			222	222	
5			222	222	

โค้ด

```
var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
var rangeLists = sheet.getActiveRangeList();
var rangeLists_RANGES = rangeLists.getRanges();

Logger.log(rangeLists); // พิมพ์ : RangeList
Logger.log(rangeLists_RANGES); // พิมพ์ : [Range, Range, Range]

rangeLists_RANGES.forEach(function(range){
  Logger.log(range.getA1Notation()); // ดูผลที่ Logs ----- > [03],[04],[05]
}); // Close - forEach
```


ผล

Logs

```
[ 01 ] RangeList
[ 02 ] [Range, Range, Range]           // เลือกไว้ 3 เรนจ์
[ 03 ] B1:C3                           // เรนจ์ที่ 1
[ 04 ] C4:D5                           // เรนจ์ที่ 2
[ 05 ] E2:E3                           // เรนจ์ที่ 3
```

7.3. getRangeList()

getRangeList(a1Notations) - เมธอดในคลาส Spreadsheet

[https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#getrangelist\(a1notations\)](https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#getrangelist(a1notations))

ใช้จับเรนจ์ทั้งหมดที่เลือก โดยคืนค่ากลับมาเป็นวัตถุ **RangeList** พารามิเตอร์ที่ต้องระบุก็คือ **เรนจ์สตรง** โดยต้องระบุในลักษณะ **String['a1Notations','a1Notations','a1Notations', ...]**

ตัวอย่าง

```
var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet() ;
var rangeList = sheet.getRangeList(['A1:D4', 'F1:H4']) ;
var ranges = rangeList.getRanges() ;
ranges.forEach(function(r){
    Logger.log(r.getA1Notation())           // ดูผลที่ Logs ----- >
});
```

ผล

Logs

```
[] A1:D4
[] F1:H4
```

7.4. getSelection() และ getCurrentCell()

getSelection() - เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet-app#getselection>

getSelection() ใช้จับเรนจ์ที่เลือก โดยจะคืนค่ากลับมาเป็นคลาส **Selection** โดยภายในวัตถุ Selection ที่คืนค่ามา มีชนิดการเลือกอยู่ เช่น **Current cell**, **Active Range** และ **RangeList**

getCurrentCell() - เมธอดในคลาส - Selection

<https://developers.google.com/apps-script/reference/spreadsheet/selection#getcurrentcell>

คืน **Current cell** ในแอ็คทีฟเรนจ์กลับมา หรือ วัตถุในคลาส **Range**

คลาส Selection

<https://developers.google.com/apps-script/reference/spreadsheet/selection.html>

คลาส Selection ใช้เข้าถึงเรนจ์ที่แอ็คทีฟ โดยคลาส Selection เป็นกลุ่มของเซลล์ที่ผู้ใช้งานเลือก ซึ่งจะปรากฏเป็นไฮไลต์สีฟ้า(ตามภาพ) และมีได้หลายบล็อก

ตัวอย่างที่ 1 - ชีทที่ถูกเลือกไว้ 2 เรนจ์ หรือ **RangeList** มี 2 **Range** เซลล์ไฮไลต์สีฟ้ามี 1 เซลล์ ก็คือ **Current cell** เป็นเซลล์ที่อยู่ในโฟกัส (มีกรอบสีน้ำเงินเข้ม) หาก **Current cell** อยู่ที่เรนจ์ใด เรนจ์นั้นเป็น **Active Range**

	A	B	C	D	E
1		111	111		
2		111	111		333
3		111	111		333
4			222	222	
5			222	222	

```
var selection = SpreadsheetApp.getActiveSpreadsheet().getSelection();
var currentCell = selection.getCurrentCell() ; // D1
```

ตัวอย่างที่ 2 - ใช้การเลือกเหมือนตัวอย่างข้างบน

```
function test_getSelection() {

    var activeSheet = SpreadsheetApp.getActiveSheet() ; // จับชีทที่เปิดอยู่
    var rangeList = activeSheet.getRangeList(['A1:B4', 'D1:E4']) ; // จับ 2 เรนจ์
    rangeList.activate() ; // เลือก 2 เรนจ์

    var selection = activeSheet.getSelection();

    Logger.log('Current Cell : ' + selection.getCurrentCell().getA1Notation());
    // พิมพ์ : Current Cell : D1

    Logger.log('Active Range: ' + selection.getActiveRange().getA1Notation());
    // พิมพ์ : Active Range : D1:E4

    var ranges = selection.getActiveRangeList().getRanges();
    for (var i = 0; i < ranges.length; i++) {
        Logger.log('Active Ranges: ' + ranges[i].getA1Notation());
        // พิมพ์ : Active Ranges : A1:B4
        // พิมพ์ : Active Ranges : D1:E4
    }
    Logger.log('Active Sheet : ' + selection.getActiveSheet().getName());
    // พิมพ์ : (ชีทที่เปิดอยู่)
}
```

ผล

Logs

```
[ 01 ] Current Cell : D1
[ 02 ] Active Range : D1:E4
[ 03 ] Active Ranges : A1:B4
[ 04 ] Active Ranges : D1:E4
[ 05 ] Active Sheet : Sheet1
```

7.5. setActiveRange()

setActiveRange(**range**) – เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#setactiverangerange>

ใช้กำหนดเรนจ์ ให้เป็นแอ็คทีฟเรนจ์ (คล้ายการเลือกเรนจ์โดยใช้เมาส์) ซึ่งจากนั้นเราจะสามารถใช้เมธอดต่างๆกับแอ็คทีฟเรนจ์ได้ เช่น getCurrentCell() หรือ getActiveRange()

ตัวอย่าง

```
var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet() ;
var range = sheet.getRange('A1:D4') ;

sheet.setActiveRange(range) ;
var selection = sheet.getSelection() ;

// Current cell: A1
var currentCell = selection.getCurrentCell() ;

// Active Range: A1:D4
var activeRange = selection.getActiveRange() ;
```

7.6. setCurrentCell()

setCurrentCell(**cell**)– เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#setcurrentcellcell>

ใช้กำหนดให้เรนจ์ 1 เซลล์ เป็น Current cell.getDataValidation พารามิเตอร์ **cell** ก็คือ เรนจ์ของ 1 เซลล์

ตัวอย่าง

```
var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet() ;
var cell = sheet.getRange('B5') ;

sheet.setCurrentCell(cell) ;
var selection = sheet.getSelection() ;

// Current cell : B5
var currentCell = selection.getCurrentCell() ;
```

7.7. activate()

activate() - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#activate>

activate() ที่ใช้กำหนดให้เรนจ์เป็นตัวแอ็คทีฟ คล้ายกับการเลือกเมาส์ที่เรนจ์

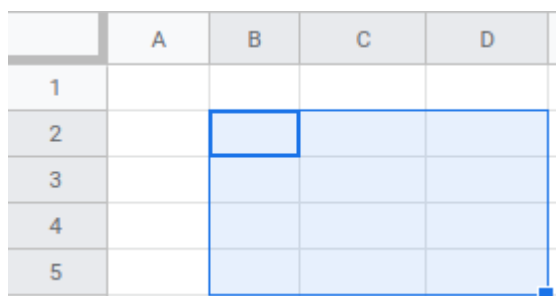
ตัวอย่าง

```
var sheet = SpreadsheetApp.getActiveSpreadsheet().getSheets()[4] ;
var range = sheet.getRange('B2:D5') ;
range.activate() ; // เลือกเรนจ์

var selection = sheet.getSelection() ;

var currentCell = selection.getCurrentCell() ; // Current cell: B2
var activeRange = selection.getActiveRange() ; // Active Range: A2:D5
```

ผล



	A	B	C	D
1				
2				
3				
4				
5				

7.8. getNextDataCell() และ getNextDataRange()

7.8.ก.) getNextDataCell()

getNextDataCell(direction) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#getnextdatacelldirection>

ใช้เลือกเรนจ์ โดยการเลือกเรนจ์แบบนี้ มีลักษณะเหมือนกับเลือกเรนจ์ไว้ แล้วกด <Ctrl><Arrow>

เริ่มจากเซลล์ที่มุมซ้ายบนของเรนจ์ ที่เลือกไว้แล้ว คำนวณค่ากลับมาเป็นเรนจ์ของเซลล์ ตามทิศทางที่กำหนด โดย เป็นเรนจ์ของเซลล์สุดท้ายที่มีข้อมูลต่อเนื่องกับเรนจ์เริ่มต้น หรือ คำนวณค่าเป็นเรนจ์ของเซลล์เริ่มต้นของข้อมูลในเรนจ์ถัดไปที่ข้อมูลขาดจากเรนจ์เริ่มต้น หรือ คำนวณค่าเป็นเรนจ์ของเซลล์ที่ขอบตารางคำนวณ

พารามิเตอร์ **direction** ก็คือ Enum Direction ดังตารางต่อไปนี้

<https://developers.google.com/apps-script/reference/spreadsheet/direction>

Property	Type	Description
UP	Enum	The direction of decreasing row indices.
DOWN	Enum	The direction of increasing row indices.
PREVIOUS	Enum	The direction of decreasing column indices.
NEXT	Enum	The direction of increasing column indices.

ตัวอย่าง

```
// สมมติว่าตารางคำนวณว่าง
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;
var range = sheet.getRange("C3:E5") ; // มิติ 3x3

// Logs "C1"
Logger.log(range.getNextDataCell(SpreadsheetApp.Direction.UP).getA1Notation()) ;

// Logs "A1"
Logger.log(range.getNextDataCell(SpreadsheetApp.Direction.PREVIOUS).getA1Notation()) ;
```

7.8.ข.) getNextDataRange()

getNextDataRange(direction) – เมธอดในคลาส Selection

<https://developers.google.com/apps-script/reference/spreadsheet/selection#getnextdatarangedirection>

ใช้เลือกเรนจ์ โดยการเลือกเรนจ์แบบนี้ มีลักษณะเหมือนกับ การเลือกเรนจ์เริ่มต้นไว้ จากนั้นกด
<Ctrl><Shift><Arrow> เพื่อเลือกเรนจ์เพิ่มเติมหรือเปลี่ยนไป ตามทิศทางของลูกศร

เริ่มต้นจากเรนจ์ที่เลือกไว้แล้ว มี Current cell และ Active Range และเคลื่อนไปตามทิศทางที่กำหนด

ตัวอย่าง

```
// สมมติว่าตารางคำนวณว่าง
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;

// เลือกเรนจ์ โดย C3 เป็น Current cell และ C3:E5 เป็น Active range.
sheet.getRange("C3:E5").activate() ;

// Logs "C1:E3"
Logger.log(SpreadsheetApp.getSelection()
            .getNextDataRange(Direction.UP).getA1Notation()) ;
```


บทที่ 8

รูปแบบ



8.1. setNumberFormat()

8.1.ก.) setNumberFormat()

Set cell number format

<https://developers.google.com/google-ads/scripts/docs/examples/spreadsheetapp#set-cell-number-format>

setNumberFormat(numberFormat) – เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setnumberformatnumberformat>

<https://developers.google.com/sheets/api/guides/formats> - Date and Number Formats

ใช้จัดรูปแบบตัวเลขหรือวันที่ในเซลล์ โดยใช้ Date and time format patterns (Format code) ถ้าจัดรูปแบบ

แบบเอง สามารถทำได้โดยใช้เมนู **Format** → **Number** → **More Formats** → **Custom number format**

เช่น

Date-time Patterns

mmmm d \[dddd\]	ตัวอย่างผลลัพธ์	April 5 [Tuesday]
dddd, m/d/yy at h:mm	ตัวอย่างผลลัพธ์	Tuesday, 4/5/16 at 16:08
hh:mm A/P".M."	ตัวอย่างผลลัพธ์	04:08 P.M.

Elapsed Time Patterns (เวลาที่ผ่านไป)

[hh]:[mm]:[ss].000	ตัวอย่างผลลัพธ์	03:13:41.255
---------------------------	-----------------	---------------------

ตัวอย่าง

```
var ssID = 'Spreadsheet Id' ;  
var ss = SpreadsheetApp.openById(ssID) ;  
var sheet = ss.getSheetByName('Sheet1') ;  
var cell = sheet.getRange('B2') ;  
cell.setNumberFormat('0.000') ;
```

ผล – เซลล์ B2 ตัวเลขเปลี่ยนเป็นทศนิยม 3 ตำแหน่ง

	A	B	C
1			
2		1.5	
3			

 →

	A	B	C
1			
2		1.500	
3			

8.1.ข.) setNumberFormats()

setNumberFormats(numberFormats) – เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setnumberformatsnumberformats>

เหมือนกันกับ **setNumberFormat()** เพียงแต่ใช้กับทุกเซลล์ในเรนจ์ โดยจะคืนค่ากลับมาเป็นวัตถุ

Range พารามิเตอร์ **numberFormats** เป็นอาร์เรย์ 2 มิติของ Date and time format patterns (**Object[][]**)

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;

// ขนาดมิติอาเรย์ของพารามิเตอร์ Formats จะต้องเท่ากับ Range
var formats = [ [ "0.000", "0,000,000", "$0.00" ] ] ; // 1 แถว 3 คอลัมน์
var range = sheet.getRange("B2:D2") ; // 1 แถว 3 คอลัมน์

range.setNumberFormats(formats) ;
```

ผล

	A	B	C	D
1				
2		12.000	12,345,678	\$10.56
3				

8.2. เส้นขอบ

setBorder(top,left,bottom,right,vertical,horizontal) - เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/range#setbordertop,-left,-bottom,-right,-vertical,-horizontal>

setBorder(top,left,bottom,right,vertical,horizontal,color,style) - เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/range#setbordertop,-left,-bottom,-right,-vertical,-horizontal,-color,-style>

พารามิเตอร์

Name	Type	Description
top	Boolean	เส้นขอบบน (true=ขีดเส้น false=เอาเส้นออก null=ปล่อยไว้เหมือนเดิม)
left	Boolean	เส้นขอบซ้าย (true=ขีดเส้น false=เอาเส้นออก null=ปล่อยไว้เหมือนเดิม)
bottom	Boolean	เส้นขอบล่าง (true=ขีดเส้น false=เอาเส้นออก null=ปล่อยไว้เหมือนเดิม)
right	Boolean	เส้นขอบขวา (true=ขีดเส้น false=เอาเส้นออก null=ปล่อยไว้เหมือนเดิม)
vertical	Boolean	เส้นกลางแนวตั้ง(เลือกหลายเซลล์) (true=ขีดเส้น false=เอาเส้นออก null=ปล่อยไว้เหมือนเดิม)
horizontal	Boolean	เส้นกลางแนวนอน(เลือกหลายเซลล์) (true=ขีดเส้น false=เอาเส้นออก null=ปล่อยไว้เหมือนเดิม)
color	String	โค้ดสีในสัญลักษณ์ CSS เช่น '#ffffff' or 'white' (หากเว้นไว้=black)
style	BorderStyle	รูปแบบของเส้น (หากเว้นไว้=solid)

Enum BorderStyle

Property	Type	Description
DOTTED	Enum	Dotted line borders.
DASHED	Enum	Dashed line borders.
SOLID	Enum	Thin solid line borders.
SOLID_MEDIUM	Enum	Medium solid line borders.
SOLID_THICK	Enum	Thick solid line borders.
DOUBLE	Enum	Two solid line borders.

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;
var cell = sheet.getRange("B2") ;

// ชิดเส้นขอบ บน-ล่าง แต่ปล่อย ซ้าย-ขวา ไว้เหมือนเดิม เส้นกลางตั้ง-นอน ไม่ชิด/เอาเส้นออก
cell.setBorder(true, null, true, null, false, false) ;
```

ตัวอย่าง - ชิดเส้นขอบสีแดงโดยรอบแอคทีฟเซลล์

```
var ac = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet().getActiveCell() ;
ac.setBorder(true,true,true,true,null,null,"Red", SpreadsheetApp.BorderStyle.SOLID ) ;
```

8.3. แบ็คกราวนด์

เมธอดกลุ่ม `setBackground` ใช้ระบายสีพื้นหลังให้เรนจ์ โดยระบุสีเป็น RGB หรือ CSS

8.3.ก.) `setBackground()` และ `setBackgrounds()`

`setBackground(color)` - เมธอดในคลาส `Range`

<https://developers.google.com/apps-script/reference/spreadsheet/range#setbackgroundcolor>

ใช้กำหนดสีของพื้นหลังให้กับเซลล์ในเรนจ์

`setBackgrounds(color)` - เมธอดในคลาส `Range`

<https://developers.google.com/apps-script/reference/spreadsheet/range#setbackgroundscolor>

ใช้กำหนดสีของพื้นหลังให้กับหลายเซลล์ในเรนจ์ โดยกำหนดให้แตกต่างกันได้

พารามิเตอร์

Name	Type	Description
<code>color</code>	String	โค้ดสีในสัญลักษณ์ CSS เช่น '#ffffff' or 'white' (หากเว้นไว้=black)

พารามิเตอร์ `color` ใช้เป็น Enum `ThemeColorType` ก็ได้

Enum `ThemeColorType`

<https://developers.google.com/apps-script/reference/spreadsheet/theme-color-type>

Property	Type	Description
<code>UNSUPPORTED</code>	Enum	Represents a theme color that is not supported.
<code>TEXT</code>	Enum	Represents the text color.
<code>BACKGROUND</code>	Enum	Represents the color to use for chart's background.
<code>ACCENT1</code>	Enum	Represents the first accent color.
<code>ACCENT2</code>	Enum	Represents the second accent color.
<code>ACCENT3</code>	Enum	Represents the third accent color.
<code>ACCENT4</code>	Enum	Represents the fourth accent color.
<code>ACCENT5</code>	Enum	Represents the fifth accent color.
<code>ACCENT6</code>	Enum	Represents the sixth accent color.
<code>HYPERLINK</code>	Enum	Represents the color to use for hyperlinks.

ตัวอย่างที่ 1

```
// ระบายพื้นหลังสีแดงให้กับเรนจ์ B2:D5
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;
var range = sheet.getRange("B2:D5") ;
range.setBackground("red") ;
```

ตัวอย่างที่ 2

```
// ระบายพื้นหลังให้กับเรนจ์ B2:D3 โดยใช้สีแตกต่างกัน
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;
var colors = [
    ["red",      "lime",    "blue"   ],
    ["#FF0000", "#00FF00", "#0000FF"]
] ;
// green เบอร์สีคือ #008000 เข้มกว่า #00FF00

var cell = sheet.getRange("B2:D3") ;
cell.setBackgrounds(colors) ;
```

ผล

	A	B	C	D
1				
2				
3				
4				

ตัวอย่างที่ 3 - ใส่สีเป็น Enum

```
var range = SpreadsheetApp.getActiveSpreadsheet().getRange('A1:B5') ;
range.setBackground(SpreadsheetApp.ThemeColorType.ACCEPT1) ;
```

8.3.ข.) setBackgroundRGB()

setBackgroundRGB(red, green, blue) - เมธอดในคลาส Range

[https://developers.google.com/apps-script/reference/spreadsheet/range.html#setBackgroundRGB\(red,-green,-blue\)](https://developers.google.com/apps-script/reference/spreadsheet/range.html#setBackgroundRGB(red,-green,-blue))

พารามิเตอร์

Name	Type	Description
red	Integer	ค่าของสีแดงในรูปแบบ RGB notation (ตัวเลข 0-255)
green	Integer	ค่าของสีเขียวในรูปแบบ RGB notation. (ตัวเลข 0-255)
blue	Integer	ค่าของน้ำเงินในรูปแบบ RGB notation. (ตัวเลข 0-255)

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;
var cell = sheet.getRange("B2") ;

// ระบายสีพื้นหลังเป็นสีขาว
cell.setBackgroundRGB(255, 255, 255) ;

// ระบายสีพื้นหลังเป็นสีแดง
cell.setBackgroundRGB(255, 0, 0) ;
```

8.3.ค.) ลิงค์เครื่องมือช่วยเลือกสี

Colors Tutorial
<https://www.w3schools.com/colors/default.asp>

HTML Color Picker
https://www.w3schools.com/colors/colors_picker.asp

Colors RGB
https://www.w3schools.com/colors/colors_rgb.asp

HTML Color Shades
https://www.w3schools.com/colors/colors_shades.asp

HTML Color Names
https://www.w3schools.com/colors/colors_names.asp

8.3.ง.) setBackgroundObject()

คลาส Color

<https://developers.google.com/apps-script/reference/spreadsheet/color>

setBackgroundObject(color) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setBackgroundObjectcolor>

ใช้ระบายสีพื้นหลังให้กับทุกเซลล์ในเรนจ์ โดยพารามิเตอร์ **color** เป็นวัตถุ color (ดูเพิ่มเติมคลาส Color)

setBackgroundObjects(color) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setBackgroundObjectscolor>

เหมือนกับ **setBackgroundObject()** เพียงแต่สามารถระบายสีในเรนจ์ให้แตกต่างกันได้ โดย

พารามิเตอร์ เป็นอาร์เรย์ของวัตถุสี (**Color[][]**) ซึ่งจะต้องแมทกับมิติของเรนจ์ที่จะระบายสีลงไป

newColor() - เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet-app#newcolor>

ตัวสร้างสำหรับสร้างวัตถุ Color ต้องจบด้วยเมธอด **build()** เพื่อสร้างวัตถุสีก่อนนำไปใช้งาน

ตัวอย่าง

```
var rgbColor = SpreadsheetApp.newColor().setRgbColor("#FF0000").build() ;
```

ตัวอย่าง - การใช้งาน setBackgroundObject()

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;
var bgColor = SpreadsheetApp.newColor()
    .setThemeColor(SpreadsheetApp.ThemeColorType.BACKGROUND)
    .build() ;
var range = sheet.getRange("B2:D5") ;
range.setBackgroundObject(bgColor) ;
```

ตัวอย่าง - การใช้งาน setBackgroundObjects()

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;

var colorAccent1 = SpreadsheetApp.newColor()
    .setThemeColor(SpreadsheetApp.ThemeColorType.ACCEPT1)
    .build() ;
var colorAccent2 = SpreadsheetApp.newColor()
    .setThemeColor(SpreadsheetApp.ThemeColorType.ACCEPT2)
    .build() ;
var colorAccent3 = SpreadsheetApp.newColor()
    .setThemeColor(SpreadsheetApp.ThemeColorType.ACCEPT3)
    .build() ;
var colorAccent4 = SpreadsheetApp.newColor()
    .setThemeColor(SpreadsheetApp.ThemeColorType.ACCEPT4)
    .build() ;

var colors = [
    [colorAccent1, colorAccent2],
    [colorAccent3, colorAccent4]
] ;

var cell = sheet.getRange("B2:C3") ;
cell.setBackgroundObjects(colors) ;
```

ผล

	A	B	C
1			
2			
3			
4			

8.4. กำหนดคุณลักษณะให้กับเร้นจ์

ข้อนี้เป็นวิธีที่ผู้เขียนใช้บ่อยในการเขียน Apps Script เพื่อตกแต่งเร้นจ์ เพราะง่าย แต่ว่า หากต้องจัดรูปแบบเซลล์เยอะๆ ซ้ำๆ เราจะต้องจัดระเบียบให้ดี ไม่ว่าจะเป็นการกำหนดสีที่เป็นมาตรฐาน หรือสร้างฟังก์ชันแยกเพื่อให้ง่ายต่อการเรียกใช้ เพื่อจัดรูปแบบเดิมซ้ำอีก ซึ่งจะสะดวกในการแก้ไขด้วย

8.4.ก.) setFontFamily() และ setFontFamilies()

setFontFamily(fontFamily) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setFontFamilyfontfamily>

ใช้กำหนดฟอนต์ให้กับทุกเซลล์ในเร้นจ์ โดยพารามิเตอร์ **fontFamily** เป็นชื่อฟอนต์

setFontFamilies(fontFamilies) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setFontFamiliesfontfamilies>

เหมือนกับ setFontFamily() เพียงแต่สามารถกำหนดให้แต่ละเซลล์ในเร้นจ์มีฟอนต์ที่แตกต่างกันได้ โดยพารามิเตอร์ **fontFamilies** เป็นชื่อฟอนต์ในอาเรย์ 2 มิติ ที่จะต้องแมทกับมิติของเร้นจ์

ตัวอย่างที่ 1

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;
var cell = sheet.getRange("B2") ;
cell.setFontFamily("Sarabun") ;
```

ตัวอย่างที่ 2

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;
var fonts = [                                     // 2x3
    ["Sarabun", "Prompt", "Kanit" ],
    ["Arial", "Sarabun", "Prompt" ]
];
var cell = sheet.getRange("B2:D3") ;             // 2x3
cell.setFontFamilies(fonts) ;
```

8.4.ข.) setFontSize() และ setFontSizes()

setFontSize(size) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setFontSize>

ใช้กำหนดขนาดของฟอนต์ให้กับทุกเซลล์ในเร้นจ์ โดยพารามิเตอร์ **size** เป็นเลขระบุขนาดของฟอนต์

setFontSizes(sizes) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setFontSizes>

เหมือนกับ setFontSize() เพียงแต่สามารถกำหนดให้แต่ละเซลล์ในเร้นจ์มีขนาดฟอนต์ที่แตกต่างกันได้

โดยพารามิเตอร์ **sizes** เป็นขนาดฟอนต์ในอาเรย์ 2 มิติ ที่จะต้องแมทกับมิติของเร้นจ์

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;
var cell = sheet.getRange("B2") ;
cell.setFontSize(20) ;
```

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;

// มิติอาร์เรย์ของ Sizes จะต้องเท่ากับมิติของเรนจ์
var fontSizes = [ 16, 20, 24 ] ; // 1x3
var range = sheet.getRange("B2:D2") ; // 1x3
range.setFontSizes(fontSizes) ;
```

8.4.ค.) setFontWeight() และ setFontWeights()

setFontWeight(fontWeight) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setFontWeightfontWeight>

ใช้กำหนดความหนาของฟอนต์ให้กับทุกเซลล์ในเรนจ์ โดยพารามิเตอร์ **fontWeight** เป็น String ดังต่อไปนี้ 'bold', 'normal' หรือ null (รีเซ็ตความหนา)

setFontWeights(fontWeights) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setFontWeightsfontWeights>

เหมือนกับ setFontWeight() เพียงแต่สามารถกำหนดให้แต่ละเซลล์ในเรนจ์มีความหนาของฟอนต์ที่แตกต่างกันได้ โดยพารามิเตอร์ fontWeights เป็นค่าความหนาฟอนต์ในอาร์เรย์ 2 มิติ ที่จะต้องเท่ากับมิติของเรนจ์

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;
var cell = sheet.getRange("B2") ;
cell.setFontWeight("bold") ;
```

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;

var fontStyles = [ [ "bold", "bold", "normal" ] ] ; // มิติ 1x3

var range = sheet.getRange("B2:D2") ; // มิติ 1x3
range.setFontWeights(fontStyles) ;
```

8.4.ง.) setFontStyle(fontStyle) และ setFontStyles(fontStyles)

setFontStyle(fontStyle) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setFontStylefontStyle>
ใช้กำหนดการเอียงของฟอนต์ให้กับทุกเซลล์ในเรนจ์ โดยพารามิเตอร์ fontStyle เป็น String ดังต่อไปนี้

'italic', 'normal' หรือ null (รีเซ็ตการเอียง)

setFontStyles(fontStyles) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setFontStylesfontStyles>
เหมือนกับ setFontStyle() เพียงแต่สามารถกำหนดให้แต่ละเซลล์ในเรนจ์มีการเอียงของฟอนต์ที่แตกต่างกันได้ โดยพารามิเตอร์ fontStyles เป็นค่าการเอียงในอาเรย์ 2 มิติ ที่จะต้องแมทกับมิติของเรนจ์

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;
var cell = sheet.getRange("B2") ;
cell.setFontStyle("italic") ;
```

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;

var fontStyles = [ ["italic", "normal"] ] ;           // มิติ 1x2
var range = sheet.getRange("B2:C2") ;               // มิติ 1x2
range.setFontStyles(fontStyles) ;
```

8.4.จ.) setFontColor() และ

setFontColor(color) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setFontColorcolor>
ใช้กำหนดสีให้กับฟอนต์ โดยพารามิเตอร์ color เป็นโค้ดสีในสัญลักษณ์ CSS เช่น '#ffffff' หรือ 'white'

setFontColors(colors) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setFontColorscolors>
เหมือนกับ setFontColor() เพียงแต่สามารถกำหนดสีให้กับฟอนต์ในแต่ละเซลล์ในเรนจ์ แตกต่างกันได้ โดยพารามิเตอร์ colors เป็นค่าสีในอาเรย์ 2 มิติ ที่จะต้องแมทกับมิติของเรนจ์

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;
var cell = sheet.getRange("B2") ;
cell.setFontColor("red") ;
```


ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;
var colors = [
    ["red", "white", "blue" ] ,
    ["#FF0000", "#FFFFFF", "#0000FF" ]
] ;
var cell = sheet.getRange("B5:D6") ;
cell.setFontColors(colors) ;
```

8.4.ฉ.) setHorizontalAlignment()

setHorizontalAlignment(**alignment**) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#sethorizontalalignmentalignment>

ใช้กำหนดการวางแนวข้อความในแนวนอน หรือ การจัดชิด ซ้าย-ขวา-เข้ากลาง โดยพารามิเตอร์

alignment เป็น String ดังต่อไปนี้ 'left', 'center', 'normal' หรือ null (รีเซ็ตแนวการวาง)

setHorizontalAlignments(**alignments**) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#sethorizontalalignmentsalignments>

เหมือนกับ setHorizontalAlignment() เพียงแต่สามารถกำหนดให้แต่ละเซลล์ในเร้นจมีการวางแนว

ข้อความ ที่แตกต่างกันได้ โดยพารามิเตอร์ **alignments** เป็นค่าการวางแนวในอาเรย์ 2 มิติ ที่จะต้องแมทกับมิติของเร้นจ

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;
var cell = sheet.getRange("B2") ;
cell.setHorizontalAlignment("center") ;
```

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;

var horizontalAlignments = [ [ "left", "right", "center" ] ] ; // 1x3
var range = sheet.getRange("B2:D2") ; // 1x3

range.setHorizontalAlignments(horizontalAlignments) ;
```

8.4.ช.) setVerticalAlignment()

setVerticalAlignment(**alignment**) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setverticalalignmentalignment>

ใช้กำหนดการวางแนวข้อความในแนวดิ่ง หรือ การจัดชิด บน-กลาง-ล่าง โดยพารามิเตอร์ **alignment**

เป็น String ดังต่อไปนี้ 'top', 'middle', 'bottom' หรือ null (รีเซ็ตแนวการวาง)

setVerticalAlignments([alignments](#)) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setverticalalignmentsalignments>

เหมือนกับ **setVerticalAlignment()** เพียงแต่สามารถกำหนดให้แต่ละเซลล์ในเรนจ์มีการวางแนว

ข้อความแนวดิ่ง ที่แตกต่างกันได้ โดยพารามิเตอร์ **alignments** เป็นค่าการวางแนวแนวดิ่งในอาเรย์ 2 มิติ ที่จะต้องแมทกับมิติของเรนจ์

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;

var cell = sheet.getRange("B2") ;
cell.setVerticalAlignment("middle") ;
```

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;

var alignments = [ [ "top", "middle", "bottom" ] ] ;
var range = sheet.getRange("B2:D2") ;
range.setVerticalAlignments(alignments) ;
```

8.4.ซ.) **setFontLine(fontLine)**

setFontLine([fontLine](#)) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setFontlinefontline>

ใช้กำหนดการขีดเส้นให้กับข้อความ โดยพารามิเตอร์ **fontLine** เป็น String ดังต่อไปนี้

'underline', 'line-through', 'none' หรือ null (รีเซ็ตการขีดเส้น)

setFontLines([fontLines](#)) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setFontlinesfontlines>

เหมือนกับ **setFontLine()** เพียงแต่สามารถกำหนดให้แต่ละเซลล์ในเรนจ์ มีการขีดเส้นที่แตกต่างกันได้

โดยพารามิเตอร์ **fontLines** เป็นค่าการขีดเส้นในอาเรย์ 2 มิติ ที่จะต้องแมทกับมิติของเรนจ์

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;
var cell = sheet.getRange("B2") ;
cell.setFontLine("line-through") ;
```

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;

var fontLines = [ [ "underline", "line-through", "none" ] ] ;
var range = sheet.getRange("B2:D2") ;
range.setFontLines(fontLines) ;
```

8.5. ตัวอย่างการกำหนดรูปแบบหลายอย่างให้กับเร้นจ์

โค้ด

```
var spreadsheet = SpreadsheetApp.getActive();  
spreadsheet.getActiveRangeList()  
    .setFontFamily('Sarabun')  
    .setFontSize(12)  
    .setFontColor("White")  
    .setHorizontalAlignment('center')  
    .setFontWeight('bold')  
    .setVerticalAlignment('middle')  
    .setBackground(SpreadsheetApp.ThemeColorType.ACCEPT1) ;
```

ผล – ก่อนรันโค้ด ให้เลือกเร้นจ์ไว้ก่อน เพราะการจัดรูปแบบจากโค้ดข้างต้น มีผลเฉพาะเร้นจ์ที่ถูกเลือกเท่านั้น

	A	B	C	D	E
1					
2					
3			1	6	11
4			2	7	12
5			3	8	13
6			4	9	14
7			5	10	15
8					

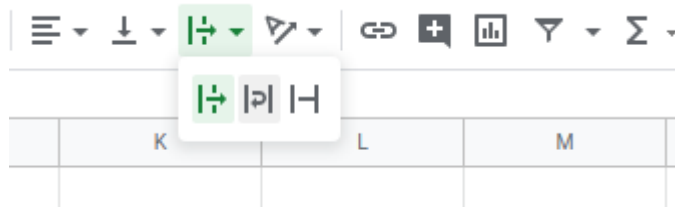
ตัวอย่างที่ 2 – จับค่ารูปแบบต่างๆของเร้นต์มา Logs ดู

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;  
var aSheet = ss.getActiveSheet() ;  
var aRange = aSheet.getActiveRange() ;  
var aValues = aRange.getValues() ;  
  
var aFontColor = aSheet.getRange(1,1,2,2).getFontColors() ;  
// [[#000000, #000000], [#000000, #000000]]  
var aFontFamilies = aSheet.getRange(1,1,2,2).getFontFamilies() ;  
// [[Sarabun, Sarabun], [Sarabun, Arial]]  
  
Logger.log(aFontColor) ;  
Logger.log(aFontFamilies) ;
```

8.6. Text wrapping

Text wrapping หรือ การห่อข้อความ ใช้กรณีที่ข้อความมีความยาว เกินความกว้างของเซลล์

Text wrapping หากกำหนดเอง สามารถทำได้โดยใช้ปุ่ม Text wrapping บนแถบเครื่องมือ ตัวอย่างตามภาพ



8.6.ก.) getWrapStrategy() และ getWrapStrategies()

getWrapStrategy() - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#getwrapstrategy>

คืนค่าเป็น text wrapping strategy (วิธีการห่อข้อความ) ของเซลล์ที่มุมซ้ายบนของเร้นจ์ที่จับมา โดยจะ

คืนค่ากลับมาเป็น Enum WrapStrategy (ดูตาราง Enum WrapStrategy)

getWrapStrategies() - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#getwrapstrategies>

คืนค่าเป็น Text wrapping strategies (วิธีการห่อข้อความ) ของทุกเซลล์ในเร้นจ์ โดยจะคืนค่ากลับมา

เป็น Enum WrapStrategy[][] หรือ Enum WrapStrategies ที่เก็บในอาร์เรย์ 2 มิติ (ดูตาราง Enum WrapStrategy)

ตัวอย่าง

```
// จับค่า Text wrapping strategy ของ B1 ออกมา Logs ดู
var sheet = SpreadsheetApp.getActiveSheet();
var range = sheet.getRange("B1:D4");

Logger.log(range.getWrapStrategy());
```

8.6.ข.) Enum WrapStrategy

Enum WrapStrategy

<https://developers.google.com/apps-script/reference/spreadsheet/wrap-strategy>

Property	Type	Description
WRAP	Enum	ข้อความที่ยาวเกินความกว้างของเซลล์ จะถูกตัดขึ้นบรรทัดใหม่อัตโนมัติ
OVERFLOW	Enum	ข้อความที่ยาวเกินความกว้างของเซลล์ จะแสดงทะลุไปยังเซลล์ถัดไป หากเซลล์ถัดไปไม่มีข้อมูลใดๆ แต่ถ้าเซลล์ถัดไปมีข้อมูลอยู่ ข้อความจะถูกตัด ให้แสดงเฉพาะส่วนที่อยู่ในความกว้างของเซลล์เท่านั้น
CLIP	Enum	ข้อความจะถูกตัด ให้แสดงเฉพาะส่วนที่อยู่ในความกว้างของเซลล์เท่านั้น

8.6.ค.) setWrapStrategy() และ setWrapStrategies()

setWrapStrategy(strategy)

<https://developers.google.com/apps-script/reference/spreadsheet/range#setwrapstrategystrategy>

ใช้เพื่อ Text wrapping strategy ให้กับทุกเซลล์ในเรนจ์ พารามิเตอร์ **strategy** ก็คือ Enum

WrapStrategy

setWrapStrategies()

<https://developers.google.com/apps-script/reference/spreadsheet/range#setwrapstrategiesstrategies>

ตัวอย่าง

```
// กำหนด Wrap strategy แบบ CLIP ให้กับทุกเซลล์ในเรนจ์ B2:D4
var sheet = SpreadsheetApp.getActiveSheet() ;
var range = sheet.getRange("B2:D4") ;
range.setWrapStrategy(SpreadsheetApp.WrapStrategy.CLIP) ;
```

ตัวอย่าง

```
// ก๊อปปี Wrap strategy จากเรนจ์ A1:B2 ไปให้กับเรนจ์ C5:D6
var sheet = SpreadsheetApp.getActiveSheet() ;
var range1 = sheet.getRange("A1:B2") ;
var range2 = sheet.getRange("C5:D6") ;
range2.setWrapStrategies(range1.getWrapStrategies()) ;
```

8.6.ง.) setWrap(isWrapEnabled) และ setWraps(isWrapEnabled)

setWrap(isWrapEnabled)

<https://developers.google.com/apps-script/reference/spreadsheet/range#setwrapiswrapenabled>

เพื่อ Cell wrap ให้กับเรนจ์ที่ระบุ ถ้า **เปิด(Enabled เป็นค่า Default)** เซลล์จะปรับขนาดให้เห็นเนื้อหาในเซลล์ทั้งหมด แต่ถ้า **ปิด(Disabled)** ก็จะแสดงให้มากที่สุด โดยไม่ปรับขนาดโดยการขึ้นบรรทัดใหม่หลายบรรทัด พารามิเตอร์ **isWrapEnabled** เป็นบูลีนกำหนดให้ Enabled หรือ Disabled

setWraps(isWrapEnabled)

<https://developers.google.com/apps-script/reference/spreadsheet/range#setwrapstrategiesstrategies>

Sets a rectangular grid of word wrap policies (must match dimensions of this range). Cells with wrap enabled (the default) resize to display their full content. Cells with wrap disabled display as much as possible in the cell without resizing or running to multiple lines.

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;
var cell = sheet.getRange("B2") ;
cell.setWrap(true) ;
```

ตัวอย่าง

```
var ss = SpreadsheetApp.getActiveSpreadsheet() ;
var sheet = ss.getSheets()[0] ;

// The size of the two-dimensional array must match the size of the range.
var wraps = [
    [ true, true, false ]
] ;

var range = sheet.getRange("B2:D2") ;
range.setWraps(wraps) ;
```

8.7. Text Style

8.7.ก.) newTextStyle() และ setTextStyle()

คลาส TextStyle

<https://developers.google.com/apps-script/reference/spreadsheet/text-style>

ใช้ระบายสไตล์ให้กับข้อความในเซลล์

newTextStyle() – เมธอดในคลาส SpreadsheetApp

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet-app#newtextstyle>

ใช้สร้าง ตัวสร้างรูปแบบข้อความ(Text Style) ซึ่งเป็นตัวกำหนดรูปแบบข้อความ เช่น ตัวหนา ขนาด

ฟอนต์ เป็นต้น โดยจะคืนค่ากลับมาเป็นวัตถุ RichTextValueBuilder

build() – เมธอดในคลาส TextStyleBuilder

ใช้สร้างรูปแบบข้อความจากตัวสร้าง

newTextStyle() และ **build()** ใช้คู่กับเริ่มต้นสร้างรูปแบบข้อความด้วยเมธอด **newTextStyle()** จบด้วย

เมธอด **build()** เพื่อสร้างรูปแบบข้อความที่พร้อมนำไป กำหนดให้กับเรนจ์

setTextStyle(style) – เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#settextstylestyle>

ใช้กำหนดสไตล์ให้กับข้อความ ในเรนจ์ที่ระบุ พารามิเตอร์ **style** ก็คือ วัตถุ TextStyle โดยคืนค่ากลับมา

เป็นวัตถุ **Range**

setTextStyles(styles) – เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#settextstylesstyles>

ใช้กำหนดสไตล์ให้กับข้อความ ในเรนจ์ที่ระบุ โดยจะแมทตามกริดของเรนจ์

ตัวอย่างที่ 1 : การใช้งาน `setTextStyle()`

```
// Sets the cells in range C5:D6 to have underlined size 15 font.

var sheet = SpreadsheetApp.getActiveSheet() ;
var range = sheet.getRange("C5:D6") ;

var style = SpreadsheetApp.newTextStyle()
    .setFontSize(15)
    .setUnderline(true)
    .build() ;

range.setTextStyle(style) ;
```

ตัวอย่างที่ 2 : การใช้งาน `setTextStyles()`

```
// Sets text styles for cells in range A1:B2
var sheet = SpreadsheetApp.getActiveSheet() ;
var range = sheet.getRange("A1:B2") ; // เรียงขนาด 2x2

var bold = SpreadsheetApp.newTextStyle()
    .setBold(true)
    .build() ;

var otherStyle = SpreadsheetApp.newTextStyle()
    .setBold(true)
    .setUnderline(true)
    .setItalic(true)
    .setForegroundColor("#335522")
    .setFontSize(44)
    .build() ;

range.setTextStyles([[bold, otherStyle], [otherStyle, bold]]) ; // Text Style ขนาด 2x2
```

8.7.ข.) เมธอดที่ใช้กำหนดรูปแบบของข้อความ

คลาส `TextStyleBuilder`

<https://developers.google.com/apps-script/reference/charts/text-style-builder>

`setBold(bold)` - เมธอดในคลาส `TextStyleBuilder`

<https://developers.google.com/apps-script/reference/spreadsheet/text-style-builder#setboldbold>

พารามิเตอร์ `bold` เป็น บูลีน

`setUnderline(underline)` - เมธอดในคลาส `TextStyleBuilder`

<https://developers.google.com/apps-script/reference/spreadsheet/text-style-builder#setunderlineunderline>

พารามิเตอร์ `underline` เป็น บูลีน

`setItalic(italic)` - เมธอดในคลาส `TextStyleBuilder`

<https://developers.google.com/apps-script/reference/spreadsheet/text-style-builder#setitalicitalic>

พารามิเตอร์ `italic` เป็น บูลีน

setForegroundColor(cssString) - เมธอดในคลาส TextStyleBuilder

<https://developers.google.com/apps-script/reference/spreadsheet/text-style-builder#setforegroundColorcssstring>

ใช้กำหนดสีให้กับฟอนต์ พารามิเตอร์ **cssString** เป็น ค่าสีในสัญลักษณ์ CSS เช่น **"#ffffff"** หรือ **"White"**

เป็นต้น

setForegroundColorObject(color)

<https://developers.google.com/apps-script/reference/spreadsheet/text-style-builder#setforegroundColorobjectcolor>

ใช้กำหนดสีให้กับฟอนต์ พารามิเตอร์ **color** เป็นวัตถุ Color

setFontSize(fontSize) - เมธอดในคลาส TextStyleBuilder

<https://developers.google.com/apps-script/reference/spreadsheet/text-style-builder#setfontsizefontsize>

พารามิเตอร์ **fontSize** เป็น เลขจำนวนเต็มขนาดของฟอนต์

setFontFamily(fontFamily) - เมธอดในคลาส TextStyleBuilder

<https://developers.google.com/apps-script/reference/spreadsheet/text-style-builder#setFontfamilyfontfamily>

พารามิเตอร์ **fontFamily** เป็นชื่อฟอนต์ เช่น Arial

setStrikethrough(strikethrough)

<https://developers.google.com/apps-script/reference/spreadsheet/text-style-builder#setstrikethroughstrikethrough>

พารามิเตอร์ **strikethrough** เป็น บูลีนว่าจะขีดเส้นคาคกลางข้อความหรือไม่

8.8. Rich Text Value

Rich Text Value หรือ ข้อความที่มีการจัดรูปแบบที่ต่างกันภายในเซลล์

คลาส RichTextValue

<https://developers.google.com/apps-script/reference/spreadsheet/rich-text-value>

ข้อความบางส่วนในเซลล์สามารถมีรูปแบบที่แตกต่างได้

รัน (ศัพท์เฉพาะ) ก็คือ ความยาวมากที่สุดที่อักขระที่เป็นส่วนหนึ่งของข้อความมีรูปแบบเดียวกัน ตัวอย่าง

"This kid *has two* apples." (สังเกตการจัดรูปแบบที่ต่างกัน) มี 4 รันก็คือ ["This ", "kid ", "has two ", "apples."]

คลาส RichTextValueBuilder

<https://developers.google.com/apps-script/reference/spreadsheet/rich-text-value-builder>

ตัวสร้าง Rich Text values

build() - เมธอดในคลาส RichTextValueBuilder

<https://developers.google.com/apps-script/reference/spreadsheet/rich-text-value-builder#build>

ใช้สร้าง Rich Text values จากตัวสร้าง Rich Text values

setText(text) - เมธอดในคลาส RichTextValueBuilder

<https://developers.google.com/apps-script/reference/spreadsheet/rich-text-value-builder#settexttext>

ใช้กำหนดข้อความ และล้างข้อความเดิมออกไป เมื่อสร้าง Rich Text values โดยเมธอดนี้ควรถูกเรียกใช้

ก่อนเมธอด **setTextStyle(startOffset, endOffset, textStyle)**

setTextStyle(textStyle) - เมธอดในคลาส RichTextValueBuilder

<https://developers.google.com/apps-script/reference/spreadsheet/rich-text-value-builder#settextstyletextstyle>

ใช้กำหนด Text Style ให้กับทั้ง Value

setTextStyle(startOffset, endOffset, textStyle) – เมธอดในคลาส RichTextValueBuilder
[https://developers.google.com/apps-script/reference/spreadsheet/rich-text-value-builder#setTextStyle\(startOffset,-endOffset,-textStyle\)](https://developers.google.com/apps-script/reference/spreadsheet/rich-text-value-builder#setTextStyle(startOffset,-endOffset,-textStyle))

ใช้กำหนด Text Style ให้กับบางส่วนของข้อความ โดยกำหนดระยะเริ่มต้นและจบได้ โดยอักษรตัวแรกเริ่มจาก 0

พารามิเตอร์

Name	Type	Description
startOffset	Integer	The start offset for the substring, inclusive.
endOffset	Integer	The end offset for the substring, exclusive.
textStyle	TextStyle	The text style being set.

ตัวอย่าง

```
// Creates a Rich Text value for the text "HelloWorld"
// with "Hello" bolded and italicized, and "World" only italicized.

var BOLD = SpreadsheetApp.newTextStyle().setBold(true).build() ;
var ITALIC = SpreadsheetApp.newTextStyle().setItalic(true).build() ;

var value = SpreadsheetApp.newRichTextValue()
    .setText("HelloWorld")
    .setTextStyle(0, 5, BOLD) // สไตล์ที่ 1
    .setTextStyle(ITALIC) // สไตล์ที่ 2
    .build() ;
```

ตัวอย่าง

```
// Creates a Rich Text value for the text "HelloWorld",
// with "Hello" bolded, and "World" italicized.

var BOLD = SpreadsheetApp.newTextStyle().setBold(true).build() ;
var ITALIC = SpreadsheetApp.newTextStyle().setItalic(true).build() ;

var value = SpreadsheetApp.newRichTextValue()
    .setText("HelloWorld")
    .setTextStyle(0, 5, BOLD)
    .setTextStyle(5, 10, ITALIC)
    .build() ;
```


บทที่ 9 การถือป้



9.1. copyTo() คลาส Range

copyTo() – เป็น Method ในหลายคลาส ใช้ก๊อปปี้ข้อมูล

copyTo([spreadsheet](#)) - เมธอดในคลาส Sheet

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#copytospreadsheet>

copyTo([destination](#)) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#copytodestination>

(กรณีอยู่ในคลาส Range) ใช้ก๊อปปี้ข้อมูลจากเรนจ์หนึ่งไปยังอีกเรนจ์หนึ่ง

9.1.ก.) copyTo(destination)

copyTo([destination](#)) - [destination](#) ก็คือ เรนจ์ปลายทาง

<https://developers.google.com/apps-script/reference/spreadsheet/range#copytodestination>

Copy data to cell range

<https://developers.google.com/google-ads/scripts/docs/examples/spreadsheetapp#copy-data-to-cell-range>

ตัวอย่าง

```
function copyData() {  
    var ssID = '1cjotusfCCq2bAkXqLYTCjowhOIMbVPWpnfL6qEZv16w';  
    var s1Name = 'Sheet1' ;  
    var s2Name = 'Sheet2' ;  
    var ss = SpreadsheetApp.openById(ssID);  
    var sheet1 = ss.getSheetByName(s1Name);  
    var sheet2 = ss.getSheetByName(s2Name);  
  
    var rangeToCopy = sheet1.getRange(1, 1, sheet1.getMaxRows(), 3);  
    // A1:C3 -----> E1:G3  
    rangeToCopy.copyTo(sheet2.getRange(1, 5));  
}
```

ผล - ข้อมูลจากเซลล์ A1:C3 ถูกก๊อปปี้ไปยังเซลล์ E1:G3

	A	B	C	D	E	F	G
1	Fruit	Cost	Quantity		Fruit	Cost	Quantity
2	Apple	15	200		Apple	15	200
3	Bananas	50	520		Bananas	50	520

9.1.ข.) copyTo(destination, options)

copyTo(destination, options)

<https://developers.google.com/apps-script/reference/spreadsheet/range#copytodestination,-options>

copyTo() โดยปกติจะก๊อปปี้ทั้งค่าและรูปแบบ แต่เราสามารถเลือกได้

Name	Type	Description
destination	Range	A destination range to copy to; only the top-left cell position is relevant.
options	Object	A JavaScript object that specifies advanced parameters, as listed below.

[options](#) เป็นตัววัตถุ มีชื่อ Key(หรือ Properties) ตามตารางดังนี้ ใช้เลือกเฉพาะรูปแบบหรือเนื้อหาในการก๊อปปี้

Name	Type	Description
formatOnly	Boolean	designates that only the format should be copied
contentsOnly	Boolean	designates that only the content should be copied

ตัวอย่าง – เมื่อรันโค้ดข้อมูลในชีต Sheet1 คอลัมน์ A:E ทุกแถว จะถูกก๊อปปี้ไปวางโดยเริ่มต้นที่เซลล์ F1 และก๊อปปี้มาเฉพาะข้อมูลไม่รวมรูปแบบ

```
function copyData() {  
  var ssID = 'Spreadsheet Id' ;  
  var sName = 'Sheet1' ;  
  var ss = SpreadsheetApp.openById(ssID) ;  
  var sheet = ss.getSheetByName(sName) ;  
  
  sheet.getRange("A:E")  
    .copyTo( sheet.getRange("F1"), { contentsOnly:true } ) ;  
  //           Range           Options  
}
```

ผล

	A	B	C	D	E	F	G	H	I
1	Fruit	Cost	Quantity	Total		Fruit	Cost	Quantity	Total
2	Apple	15	200	3,000		Apple	15	200	3,000
3	Bananas	50	520	26,000		Bananas	50	520	26,000
4	Mangoes	20	1,025	20,500		Mangoes	20	1,025	20,500
5	Mangoes	20	123	20,500		Mangoes	20	123	20,500
6	Durian	150	50	7,500		Durian	150	50	7,500
7	Durian	150	50	7,500		Durian	150	50	7,500
8	Longan	60	450	27,000		Longan	60	450	27,000
9	Total			132,500		Total			132,500

9.1.ค.) copyTo(destination, copyPasteType, transposed)

copyTo(destination, copyPasteType, transposed)

<https://developers.google.com/apps-script/reference/spreadsheet/range#copytodestination,-copypastetype,-transposed>

พารามิเตอร์

Name	Type	Description
destination	Range	A destination range to copy to; only the top-left cell position is relevant.
copyPasteType	CopyPasteType	A type that specifies how the range contents are pasted to the destination. (ดูตาราง Enum CopyPasteType)
transposed	Boolean	Whether the range should be pasted in its transposed orientation.

Enum CopyPasteType

<https://developers.google.com/apps-script/reference/spreadsheet/copy-paste-type.html>

Property	Type	Description
PASTE_NORMAL	Enum	Paste values, formulas, formats and merges.
PASTE_NO_BORDERS	Enum	Paste values, formulas, formats and merges but without borders.
PASTE_FORMAT	Enum	Paste the format and data validation only.
PASTE_FORMULA	Enum	Paste the formulas only.
PASTE_DATA_VALIDATION	Enum	Paste the data validation only.
PASTE_VALUES	Enum	Paste the values ONLY without formats, formulas or merges.
PASTE_CONDITIONAL_FORMATTING	Enum	Paste the color rules only.
PASTE_COLUMN_WIDTHS	Enum	Paste the column widths only.

ตัวอย่าง – เมื่อรันโค้ดข้อมูลในชีต Sheet1 คอลัมน์ A:E ทุกแถว จะถูกก๊อปปี้ไปวางโดยเริ่มต้นที่เซลล์ F1

```
function copyData() {  
    var ssID = 'Spreadsheet Id' ;  
    var sName = 'Sheet1' ;  
    var ss = SpreadsheetApp.openById(ssID) ;  
    var sheet1 = ss.getSheetByName(sName) ;  
  
    sheet1.getRange("A:E")  
        .copyTo(  
            sheet1.getRange("F1") ,  
            spreadsheetApp.CopyPasteType.PASTE_VALUES  
        ) ;  
}
```

	A	B	C	D	E	F	G	H	I
1	Fruit	Cost	Quantity	Total		Fruit	Cost	Quantity	Total
2	Apple	15	200	3,000		Apple	15	200	3,000
3	Bananas	50	520	26,000		Bananas	50	520	26,000
4	Mangoes	20	1,025	20,500		Mangoes	20	1,025	20,500
5	Mangoes	20	123	20,500		Mangoes	20	123	20,500
6	Durian	150	50	7,500		Durian	150	50	7,500
7	Durian	150	50	7,500		Durian	150	50	7,500
8	Longan	60	450	27,000		Longan	60	450	27,000
9	Total			132,500		Total			132,500

9.2. ก๊อปปี้รูปแบบไปที่เรนจ์

9.2.ก.) copyFormatToRange()

Copy formatting to cell range

<https://developers.google.com/google-ads/scripts/docs/examples/spreadsheetapp#copy-formatting-to-cell-range>

Get the last cell on a spreadsheet in which data is present

<https://developers.google.com/google-ads/scripts/docs/examples/spreadsheetapp#get-the-last-cell-on-a-spreadsheet-in-which-data-is-present>

`copyFormatToRange(sheet, column, columnEnd, row, rowEnd)`

<https://developers.google.com/apps-script/reference/spreadsheet/range#copyformattoangesheet,-column,-columnend,-row,-rowend>

ใช้ก๊อปปี้รูปแบบจากเรนจ์ต้นทาง ไปยังปลายทาง ถ้าปลายทางเล็กหรือใหญ่กว่าต้นทาง จะทำการวางให้

จนเต็ม

Name	Type	Description
<code>sheet</code>	Sheet	The target sheet.
<code>column</code>	Integer	The first column of the target range.
<code>columnEnd</code>	Integer	The end column of the target range.
<code>row</code>	Integer	The start row of the target range.
<code>rowEnd</code>	Integer	The end row of the target range.

ตัวอย่าง

```
function copyFormatToRange() {
  var ssID = 'Spreadsheet Id' ;
  var ss = SpreadsheetApp.openById(ssID) ;

  var ssName = 'Sheet1' ;
  var dsName = 'Sheet2' ;

  var sSheet = ss.getSheetByName(ssName) ;
  var dSheet = ss.getSheetByName(dsName) ;

  var sRange = sSheet.getRange('B2:D4');

  sRange.copyFormatToRange(dSheet, 4, 6, 4, 6) ;
}
```

เรนจ์ต้นทาง B2:D4 อยู่ในชีท Sheet1

	A	B	C	D	E
1					
2		Fruit	Cost	Quantity	Total
3		Apple	15	200	3,000
4		Bananas	50	520	26,000
5		Mangoes	20	1,025	20,500
6		Mangoes	20	123	20,500

หลังรันโค้ด ตำแหน่งปลายทางที่อยู่ในชีต Sheet2 รูปแบบเปลี่ยนไปตามภาพ

	A	B	C	D	E	F
1						
2		Fruit	Cost	Quantity	Total	
3		Apple	15	200	3000	
4		Bananas	50	520	26000	
5		Mangoes	20	1025	20,500	
6		Mangoes	20	123	20,500	
7		Durian	150	50	7500	
8		Durian	150	50	7500	

9.3. ก๊อปปี้ไฟล์ Google Sheets ด้วย copy()

9.3.ก.) copy()

Make a copy of a spreadsheet

<https://developers.google.com/google-ads/scripts/docs/examples/spreadsheetapp#make-a-copy-of-a-spreadsheet>

copy(name)

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#copynome>

ใช้ก๊อปปี้ไฟล์ Google Sheets คืนค่ากลับมาเป็นไฟล์ที่ถูกก๊อปปี้ พารามิเตอร์ name คือชื่อของไฟล์ใหม่

```
function copyASpreadsheet() {
    var ssID = 'Spreadsheet Id' ;
    var ss = SpreadsheetApp.openById(ssID) ;

    var newSS = ss.copy('Copy of ' + ss.getName()) ;

    Logger.log('New spreadsheet ID : %s.', newSS.getId()) ;
}
```


ผล – หลังรันโค้ดไฟล์ Google Sheets จะถูกก๊อปปี้ไปไว้ที่หน้าแรก Drive และถ้าดูที่ Log จะได้ผลดังต่อไปนี้

Logs

[] New spreadsheet ID : 1UD3DT9iENoK56tWJrqwe0Di-kxvYxkujwXXXXXXXXXX.

บทที่ 10
Banding หรือ
Alternating Colors



10.1. applyRowBanding()

คลาส Banding

<https://developers.google.com/apps-script/reference/spreadsheet/banding>

ใช้เข้าถึงและแก้ไข Banding (Alternating colors)

applyRowBanding(bandingTheme, showHeader, showFooter) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#applyrowbandingbandingtheme,-showheader,-showfooter>

applyRowBanding() ใช้ใส่ Alternating Colors ให้กับเร้นจ์ โดยคืนค่ากลับมาเป็นวัตถุ Banding

พารามิเตอร์

Name	Type	Description
bandingTheme	BandingTheme	ชื่อของ Alternating colors (ดูตาราง Enum BandingTheme)
showHeader	Boolean	บูลีน ที่กำหนดการแสดงผลส่วนหัวของ Alternating colors
showFooter	Boolean	บูลีน ที่กำหนดการแสดงผลส่วนท้ายของ Alternating colors

Enum BandingTheme

<https://developers.google.com/apps-script/reference/spreadsheet/banding-theme.html>

การเลือกสีให้พิมพ์ [SpreadsheetApp.BandingTheme.\(Property ตามตาราง\)](#)

Property	Type	Description
LIGHT_GREY	Enum	A light grey banding theme.
CYAN	Enum	A cyan banding theme.
GREEN	Enum	A green banding theme.
YELLOW	Enum	A yellow banding theme.
ORANGE	Enum	An orange banding theme.
BLUE	Enum	A blue banding theme.
TEAL	Enum	A teal banding theme.
GREY	Enum	A grey banding theme.
BROWN	Enum	A brown banding theme.
LIGHT_GREEN	Enum	A light green banding theme.
INDIGO	Enum	An indigo banding theme.
PINK	Enum	A pink banding theme.

ตัวอย่าง

```
var ss = SpreadsheetApp.getActive() ;
var sheet1 = ss.getSheetByName('Sheet1') ;
var lastRow = sheet1.getLastRow() ;
var lastColumn = sheet1.getLastColumn() ;
var dataRange = sheet1.getDataRange() ;      // จับข้อมูลใน sheet1 ทั้งหมด

dataRange.clearFormat() ;                    // ล้างรูปแบบเก่าทิ้งไปก่อน

// ใส่ Alternating Colors
dataRange.applyRowBanding(SpreadsheetApp.BandingTheme.GREEN,true,true) ;

// จัดรูปแบบตัวเลขให้กับคอลัมน์สุดท้าย
sheet1.getRange(2,lastColumn,lastRow-1,1).setNumberFormat("#,##0.00 บาท") ;

// แถวสุดท้ายตัวหนา
sheet1.getRange(lastRow,1,1,lastColumn).setFontWeight("Bold") ;
```

10.2. จับ Banding

`getBandings()` - เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#getbandings>

ใช้จับ Banding หรือ Alternating colors ทั้งหมดที่มีใน Spreadsheet คืนค่ากลับมาเป็น `Banding[]`-

Banding ในอาร์เรย์

`getBandings()` - เมธอดในคลาส Sheet

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#getbandings>

ใช้จับ Banding หรือ Alternating colors ทั้งหมดที่มีในชีต คืนค่ากลับมาเป็น `Banding[]`-Banding ใน

อาร์เรย์

`getBandings()` - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#getbandings>

ใช้จับ Banding หรือ Alternating colors ที่ใส่ให้กับเรนจ์ คืนค่ากลับมาเป็น `Banding[]`-Banding ในอาร์เรย์

อาร์เรย์

ดูตัวอย่างในข้อถัดไป

10.3. การลบ Banding หรือ Alternating Colors

`remove()` - เมธอดในคลาส Banding

<https://developers.google.com/apps-script/reference/spreadsheet/banding#remove>

ใช้ถอด Banding หรือ Alternating Colors ออก

ตัวอย่าง - โค้ดต่อไปนี้จะลบ Banding หรือ Alternating colors ทั้งหมดใน Spreadsheet ทั้งหมด

```
// วนลูปลบ Banding ทั้งหมดใน Spreadsheet ทั้งหมด
var spreadsheet = SpreadsheetApp.getActive() ;
spreadsheet.getBandings().forEach(function(banding) {
    banding.remove() ;
}) ;
```

ตัวอย่าง - โค้ดต่อไปนี้จะล้าง Alternating colors ทั้งหมดใน Spreadsheet ทั้งหมดอันเก่าออก จากเรนจ์ที่ใช้เมาส์เลือก แล้วใส่อันใหม่

```
// วนลูปลบ Banding ทั้งหมดใน Spreadsheet ทั้งหมด
var spreadsheet = SpreadsheetApp.getActive() ;
spreadsheet.getBandings().forEach(function (banding) {
    banding.remove() ;
}) ;

// ใส่ Banding ตัวใหม่ให้กับเรนจ์ที่เลือกโดยเมาส์ - สี Blue
spreadsheet.getActiveRange().applyRowBanding(SpreadsheetApp.BandingTheme.BLUE) ;
```


บทที่ 11

Data validation



11.1. สร้าง Data validation ให้กับเซลล์ในเรนจ์

ก่อนศึกษาบทนี้ ควรใช้งาน Data Validation ของ Google Sheets ให้คล่องก่อน จะทำให้เข้าใจได้ง่ายขึ้น

11.1.ก.) คลาสและเมธอดที่สำคัญ

ลิงค์ต่างๆ ต่อไปนี้เป็นคลาสและเมธอดที่สำคัญ หรือเป็นตัวอย่างที่จะใช้อธิบายโค้ดตัวอย่าง เพื่อสร้าง Data Validation ให้กับเซลล์หรือเรนจ์

Add data validation rule

<https://developers.google.com/google-ads/scripts/docs/examples/spreadsheetapp#add-data-validation-rule>

newDataValidation() - เมธอดในคลาส SpreadsheetApp

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet-app#newdatavalidation>

ใช้สร้างตัวสร้าง Data validation rule โดยจะคืนค่ากลับมาเป็นคลาส **DataValidationBuilder** โดย

Data validation rule จะถูกสร้างเมื่อจบด้วย Method **build()** ของคลาส **DataValidationBuilder**

คลาส DataValidationBuilder – เป็นคลาสที่ใช้สร้าง Data validation rules

<https://developers.google.com/apps-script/reference/spreadsheet/data-validation-builder.html>

build() - เมธอดในคลาส DataValidationBuilder

<https://developers.google.com/apps-script/reference/spreadsheet/data-validation-builder.html#build>

ใช้สร้าง Data validation rule จากการตั้งค่าที่ป้อนให้กับตัวสร้าง

requireNumberBetween(start, end) - เมธอดในคลาส TextValidationBuilder

<https://developers.google.com/apps-script/reference/forms/text-validation-builder#requirenumberbetweenstart,-end>

ใช้กำหนดกฎของ Data validation แบบตัวเลขที่ต้องอยู่ระหว่าง หรือ ตัวเลขแบบเจาะจง 2 ตัว

Name	Type	Description
start	Number	The lowest acceptable value.
end	Number	The highest acceptable value.

setAllowInvalid(allowInvalidData) - เมธอดในคลาส DataValidationBuilder

<https://developers.google.com/apps-script/reference/spreadsheet/data-validation-builder#setallowinvalidallowinvaliddata>

ใช้กำหนดว่าจะแสดงค่าเตือนหรือจะปฏิเสธไม่รับข้อมูล เมื่อไม่กรอกค่าลงในเซลล์ นอกเหนือกฎของ

Data validation (**allowInvalidData** เป็น **Boolean**)

setDataValidation(rule) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setdatavalidationrule>

ใส่ Data validation rule ให้กับเซลล์ โดยพารามิเตอร์ **rule** เป็น **Data Validation rule**

setDataValidations(rules) - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setdatavalidationsrules>

ใส่ Data validation rules ให้กับทุกเซลล์ในเรนจ์ โดยพารามิเตอร์ **rules** เป็น **Data Validation rules**

หลายตัวที่เก็บในอาร์เรย์ 2 มิติ (**Data Validation rule[][]**)

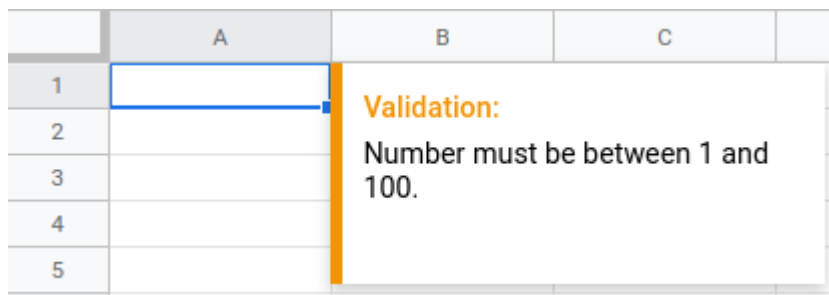
11.1.ข.) ตัวอย่างที่ 1

โค้ดต่อไปนี้จะใช้ Data Validation แบบ ตัวเลขต้องอยู่ระหว่าง ให้กับเซลล์ A1

```
var ss = SpreadsheetApp.openById('Spreadsheet Id') ;
var sheet = ss.getSheetByName('Sheet1') ;
var cell = sheet.getRange('A1') ; // จับเซลล์ A1

// สร้าง Rule จบด้วย .build() เสมอ >> นำ Rule ไป Set ให้เรนจ์
var rule = SpreadsheetApp.newDataValidation() // สร้าง Validation ตัวใหม่
    .requireNumberBetween(1,100) // รับเลข 1-100 เท่านั้น
    .setAllowInvalid(false) // คิยนอกกฎไม่รับข้อมูล
    .setHelpText('Number must be between 1 and 100.')
    .build() ;
cell.setDataValidation(rule) ;
```

ผล



11.1.ค.) ตัวอย่างที่ 2

โค้ดต่อไปนี้จะดึงข้อมูลจากเรนจ์ Sheet2!A1:A10 มาเป็นข้อยสในการทำ Data validation แบบ List from range (รายการจากเรนจ์) ให้กับทุกเซลล์ในเรนจ์ Sheet1!B1:B5

```
var srcRange = SpreadsheetApp.getActive().getSheetByName('Sheet2').getRange('A1:A10') ;
var dstRange = SpreadsheetApp.getActive().getSheetByName('Sheet1').getRange('A1:B5') ;

// สร้าง Data validation rule แบบรายการจากเรนจ์ ให้กับ srcRange
var rule = SpreadsheetApp.newDataValidation().requireValueInRange(srcRange).build() ;

// จับ Data validation ของเรนจ์ dstRange - ได้ null มาก่อน เพราะยังไม่มี
// จุดประสงค์จริงๆคือ สร้างอาร์เรย์ตามจำนวนของเซลล์ในเรนจ์ dstRange
// ถ้า Logs ดู จะได้ : [[null, null], [null, null], [null, null], [null, null], [null, null]]
// เป็นอาร์เรย์ 2 มิติขนาด 5 แถว 2 คอลัมน์
var rules = dstRange.getDataValidations() ;
for (var i = 0 ; i < rules.length ; i++) {
    for (var j = 0 ; j < rules[i].length ; j++) {
        rules[i][j] = rule ; // วนลูปจับ Data validation rule ใส่อาร์เรย์ rules
    }
}

dstRange.setDataValidations(rules) ; // ใส่ Data validation rules ให้กับทุกเซลล์ในเรนจ์
```

ข้อมูลใน **Sheet2!A1:A10** มีดังต่อไปนี้

	A
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

ผลหลังรันโค้ด เร็นจ์ **Sheet1!B1:B5** (5 แถว 2 คอลัมน์) ทำ Data Validation แบบ List form range โดยมีข้อมูลตามภาพ

	A	B
1		6 ▼
2	1	▼
3	2	8 ▼
4	3	▼
5	4	▼
6	5	
7	6	
8	7	
9	8	
10	9	
11	10	
12		
13		
14		

11.2. ดูและอัปเดต Data validation

11.2.ก.) getDataValidation()

getDataValidation() - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#getdatavalidation>

ใช้จับ Data validation ของเซลล์ หากเป็นเร้นจ์จะจับที่มุมซ้ายบนของเร้นจ์ โดยจะคืนค่ากลับมาเป็น

คลาส **DataValidation** หากเร้นจ์ไม่มี Data validation จะคืนค่า **null**

`getCriteriaType()` - เป็น เมธอดในคลาส `DataValidation`

<https://developers.google.com/apps-script/reference/spreadsheet/data-validation#getcriteriatype>

ใช้จับชนิดของ Rule's criteria ของ Data validation rule โดยคืนค่ากลับมาเป็น Enum

DataValidationCriteria

Enum `DataValidationCriteria`

<https://developers.google.com/apps-script/reference/spreadsheet/data-validation-criteria.html>

Property	Type	Description
<code>DATE_AFTER</code>	Enum	Requires a date that is after the given value.
<code>DATE_BEFORE</code>	Enum	Requires a date that is before the given value.
<code>DATE_BETWEEN</code>	Enum	Requires a date that is between the given values.
<code>DATE_EQUAL_TO</code>	Enum	Requires a date that is equal to the given value.
<code>DATE_IS_VALID_DATE</code>	Enum	Requires a date.
<code>DATE_NOT_BETWEEN</code>	Enum	Requires a date that is not between the given values.
<code>DATE_ON_OR_AFTER</code>	Enum	Require a date that is on or after the given value.
<code>DATE_ON_OR_BEFORE</code>	Enum	Requires a date that is on or before the given value.
<code>NUMBER_BETWEEN</code>	Enum	Requires a number that is between the given values.
<code>NUMBER_EQUAL_TO</code>	Enum	Requires a number that is equal to the given value.
<code>NUMBER_GREATER_THAN</code>	Enum	Require a number that is greater than the given value.
<code>NUMBER_GREATER_THAN_OR_EQUAL_TO</code>	Enum	Requires a number that is greater than or equal to the given value.
<code>NUMBER_LESS_THAN</code>	Enum	Requires a number that is less than the given value.
<code>NUMBER_LESS_THAN_OR_EQUAL_TO</code>	Enum	Requires a number that is less than or equal to the given value.
<code>NUMBER_NOT_BETWEEN</code>	Enum	Requires a number that is not between the given values.
<code>NUMBER_NOT_EQUAL_TO</code>	Enum	Requires a number that is not equal to the given value.
<code>TEXT_CONTAINS</code>	Enum	Requires that the input contains the given value.
<code>TEXT_DOES_NOT_CONTAIN</code>	Enum	Requires that the input does not contain the given value.
<code>TEXT_EQUAL_TO</code>	Enum	Requires that the input is equal to the given value.
<code>TEXT_IS_VALID_EMAIL</code>	Enum	Requires that the input is in the form of an email address.
<code>TEXT_IS_VALID_URL</code>	Enum	Requires that the input is in the form of a URL.
<code>VALUE_IN_LIST</code>	Enum	Requires that the input is equal to one of the given values.
<code>VALUE_IN_RANGE</code>	Enum	Requires that the input is equal to a value in the given range.
<code>CUSTOM_FORMULA</code>	Enum	Requires that the input makes the given formula evaluate to true.
<code>CHECKBOX</code>	Enum	Requires that the input is a custom value or a boolean; rendered as a checkbox.

`getCriteriaValues()` - เป็น เมธอดในคลาส `DataValidation`

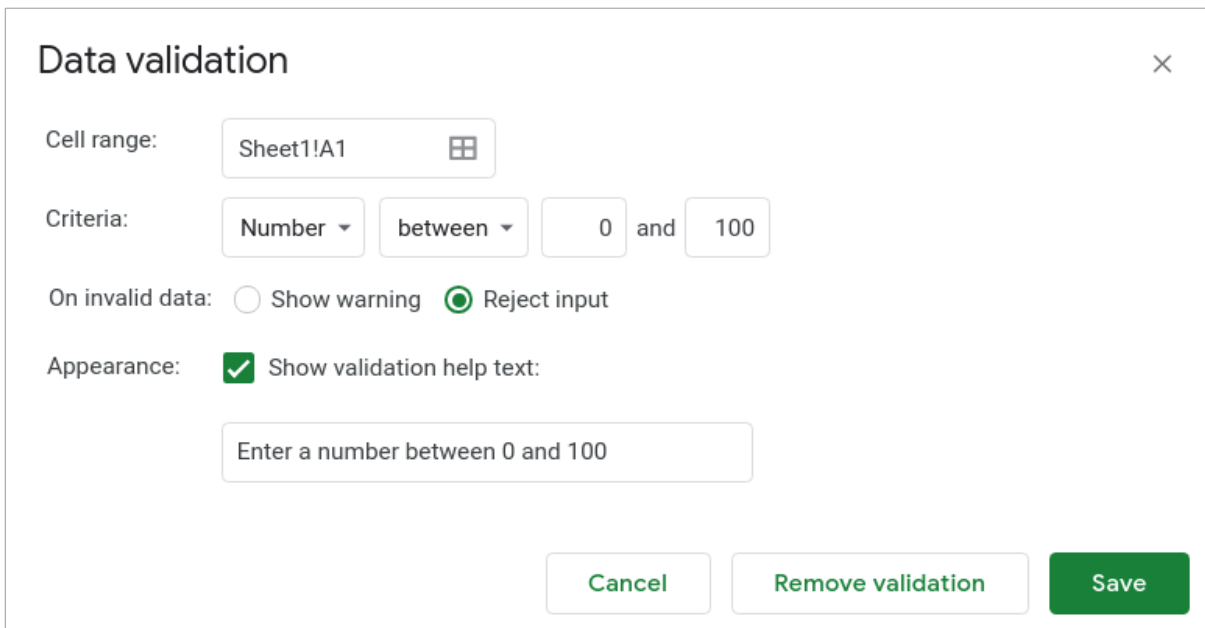
<https://developers.google.com/apps-script/reference/spreadsheet/data-validation#getcriteriavalues>

ใช้จับ Arguments ของ Rule's criteria (การตั้งค่าของ Data validation rule) โดยจะคืนค่ากลับมาเป็น

`Object[]` หรือ ก้อนอาเรย์ของ Arguments

ตัวอย่าง

เซลล์ A1 ทำ Validation ไว้ตามภาพ เป็น Validation ที่รับแต่ข้อมูลตัวเลขตั้งแต่ 0-100



โค้ดต่อไปนี้ ใช้จับ 1. Rule ของ Data validation และ 2. Arguments ของ Rule ออกมา Logs ดู

```
var cell = SpreadsheetApp.getActive().getRange('A1') ;
var rule = cell.getDataValidation() ; // จับ Data Validation

Logger.log(rule) ; // พิมพ์ : DataValidation (คืนค่าเป็นคลาส DataValidation)

if (rule != null) // ถ้าเซลล์ทำ Data validation ไว้
{
    var criteria = rule.getCriteriaType() ; // พิมพ์ : NUMBER_BETWEEN
    var args = rule.getCriteriaValues() ; // พิมพ์ : [0.0, 100.0]
    Logger.log('The data validation rule is %s %s', criteria, args) ; // ดูผลที่ Logs ---- >
}
else
{
    // ถ้าเซลล์ไม่ได้ทำ Data validation ไว้
    Logger.log('The cell does not have a data validation rule.') ; // หากเซลล์ไม่ทำ Validation
}
```

เมื่อรันโค้ด จะได้ผลลัพธ์ใน Logs ดังนี้

Logs

```
[ ] The data validation rule is NUMBER_BETWEEN [0.0, 100.0]
//                               criteria          args
```

11.2.ข.) getDataValidations() - มี s

Update data validation rules

<https://developers.google.com/google-ads/scripts/docs/examples/spreadsheetapp#update-data-validation-rules>

getDataValidations() - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#getdatavalidations>

ใช้จับ Data validation ของทุกเซลล์ในเรนจ์ คืนค่ากลับมาเป็นก้อนอาร์เรย์ของ Data validation ซึ่งเป็นอาร์เรย์ 2 มิติ (DataValidation[][]) หากเรนจ์ไม่มี Data Validation จะคืนค่า **null**

ตัวอย่าง

ชีท Sheet1 มีเพียง 15 เซลล์ตามภาพ

	A	B	C
1	2018-01-01		
2	2018-12-31		
3			
4			2018-06-15
5			

ชีท Sheet1 ทำ Validation ไว้ 3 เซลล์(ตามภาพข้างต้น) โดยกำหนด Rule และ Arguments ไว้ตามภาพต่อไปนี้ ก็คือ รับเฉพาะวันที่ระหว่าง 1 ม.ค. 2018 – 31 ธ.ค. 2018

Data validation

Cell range: Sheet1!A1

Criteria: Date between 1/1/2018 and 31/12/2018

On invalid data: ☐ Show warning ☒ Reject input

Appearance: ☒ Show validation help text:

Date between 1/1/2018 and 31/12/2018 [Reset](#)

Cancel

Remove validation

Save

โค้ดต่อไปนี้จะอัปเดตเซลล์ที่ทำ Validation ที่ใช้ Rule แบบ Date Between ใหม่ ให้เปลี่ยนไปรับวันที่
ระหว่าง วันที่ 1 ม.ค. 2019 – 31 ธ.ค. 2019

```
function updateDataValidationRules() {  
    var ssID = 'Spreadsheet Id' ;  
    var sName = 'Sheet1' ;  
    var ss = SpreadsheetApp.openById(ssID) ;  
    var sheet = ss.getSheetByName(sName) ;  
  
    // กฎเดิมรับเฉพาะวันที่ในปี 2018 เท่านั้น  
    // เราจะเปลี่ยน Arguments ของกฎให้รับวันที่ในปี 2019 เท่านั้น  
    var oldDates = [new Date('1/1/2018'), new Date('12/31/2018')] ;  
    var newDates = [new Date('1/1/2019'), new Date('12/31/2019')] ;  
  
    // จับเร้นจ์ทั้งหมดในชีท แม้จะเป็นเซลล์ว่างก็ตาม เพราะเซลล์ว่างอาจทำ Validation ไว้  
    var range = sheet.getRange(1, 1, sheet.getMaxRows(), sheet.getMaxColumns()) ;  
    Logger.log(rules) ; // ดูผลที่ Logs ---- >  
  
    // จับ Validations ทั้งหมดในเร้นจ์  
    var rules = range.getDataValidations() ;  
  
    for (var i = 0 ; i < rules.length ; i++) { // แถว // for #2  
        for (var j = 0 ; j < rules[i].length ; j++) { // คอลัมน์ในแถว // for #1  
            var rule = rules[i][j];  
            if (rule) { // ถ้าไม่ใช่ null // if - #2  
                var criteria = rule.getCriteriaType(); // จับ Type ของ DataValidation  
                var args = rule.getCriteriaValues(); // จับ Argument ของ DataValidation  
  
                // ถ้า Type เป็น DATE_BETWEEN และ  
                // Argument ตัวที่ 1 เป็นวันที่ 1/1/2018 และ  
                // Argument ตัวที่ 2 เป็นวันที่ 12/31/2018  
                if (criteria == SpreadsheetApp.DataValidationCriteria.DATE_BETWEEN && // if - #1  
                    args[0].getTime() == oldDates[0].getTime() &&  
                    args[1].getTime() == oldDates[1].getTime()) {  
                    // สร้างตัวสร้าง Validation จากกฎเดิม(copy) แต่เปลี่ยนวันที่  
                    rules[i][j] = rule.copy().withCriteria(criteria, newDates).build();  
                } // Close - if #1  
            } // Close - if #2  
        } // Close - for #1  
    } // Close - for #2  
  
    range.setDataValidations(rules);  
}  
// Close - function
```

```
ผลลัพธ์จากโค้ดบรรทัด var rules = range.getDataValidations();  

Logger.log(rules) ;
```

จับ Validation ของเซลล์ในเรนจ์ทั้งหมด มาใส่อาเรย์ 2 มิติ (ตัวแปร `rules`) โดยมีสมาชิกในอาเรย์เป็น **คลาส DataValidation**(หากเซลล์มี Validation) และ **null**(หากเซลล์ไม่มี Validation)

Logs

```
[ 01 ] [[ DataValidation , null , null ] , // แถวที่ 1  

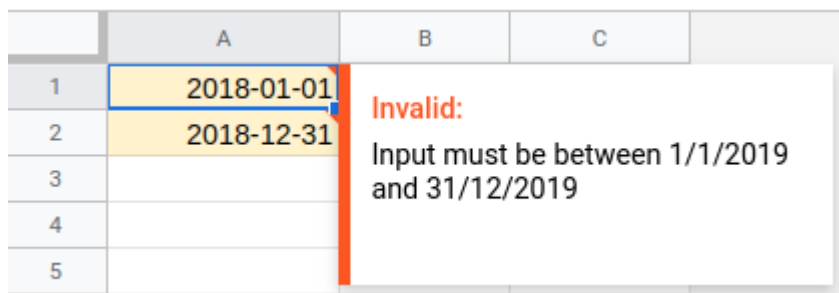
[ DataValidation , null , null ] , // แถวที่ 2  

[ null , null , null ] , // แถวที่ 3  

[ null , null , DataValidation ] , // แถวที่ 4  

[ null , null , null ] ] // แถวที่ 5
```

หลังรันโค้ด เซลล์ที่ทำ Validation ที่ใช้ Rule แบบ Date Between เปลี่ยน Arguments ไป (เปลี่ยนการตั้งค่าของกฎ) จึงปรากฏค่าเตือนตามภาพ เพราะค่าในเซลล์ตอนนี้ ไม่ได้เป็นไปตามกฎใหม่ของ Validation แล้ว



11.3. Checkboxes

11.3.ก.) insertCheckboxes()

`insertCheckboxes()` - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#insertcheckboxes>

`insertCheckboxes(checkedValue)` - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#insertcheckboxescheckedvalue>

`insertCheckboxes(checkedValue, uncheckedValue)` - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#insertcheckboxescheckedvalue-uncheckedvalue>

ใช้กรอก Checkboxes ลงแต่ละเซลล์ในเรนจ์

พารามิเตอร์

Name	Type	Description
<code>checkedValue</code>	Object	ค่าของ Checkbox เมื่อถูกเช็ค – ถ้าเว้นไว้จะเป็นบูลีน TRUE
<code>uncheckedValue</code>	Object	ค่าของ Checkbox เมื่อไม่ถูกเช็ค – ถ้าเว้นไว้จะเป็นบูลีน FALSE

ตัวอย่าง

```
function test_insert_cb(){  
  
  var range1 = SpreadsheetApp.getActive().getRange('A1:A5') ;  
  var range2 = SpreadsheetApp.getActive().getRange('B1:B5') ;  
  var range3 = SpreadsheetApp.getActive().getRange('C1:C5') ;  
  
  range1.insertCheckboxes('yes', 'no') ;  
  range2.insertCheckboxes('yes', 'no').setValues( [['no'], ['yes'], ['no'], ['yes'], ['yes']] ) ;  
  range3.insertCheckboxes().setValues( [[true], [true], [true], [true], [false]] ) ;  
}
```

ผล

	yes			
		A	B	C
1		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
4		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

11.3.ข.) requireCheckbox()

requireCheckbox() - เมธอดในคลาส DataValidationBuilder

<https://developers.google.com/apps-script/reference/spreadsheet/data-validation-builder#requirecheckbox>

requireCheckbox(**checkedValue**) - เมธอดในคลาส DataValidationBuilder

<https://developers.google.com/apps-script/reference/spreadsheet/data-validation-builder#requirecheckboxcheckedvalue>

requireCheckbox(**checkedValue**, **uncheckedValue**) - เมธอดในคลาส DataValidationBuilder

<https://developers.google.com/apps-script/reference/spreadsheet/data-validation-builder#requirecheckboxcheckedvalue-uncheckedvalue>

การใส่ Checkboxes ลงในเซลล์อีกวิธีหนึ่ง ก็คือ การทำ Validation แบบ Checkbox โดยใช้เมธอด
requireCheckbox()

ตัวอย่าง

```
var range = SpreadsheetApp.getActive().getRange('A1:A5') ;  
var rule = SpreadsheetApp.newDataValidation()  
    .setAllowInvalid(false)  
    .requireCheckbox()  
    .setHelpText('ให้ติ๊กเท่านั้น')  
    .build() ;  
  
range.setDataValidation(rule).setValue(true) ;
```


ผล

fx TRUE					
	A	B	C	D	
1	<input checked="" type="checkbox"/>				
2	<input checked="" type="checkbox"/>				
3	<input checked="" type="checkbox"/>				
4	<input checked="" type="checkbox"/>				
5	<input checked="" type="checkbox"/>				
6					

Validation:
ให้ติ๊กเท่านั้น

บทที่ 12

การเรียง (Sorting)



12.1. การเรียงข้อมูลโดยใช้เมธอด sort()

มีวิธีการเรียงข้อมูลหลากหลายวิธี เช่น ใช้เมธอด sort() ของอาร์เรย์ แต่ในบทนี้ เป็นการเรียงโดยใช้ เซอร์วิสของ Google ก็คือ เมธอด sort() ในคลาส Range และ เมธอด sort() ในคลาส Sheet

12.1.ก.) การเรียงข้อมูลในเรนจ์ (เมธอดในคลาส Range)

Sort a range of values by multiple columns

<https://developers.google.com/google-ads/scripts/docs/examples/spreadsheetapp#sort-a-range-of-values-by-multiple-columns>

sort(sortSpecObj) – เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#sortsortspecobj>

ใช้เรียงเซลล์ในเรนจ์ โดยคืนค่ากลับมาเป็นเรนจ์ที่เรียงข้อมูลแล้ว พารามิเตอร์ sortSpecObj ก็คือ คอลัมน์ที่จะใช้เป็นคีย์ในการเรียง มีวิธีการระบุพารามิเตอร์หลายแบบ ให้ดูจากตัวอย่างจะเข้าใจง่ายที่สุด

ตัวอย่างที่ 1

```
var ssID = 'Spreadsheet Id' ;
var sName = 'Sheet1' ;
var ss = SpreadsheetApp.openById(ssID) ;
var sheet = ss.getSheetByName(sName) ;

// จัปไปที่เรนจ์ที่จะเรียงข้อมูล ***
var range = sheet.getRange('A2:D8') ;

// เรียงตามคอลัมน์ที่ 1 (A)
range.sort(1) ;

// เรียงตามคอลัมน์ที่ 2 (B)
range.sort(2) ;

// เรียงตามคอลัมน์ที่ 2 (B) แบบ Descending
range.sort( {column: 2, ascending: false} ) ;

// เรียงตามคอลัมน์ที่ 2 (B) แบบ Descending
// จากนั้นเรียงต่อตามคอลัมน์ที่ 1 (A) แบบ Ascending
// สังเกต - ใส่พารามิเตอร์เป็นอาร์เรย์ ที่มีสมาชิกเป็นวัตถุ
range.sort([ { column: 2, ascending: false },
             { column: 1, ascending: true } ] ) ;

// เรียงตามคอลัมน์ที่ 2 (B) แบบ Descending
// จากนั้นเรียงต่อตามคอลัมน์ที่ 1 (A) แบบ Ascending
// ( ใส่ 1 ตัวเดียวคือ คอลัมน์ที่ 1 - ไม่ใส่ Sort order ค่า Default คือเรียงแบบ Ascending
range.sort([ { column: 2, ascending: false}, 1 ] ) ;

// เรียงตามคอลัมน์ที่ 2 (B) แบบ Ascending
// จากนั้นเรียงต่อตามคอลัมน์ที่ 1 (A) แบบ Ascending
range.sort([2, 1] ) ;

// ... โค้ดข้างบน ให้ผลลัพธ์เช่นเดียวกับโค้ดต่อไปนี้
range.sort([ { column: 2, ascending: true },
             { column: 1, ascending: true } ] ) ;
```

ตัวอย่างที่ 2

เร้นจ์ก่อนเรียงข้อมูล มีข้อมูลดังต่อไปนี้

	A	B	C	D	E
1	Order	Fruit	Unit Price	Quantity	Total
2	1	Apple	15	200	3,000
3	2	Bananas	50	520	26,000
4	3	Mangoes	20	1,025	20,500
5	4	Mangoes	20	123	2,460
6	5	Durian	150	560	84,000
7	6	Durian	150	50	7,500
8	7	Longan	60	450	27,000

โค้ดต่อไปนี้เรียงข้อมูลในเร้นจ์ **A2:E8** โดยใช้คอลัมน์ B (Fruit) เป็นคีย์ที่ 1 จากนั้นเรียงต่อโดยใช้คอลัมน์ D (Quantity) เป็นคีย์ที่ 2

```
var ssID = 'Spreadsheet Id' ;  
var sName = 'Sheet1' ;  
var ss = SpreadsheetApp.openById(ssID) ;  
var sheet = ss.getSheetByName(sName) ;  
  
var range = sheet.getRange('A2:E8') ; // เร้นจ์ที่จะเรียง  
  
range.sort([ {column: 2, ascending: true } ,  
             {column: 4, ascending: false } ]) ;
```

ผล - หลังรันโค้ดได้ผลลัพธ์ดังต่อไปนี้

	A	B	C	D	E
1	Order	Fruit	Unit Price	Quantity	Total
2	1	Apple	15	200	3,000
3	2	Bananas	50	520	26,000
4	5	Durian	150	560	84,000
5	6	Durian	150	50	7,500
6	7	Longan	60	450	27,000
7	3	Mangoes	20	1,025	20,500
8	4	Mangoes	20	123	2,460

12.1.ข.) การเรียงข้อมูลในชีท (เมธอดในคลาส Sheet)

Sort a spreadsheet by a specified column

<https://developers.google.com/google-ads/scripts/docs/examples/spreadsheetapp#sort-a-spreadsheet-by-a-specified-column>

`sort(columnPosition, ascending)` – เป็น เมธอดในคลาส Sheet

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#sortcolumnposition,-ascending>

`sort()` ใช้เรียงข้อมูลทั้งหมดในชีท โดยใช้คอลัมน์ที่ระบุเป็นคีย์ในการเรียง

พารามิเตอร์

Name	Type	Description
<code>columnPosition</code>	Integer	The <code>column</code> to sort by.
<code>ascending</code>	Boolean	<code>true</code> for ascending sorts, <code>false</code> for descending.

ตัวอย่าง

```
var ssID = 'Spreadsheet Id' ;  
var sName = 'Sheet1' ;  
var ss = SpreadsheetApp.openById(ssID) ;  
var sheet = ss.getSheetByName(sName) ;  
  
sheet.sort(1, true) ;
```

บทที่ 13
ฟิลเตอร์
(ตัวกรอง)



13.1. คลาสที่เกี่ยวข้องกับฟิลเตอร์

คลาส **Filter** (คลาสตัวกรอง)

<https://developers.google.com/apps-script/reference/spreadsheet/filter>

เข้าถึงและแก้ไขฟิลเตอร์ที่มีอยู่แล้ว หากต้องการสร้างฟิลเตอร์ใหม่ ให้ใช้ `Range.createFilter()`

คลาส **FilterCriteria** (หรือกฎการกรอง)

<https://developers.google.com/apps-script/reference/spreadsheet/filter-criteria>

ใช้เข้าถึงกฎของฟิลเตอร์(กฎการกรอง) การสร้างกฎใหม่ใช้ `SpreadsheetApp.newFilterCriteria()` และ

คลาส **FilterCriteriaBuilder**

คลาส **FilterCriteriaBuilder** (ตัวสร้างกฎการกรอง)

<https://developers.google.com/apps-script/reference/spreadsheet/filter-criteria-builder>

ตัวสร้างสำหรับ **FilterCriteria**

13.2. การจับฟิลเตอร์

13.2.ก.) เมธอดที่เกี่ยวข้อง

getFilter() - เมธอดในคลาส **Sheet**

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#getfilter>

ใช้จับฟิลเตอร์ที่มีอยู่ในชีต หากมีฟิลเตอร์จะคืนค่าเป็น **คลาส Filter** (คลาสตัวกรอง) หรือ **null** หากไม่มี

getColumnFilterCriteria(columnPosition) - เมธอดในคลาส **Filter**

<https://developers.google.com/apps-script/reference/spreadsheet/filter#getcolumnfiltercriteriacolumnposition>

ใช้จับกฎการกรองแบบเจาะจงคอลัมน์ โดยคืนค่ากลับมาเป็น **คลาส FilterCriteria(กฎการกรอง)** หาก

ไม่มีกฎการกรองจะคืนค่าเป็น **null** พารามิเตอร์ **columnPosition** ก็คือ ตำแหน่งของคอลัมน์

*** ตัวนี้มีทั้ง `getColumnFilterCriteria`, `setColumnFilterCriteria` และ `removeColumnFilterCriteria`

getHiddenValues() - เมธอดในคลาส **FilterCriteriaBuilder**

<https://developers.google.com/apps-script/reference/spreadsheet/filter-criteria-builder#gethiddenvalues>

เมื่อมีการกำหนดกฎการกรอง `getHiddenValues()` ใช้จับค่าที่ซ่อนจากการกรอง โดยคืนค่ากลับมาเป็น

String[]- อาร์เรย์ของข้อความ

getVisibleValues() - เมธอดในคลาส **FilterCriteriaBuilder** - ยังใช้งานไม่ได้

<https://developers.google.com/apps-script/reference/spreadsheet/filter-criteria-builder#getvisiblevalues>

เมื่อมีการกำหนดกฎการกรอง `getVisibleValues()` ใช้จับค่าที่แสดงในผลลัพธ์การกรอง โดยคืนค่ากลับ

มาเป็น **String[]**- อาร์เรย์ของข้อความ

getRange() - เมธอดในคลาส **Filter**

<https://developers.google.com/apps-script/reference/spreadsheet/filter#getrange>

ใช้จับเรนจ์ที่ถูกใส่ฟิลเตอร์ โดยคืนค่ากลับมาเป็น **คลาส Range**

13.2.ข.) ตัวอย่างที่ 1 : จับฟิเตอร์ที่กรองแบบ Filter by values ไว้อย่างเดียว

ตารางข้อมูลต่อไปนี้ ใส่ฟิเตอร์ไว้โดยใช้วิธีปกติ ก็คือ เลือกจากเมนู Data → Create a filter จากนั้น กรองคอลัมน์ "หมวด" โดยตั้งกรองเฉพาะค่าในหมวดที่เป็น OTC เท่านั้น

	A	B	C	D
1	รหัส	หมวด	สินค้าหรือบริการ	ราคาต่อหน่วย
2	OTC0020	OTC		100
3	OTC0019	OTC		50
28				
29				
30				
31				
32				
33				
34				
35				
36				
37				
38				
39				
40				
41				

ผลการกรอง

	A	B	C	D
1	รหัส	หมวด	สินค้าหรือบริการ	ราคาต่อหน่วย
2	OTC0020	OTC	Test 2	100
3	OTC0019	OTC	Test 1	50

ตัวอย่าง – โค้ดต่อไปนี้ จับคุณสมบัติของตัวกรองออกมา Logs ดู

```

var sheet = SpreadsheetApp.getActiveSheet() ;
var filter = sheet.getFilter() // จับฟิเตอร์ในชีทที่ระบุ
Logger.log(filter) ; // พิมพ์ : filter (วัตถุ)

var filterCriteria = filter.getColumnFilterCriteria(2) ; // จับกฎในคอลัมน์ที่ 2
Logger.log(filterCriteria) ; // พิมพ์ : filterCriteria (วัตถุ)

var hiddenValues = filterCriteria.getHiddenValues() ;
Logger.log(hiddenValues) ; // พิมพ์ : [DCC, DGG] - ถูกซ่อนอยู่

```

13.2.ค.) ตัวอย่างที่ 2 : จับฟิเตอร์ที่กรองแบบ Filter by values และ Filter by condition

ตารางข้อมูลต่อไปนี้ ใส่ฟิเตอร์ไว้โดยใช้วิธีปกติ ก็คือ เลือกจากเมนู Data → Create a filter จากนั้น กรองคอลัมน์ "สินค้าหรือบริการ" (คอลัมน์ที่ 3) โดยกำหนดกฎการกรองไว้ทั้งแบบ

Filter by condition ก็คือ Text contain - "ส่วนลด" และ

Filter by values ก็คือ ตี๊กที่ "ส่วนลด 10 บาท", "ส่วนลด 100 บาท", "ส่วนลด 150 บาท"

	A	B	C	D
1	รหัส	หมวด	สินค้าหรือบริการ	ราคาต่อหน่วย
15	DCC			150
16	DCC			100
25	DCC			10
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				
36				
37				
38				
39				
40				
41				
42				
43				
44				

ผลการกรอง

	A	B	C	D
1	รหัส	หมวด	สินค้าหรือบริการ	ราคาต่อหน่วย
15	DCC0150	DCC	ส่วนลด 150 บาท	150
16	DCC0100	DCC	ส่วนลด 100 บาท	100
25	DCC0010	DCC	ส่วนลด 10 บาท	10
26				

ตัวอย่าง – โค้ดต่อไปนี้ จับคุณสมบัติของตัวกรองออกมา Logs ดู

```
var sheet = SpreadsheetApp.getActiveSheet() ;
var filter = sheet.getFilter() ; // จับฟิลเตอร์ในชีทที่ระบุ
var filterCriteria = filter.getColumnFilterCriteria(3) ; // จับกฎการกรองในคอลัมน์ที่ 3
Logger.log(filterCriteria) ; // ดูผลที่ Logs ----- > [01]

Logger.log(criteria.getCriteriaType()) ; // ดูผลที่ Logs ----- > [02]
Logger.log(criteria.getCriteriaValues()) ; // ดูผลที่ Logs ----- > [03]
Logger.log(criteria.getHiddenValues()) ; // ดูผลที่ Logs ----- > [04]
Logger.log(criteria.getVisibleValues()) ; // ดูผลที่ Logs ----- > [05]
```

ผล

```
[ 01 ] FilterCriteria
[ 02 ] TEXT_CONTAINS
[ 03 ] [ส่วนลด]
[ 04 ] [ส่วนลด 20 บาท, ส่วนลด 200 บาท, ส่วนลด 250 บาท, ส่วนลด 30 บาท, ส่วนลด 300 บาท,
ส่วนลด 350 บาท, ส่วนลด 40 บาท, ส่วนลด 400 บาท, ส่วนลด 450 บาท, ส่วนลด 50 บาท, ส่วนลด
500 บาท, ส่วนลด 60 บาท, ส่วนลด 70 บาท, ส่วนลด 80 บาท, ส่วนลด 90 บาท]
[ 05 ] []
```

13.3. การสร้างฟิลเตอร์

13.3.ก.) เมธอดที่เกี่ยวข้อง

createFilter() - เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#createfilter>

ใช้สร้างฟิลเตอร์ให้กับเรนจ์ที่จับมา ซึ่งใน 1 ชีทมีฟิลเตอร์ได้เพียง 1 เท่านั้น **createFilter()** คืนค่ากลับมา

เป็นคลาส **Filter**

newFilterCriteria() - เมธอดในคลาส SpreadsheetApp

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet-app#newfiltercriteria>

ใช้สร้างตัวสร้างสำหรับ **คลาส FilterCriteria(กฎการกรอง)** โดยคืนค่ากลับมาเป็นคลาส

FilterCriteriaBuilder

setColumnFilterCriteria(columnPosition, filterCriteria) - เมธอดในคลาส Filter

<https://developers.google.com/apps-script/reference/spreadsheet/filter#setcolumnfiltercriteriacolumnposition,-filtercriteria>

ใช้กำหนดกฎของฟิลเตอร์(กฎการกรอง) ให้กับคอลัมน์แบบเจาะจง

พารามิเตอร์

Name	Type	Description
columnPosition	Integer	The 1-indexed position of the column.
filterCriteria	FilterCriteria	The filter criteria to set. If null , remove filter criteria from the specified column; alternatively, use removeColumnFilterCriteria(columnPosition)

setHiddenValues(values) - เมธอดในคลาส FilterCriteriaBuilder

<https://developers.google.com/apps-script/reference/spreadsheet/filter-criteria-builder#sethiddenvaluesvalues>

ใช้กำหนดค่าที่จะซ่อน(แสดงค่าอื่นๆไว้) โดยพารามิเตอร์ **values** เป็น **String[]** - อาร์เรย์ของค่าที่จะซ่อน

โดยจะคืนค่ากลับมาเป็นคลาส **FilterCriteriaBuilder**

setVisibleValues(values) - ยังใช้งานไม่ได้ แต่ในเอกสารเผยแพร่ของ Google มีแล้ว

<https://developers.google.com/apps-script/reference/spreadsheet/filter-criteria-builder#setvisiblevaluesvalues>

ใช้กำหนดค่าที่จะแสดง(ซ่อนค่าอื่นๆไป) โดยพารามิเตอร์ **values** เป็น **String[]** - อาร์เรย์ของค่าที่จะ

แสดง โดยจะคืนค่ากลับมาเป็นคลาส **FilterCriteriaBuilder**

build() - เมธอดในคลาส FilterCriteriaBuilder

<https://developers.google.com/apps-script/reference/spreadsheet/filter-criteria-builder?hl=lt#build>

สร้างกฎการกรองจากการตั้งค่าต่างๆที่กำหนดไว้ โดยจะคืนค่ากลับมาเป็นคลาส **FilterCriteriaBuilder** -

หลังจากเรียงตั้งการกฎการกรองต่างๆแล้ว ต้องจบด้วย **.build()** เสมอ

13.3.ข.) ตัวอย่างที่ 1 ใส่ฟิลเตอร์ให้กับเร็นจ์

ตัวอย่างต่อไปนี้ สร้างฟิลเตอร์ให้กับเร็นจ์ แต่ยังไม่ได้กำหนดกฎการกรอง ให้ผลเหมือนกับการเลือกเร็นจ์
แล้วไปที่เมนู **Data → Create a filter**

```
function FilterSheet() {  
    var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
    sheet.getRange('A1').getDataRegion().createFilter();  
}
```

ผลลัพธ์ - สังเกตที่หัวตารางมีสัญลักษณ์ตัวกรองอยู่

	A	B	C	D
1	รหัส	หมวด	สินค้าหรือบริการ	ราคาต่อหน่วย
2	OTC0020	OTC	Test 2	100
3	OTC0019	OTC	Test 1	50
4	DGG4050	DGG	อาบน้ำและตัดขนหมา (40-50 กก.)	1200
5	DGG3040	DGG	อาบน้ำและตัดขนหมา (30-40 กก.)	1000
6	DGG2030	DGG	อาบน้ำและตัดขนหมา (20-30 กก.)	750
7	DGG1520	DGG	อาบน้ำและตัดขนหมา (15-20 กก.)	550
8	DGG0715	DGG	อาบน้ำและตัดขนหมา (7-15 กก.)	350
9	DGG0017	DGG	อาบน้ำและตัดขนหมา (1- 7 กก.)	250

13.3.ค.) ตัวอย่างที่ 2 ใส่ฟิลเตอร์ให้กับเร้นจ์และกรองแบบ Filter by values

ตัวอย่างนี้ พัฒนาต่อจากตัวอย่างในข้อก่อนหน้า โดยเพิ่มกฎการกรองแบบ Filter by values ลงไป

```
// (1.) สร้างฟิลเตอร์ให้กับเร้นจ์ - ข้อมูลที่ติดกับ A1 ทั้งหมด
var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet() ;
sheet.getRange('A1').getDataRegion().createFilter() ;

// (2.) สร้างกฎการกรอง
var criteria = SpreadsheetApp.newFilterCriteria()
    .setHiddenValues(['DGG','DCC'])
    .build() ;

// (3.) จับฟิลเตอร์ในชีท จากนั้นกำหนดกฎการกรองใหม่ลงไป
sheet.getFilter().setColumnFilterCriteria(2, criteria) ;
```

ผล - สังเกตคอลัมน์ที่ 2 กรองโดยซ่อน DGG, DCC ไว้ และสังเกตที่แถว 4-27 ถูกซ่อนไว้

	A	B	C	D
1	รหัส = หมวด ▾		สินค้าหรือบริการ =	ราคาต่อหน่วย =
2	OTC0020	OTC	Test 2	100
3	OTC0019	OTC	Test 1	50
28				

13.4. การแก้ไขกฎการกรอง

การแก้ไขกฎการกรอง หรือ การเปลี่ยนเงื่อนไขการกรอง ให้จับฟิลเตอร์ที่มีอยู่แล้วไว้ก่อน จากนั้นใช้ setColumnFilterCriteria() เพื่อใส่กฎการกรองตัวใหม่ให้กับเร้นจ์

13.4.ก.) ตัวอย่างที่ 1

```
var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet() ;

// เร้นจ์ที่จับ มีฟิลเตอร์อยู่แล้ว
var range = sheet.getRange('A1').getDataRegion() ;

// สร้างกฎการกรอง
var criteria = SpreadsheetApp.newFilterCriteria()
    .setHiddenValues(['DGG','DCC'])
    .build() ;

// จับไปที่ฟิลเตอร์ของเร้นจ์ก่อน จากนั้นจึงกำหนดกฎการกรองลงไป
range.getFilter().setColumnFilterCriteria(2, criteria) ;
```

13.4.ข.) ตัวอย่างที่ 2 ฟิลเตอร์โดยใช้ค่าในเซลล์(Dropdown)

```
function FilterSheet() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();

  // เรนจ์ที่มีข้อมูล
  var range = sheet.getRange('A1').getDataRegion() ;

  // Filter value - จับค่าที่ใช้เป็นเงื่อนไขการกรอง
  var valueCondition = sheet.getRange('F1').getValue() ;    // จับ 1 ค่า - eg. OTC

  // ค่าที่ใช้เป็นเงื่อนไขการกรองทั้งหมด
  var valuesAllCondition = [ 'DCC' , 'DGG' , 'OTC' ] ;

  var valuesFilterOut = [] ;
  valuesAllCondition.forEach(function(item){
    if(item != valueCondition){
      valuesFilterOut.push(item) ;
    }
  }) ; // Close - forEach

  Logger.log(valuesFilterOut) ; // [ DCC , DGG ] - เป็นอาร์เรย์เก็บค่าที่จะใช้ซ่อนเมื่อกรอง

  var filter = sheet.getFilter() ;
  var criteria = SpreadsheetApp.newFilterCriteria()
    .setHiddenValues(valuesFilterOut)
    .build() ;

  if(filter != null){ // ถ้ามี Filter แล้ว
    range.getFilter().setColumnFilterCriteria(2, criteria) ;
  } else if(filter == null){ // ถ้าไม่มี Filter
    range.createFilter().setColumnFilterCriteria(2, criteria) ;
  } // Close - if
}
```

ผล - การทดสอบ ที่เซลล์ F1 เลือกหรือกรอกค่าที่จะใช้เป็นเงื่อนไขการกรอง จากนั้นจึงรันสคริปต์

	A	B	C	D	E	F
1	รหัส = หมวด ▾	สินค้าหรือบริการ	ราคาต่อหน่วย =			DGG ▾
4	DGG1520	DGG	อาบน้ำและตัดขนหมา (15-20 กก.)	550		
5	DGG0715	DGG	อาบน้ำและตัดขนหมา (7-15 กก.)	350		
6	DGG0017	DGG	อาบน้ำและตัดขนหมา (1- 7 กก.)	250		
12						

	A	B	C	D	E	F
1	รหัส = หมวด ▾	สินค้าหรือบริการ	ราคาต่อหน่วย =			OTC ▾
2	OTC0020	OTC	Test 2	100		
3	OTC0019	OTC	Test 1	50		
12						
13						

หมายเหตุ : ใช้การกรองแบบ Filter by condition เช่น whenTextContains ง่ายกว่า

13.5. การถอดกฎการกรองหรือถอดฟิลเตอร์

13.5.ก.) remove() - ถอดกฎการกรอง

remove() - เมธอดในคลาส Filter

<https://developers.google.com/apps-script/reference/spreadsheet/filter#remove>

ใช้ถอด Filter ออกจากเรนจ์หรือชีท (1 ชีท มีได้ 1 ฟิลเตอร์)

ตัวอย่าง

```
var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet() ;  
sheet.getFilter().remove() ;
```

13.5.ข.) removeColumnFilterCriteria() - ถอดคอลัมน์ที่มีการกรอง

removeColumnFilterCriteria(columnPosition)

<https://developers.google.com/apps-script/reference/spreadsheet/filter#removecolumnfiltercriteriacolumnposition>

ใช้ถอดกฎการกรองออกจากคอลัมน์ที่ระบุ แต่ไม่ได้ถอดฟิลเตอร์ออกจากเรนจ์

ตัวอย่าง - <https://caymezon.com/gas-filter/>

```
function Sample() {  
    var sheet = SpreadsheetApp.getActiveSheet();  
    var filter = sheet.getFilter();  
    var criteria = filter.getColumnFilterCriteria(1);  
    if(criteria != null){  
        filter.removeColumnFilterCriteria(1)  
    }  
}
```

13.6. การเรียงข้อมูลผลการกรอง

sort(columnPosition, ascending) - เมธอดในคลาส Filter

<https://developers.google.com/apps-script/reference/spreadsheet/filter#sortcolumnposition,-ascending>

ใช้เรียงเรนจ์ที่กรองแล้วตามคอลัมน์ที่กำหนด การเรียงนี้ไม่รวมแถวแรก(หัวตาราง) ของเรนจ์ที่ใส่ฟิลเตอร์ โดย sort() คืนค่ากลับมาเป็นคลาส Filter

พารามิเตอร์

Name	Type	Description
columnPosition	Integer	The 1-indexed position of the column.
ascending	Boolean	If true, sort the filtered range in ascending order; if false, sort the filtered range in descending order.

13.7. withCriteria()

13.7.ก.) เมธอดที่เกี่ยวข้อง

getCriteriaType() – เมธอดในคลาส FilterCriteriaBuilder

<https://developers.google.com/apps-script/reference/spreadsheet/filter-criteria-builder#getcriteriatype>

คืนค่าเป็นชนิดของกฎการกรอง ตามที่ระบุใน Enum **BooleanCriteria** การจับค่าของกฎสามารถทำได้

โดยใช้ Method **getCriteriaValues()** การใช้ค่าเหล่านี้เพื่อสร้างหรือแก้ไขกฎของฟิลเตอร์ ให้ดูเพิ่มเติม

Method **withCriteria(criteria, args)**

Enum BooleanCriteria – กฎการกรองโดยใช้เงื่อนไข Filter by condition

<https://developers.google.com/apps-script/reference/spreadsheet/boolean-criteria>

Property	Type	Description
CELL_EMPTY	Enum	The criteria is met when a cell is empty.
CELL_NOT_EMPTY	Enum	The criteria is met when a cell is not empty.
DATE_AFTER	Enum	The criteria is met when a date is after the given value.
DATE_BEFORE	Enum	The criteria is met when a date is before the given value.
DATE_EQUAL_TO	Enum	The criteria is met when a date is equal to the given value.
DATE_AFTER_RELATIVE	Enum	The criteria is met when a date is after the relative date value.
DATE_BEFORE_RELATIVE	Enum	The criteria is met when a date is before the relative date value.
DATE_EQUAL_TO_RELATIVE	Enum	The criteria is met when a date is equal to the relative date value.
NUMBER_BETWEEN	Enum	The criteria is met when a number that is between the given values.
NUMBER_EQUAL_TO	Enum	The criteria is met when a number that is equal to the given value.
NUMBER_GREATER_THAN	Enum	The criteria is met when a number that is greater than the given value.
NUMBER_GREATER_THAN_OR_EQUAL_TO	Enum	The criteria is met when a number that is greater than or equal to the given value.
NUMBER_LESS_THAN	Enum	The criteria is met when a number that is less than the given value.
NUMBER_LESS_THAN_OR_EQUAL_TO	Enum	The criteria is met when a number that is less than or equal to the given value.
NUMBER_NOT_BETWEEN	Enum	The criteria is met when a number that is not between the given values.
NUMBER_NOT_EQUAL_TO	Enum	The criteria is met when a number that is not equal to the given value.
TEXT_CONTAINS	Enum	The criteria is met when the input contains the given value.
TEXT_DOES_NOT_CONTAIN	Enum	The criteria is met when the input does not contain the given value.
TEXT_EQUAL_TO	Enum	The criteria is met when the input is equal to the given value.
TEXT_STARTS_WITH	Enum	The criteria is met when the input begins with the given value.
TEXT_ENDS_WITH	Enum	The criteria is met when the input ends with the given value.
CUSTOM_FORMULA	Enum	The criteria is met when the input makes the given formula evaluate to true.

getCriteriaValues() – เมธอดในคลาส FilterCriteriaBuilder

<https://developers.google.com/apps-script/reference/spreadsheet/filter-criteria-builder#getcriteriavalues>

คืนค่าเป็นอาร์เรย์ของอาร์กิวเมนต์สำหรับ Boolean criteria หรือ **Object[]**

withCriteria(criteria, args) – เมธอดในคลาส FilterCriteriaBuilder

<https://developers.google.com/apps-script/reference/spreadsheet/filter-criteria-builder#withcriteriacriteria,-args>

ใช้กำหนดกฎของการกรอง ที่นำมาจากกฎการกรองที่มีอยู่แล้ว โดย **criteria** และ **aruments** ได้มาจาก

getCriteriaType() และ **getCriteriaValues()** ตามลำดับ

ใช้ตรงๆแบบนี้ **withCriteria(SpreadsheetApp.BooleanCriteria.TEXT_CONTAINS,'10')** ใช้ไม่ได้

Name	Type	Description
<code>criteria</code>	BooleanCriteria	The type of boolean criteria.
<code>args</code>	Object[]	An array of arguments appropriate to the criteria type; the number of arguments and their type match the corresponding when. method above.

ตัวอย่าง – ดูปฏิกิริยาการกรองจากคอลัมน์มาใช้ แล้วปรับแต่งเล็กน้อย

```
// สร้างกฎการกรองจากกฎที่มีอยู่แล้ว + เป็นการกำหนดรูปแบบมีเงื่อนไขด้วย
// Builds a filter criteria that is based on an existing criteria
var sheet = SpreadsheetApp.getActiveSheet() ;
var criteria = sheet.getFilter().getColumnFilterCriteria(1) ;
var newCriteria = SpreadsheetApp.newFilterCriteria()
    .withCriteria(criteria.getCriteriaType(), criteria.getCriteriaValues())
    .setHiddenValues(['c'])
    .build() ;
```

13.8. กฎการกรองแบบ Filter by condition

คลาส FilterCriteriaBuilder

<https://developers.google.com/apps-script/reference/spreadsheet/filter-criteria-builder?hl=ja>

Method ตระกูล when ใช้กำหนดกฎการกรองแบบ Filter by condition เช่น whenCellEmpty(), whenDateAfter(date), whenFormulaSatisfied(), whenNumberBetween(), whenNumberEqualTo(), whenNumberGreaterThan(), whenTextContains() เป็นต้น

13.8.ก.) ตัวอย่างที่ 1 – whenTextContains()

<https://caymezon.com/gas-filter/>

```
function whenTextContainsSample() {
    var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet() ;
    var range = sheet.getRange('A1').getDataRegion() ;
    var filter = range.getFilter() ;
    var criteria = SpreadsheetApp.newFilterCriteria()
        .whenTextContains("Test") // มีคำว่า Test
        .build() ;

    if(filter != null){
        range.getFilter().setColumnFilterCriteria(3, criteria) ;
    }else{
        range.createFilter().setColumnFilterCriteria(3, criteria) ;
    }
}
```

ผล

	A	B	C	D
1	รหัส	หมวด	สินค้าหรือบริการ	ราคาต่อหน่วย
2	OTC0020	OTC	Test 0020	100
3	OTC0019	OTC	Test 0019	50

13.8.ข.) ตัวอย่างที่ 2 – whenDateEqualTo()

<https://caymezon.com/gas-filter/>

```
function whenTextContainsSample() {  
  
    var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet() ;  
    var range = sheet.getRange('A1').getDataRegion() ;  
    var filter = range.getFilter() ;  
    var criteria = SpreadsheetApp.newFilterCriteria()  
                    .whenDateEqualTo(SpreadsheetApp.RelativeDate.TODAY)  
                    .build() ; // เป็นวันนี้  
  
    if(filter != null){  
        range.getFilter().setColumnFilterCriteria(1, criteria) ;  
    }else{  
        range.createFilter().setColumnFilterCriteria(1, criteria) ;  
    }  
}
```

ผล - กรองเฉพาะค่าที่เท่ากับวันนี้

	A	B
1	Date	Count
3	14 Mar 2020	9
7	14 Mar 2020	6
9	14 Mar 2020	9
14		

บทที่ 14

การสร้าง Chart



14.1. สร้าง Bar Chart

ดูตัวอย่างการสร้างชาร์ตกันก่อน แล้วค่อยไปดูอธิบายเมทอดและคุณสมบัติของวัตถุที่ใช้

14.1.ก.) ตัวอย่างการสร้าง Bar Chart - คอลัมน์ Cost

ที่มา : คอร์ส Apps Script Blastoff (เรียนฟรี)

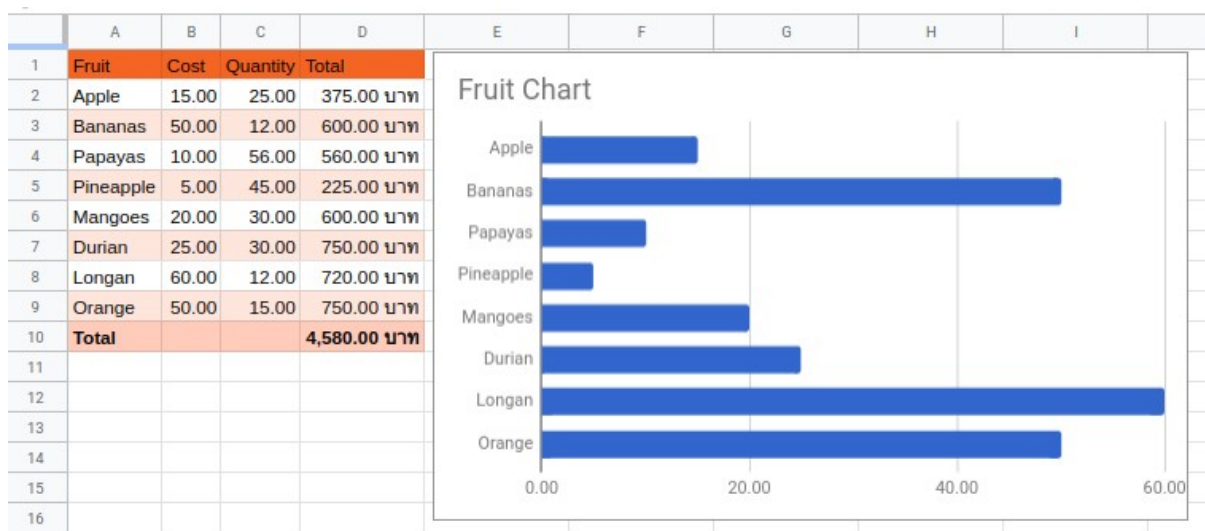
<https://courses.benlcollins.com/courses/enrolled/435404>

<https://courses.benlcollins.com/courses/435404/lectures/6708194>

โค้ดต่อไปนี้ สร้าง Bar chart จากข้อมูลในชีท โดยใช้ข้อมูลเฉพาะในคอลัมน์ **A(Fruit)** และ **B (Cost)**

```
function addChart_1() {  
  var ss = SpreadsheetApp.getActive();  
  var sheet1 = ss.getSheetByName('Sheet1');  
  
  // จับข้อมูลในเรนจ์ Sheet1!A1:B9 - 2 คอลัมน์แรก  
  var dataRange = sheet1.getRange("A1:B9"); // เรนจ์ที่จะใช้สร้างชาร์ต  
  
  var myNewChart = sheet1.newChart(); // สร้างวัตถุชาร์ตว่างๆ  
  
  myNewChart.addRange(dataRange) // เรนจ์ที่จะใช้สร้างชาร์ต  
    .setChartType(Charts.ChartType.BAR) // ชนิดของชาร์ตที่จะสร้าง  
    .setPosition(1,5,0,0) // ตำแหน่งวางชาร์ต แถวที่ 1 คอลัมน์ที่ 5  
    .setNumHeaders(1) // มีหัวตาราง 1 แถว  
    .setOption("title","Fruit Chart"); // กำหนด Title เป็น Fruit Chart  
  
  sheet1.insertChart(myNewChart.build()); // แทรกชาร์ตลงในชีท  
}
```

ผล



14.1.ข.) ตัวอย่างการสร้าง Bar Chart- คอลัมน์ Total

โค้ดต่อไปนี้จะสร้าง Bar chart จากข้อมูลในชีต พิเศษกว่าข้อก่อนหน้า ตรงที่มีเร้นจ์ 2 เร้นจ์ป้อนให้กับตัวสร้างชาร์ต ซึ่งต้องใช้เมธอด `setMergeStrategy` ในการรวมเร้นจ์ด้วย

```
function addChart(){
  var ss = SpreadsheetApp.getActive() ;
  var sheet1 = ss.getSheetByName('Sheet1') ;
  var lastRow = sheet1.getLastRow() ;
  var lastColumn = sheet1.getLastColumn() ;

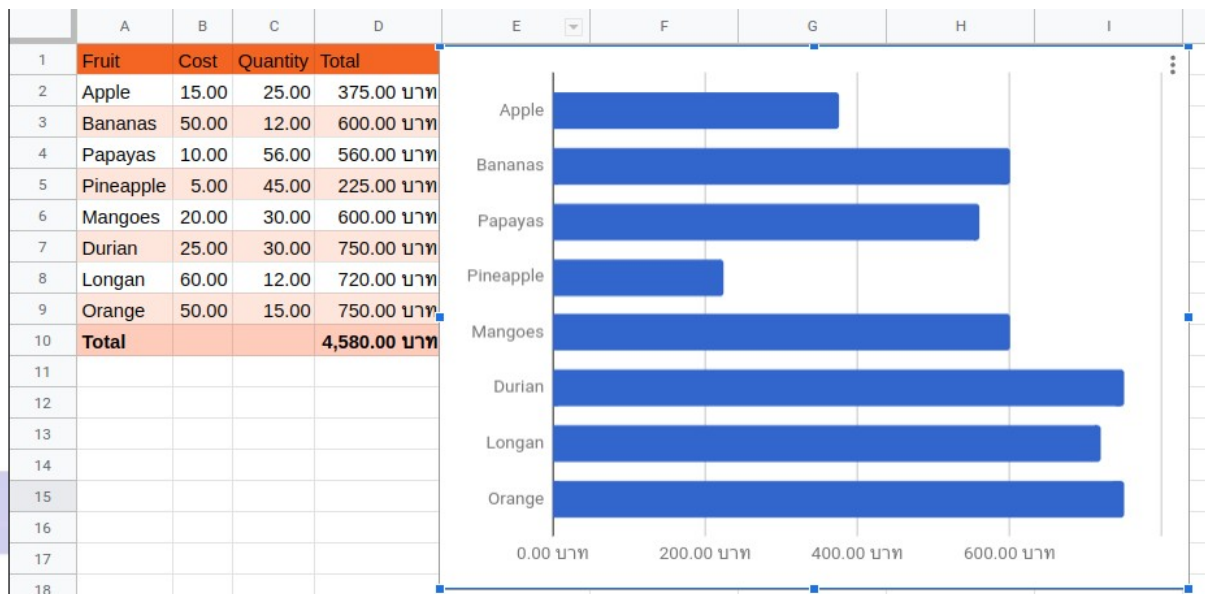
  var totalChartLabels = sheet1.getRange(1, 1, lastRow-1, 1) ;
  // คอลัมน์ที่ 1 (A) - คอลัมน์ Fruit

  var totalChartValues = sheet1.getRange(1, lastColumn, lastRow-1, 1) ;
  // คอลัมน์สุดท้าย (D) - คอลัมน์ Total

  var totalChart = sheet1.newChart()
    .setChartType(Charts.ChartType.BAR)
    .addRange(totalChartLabels) // เร้นจ์ที่ 1 - Label
    .addRange(totalChartValues) // เร้นจ์ที่ 2 - Values
    .setMergeStrategy(Charts.ChartMergeStrategy.MERGE_COLUMNS)
    .setPosition(1,6,0,0)
    .setNumHeaders(1)
    .build() ;

  sheet1.insertChart(totalChart) ;
}
```

ผล



หมายเหตุ : ดูการสร้าง Chart แบบ Line ได้ที่ลิงค์ต่อไปนี้

Add a line chart - สร้าง Chart แบบเส้น

<https://developers.google.com/google-ads/scripts/docs/examples/spreadsheetapp#add-a-line-chart>

14.2. newChart() และ build()

NewChart() - เมธอดในคลาส Sheet

<https://developers.google.com/apps-script/reference/spreadsheet/sheet.html#newchart>

คืนค่ากลับมาเป็นตัวสร้างชาร์ต (Chart builder) หรือ คลาส **EmbeddedChartBuilder** แต่ชาร์ตจะยังไม่ถูกสร้างจนกว่าจะใช้ Method **build()**

คลาส **EmbeddedChartBuilder**

<https://developers.google.com/apps-script/reference/spreadsheet/embedded-chart-builder.html>

build() - เมธอดในคลาส **EmbeddedChartBuilder**

<https://developers.google.com/apps-script/reference/spreadsheet/embedded-chart-builder.html#build>

ใช้สร้างชาร์ต จากตัวสร้างชาร์ตที่กำหนดค่าต่างๆไว้

```
var chartBuilder = sheet.newChart();  
Logger.log(chartBuilder);           // พิมพ์ : EmbeddedChartBuilder
```

14.3. setChartType()

setChartType(type) - เมธอดในคลาส **EmbeddedChartBuilder**

<https://developers.google.com/apps-script/reference/spreadsheet/embedded-chart-builder.html#setcharttypetype>

ใช้เปลี่ยนชนิดของชาร์ต โดยจะคืนค่ากลับมาเป็นคลาส **EmbeddedChartBuilder**

พารามิเตอร์ **type** คุ้ได้จากตาราง Enum **ChartType** ดังต่อไปนี้ การเลือกชนิดของชาร์ต เช่น

Chart.ChartType.(Property)

Enum **ChartType**

<https://developers.google.com/apps-script/reference/charts/chart-type.html>

Property	Type	Description
TIMELINE	Enum	Timeline chart
AREA	Enum	Area chart
BAR	Enum	Bar chart
BUBBLE	Enum	Bubble chart
CANDLESTICK	Enum	Candlestick chart
COLUMN	Enum	Column chart
COMBO	Enum	Combo chart
GAUGE	Enum	Gauge chart
GEO	Enum	Geo chart
HISTOGRAM	Enum	Histogram
RADAR	Enum	Radar chart
LINE	Enum	Line chart
ORG	Enum	Org chart
PIE	Enum	Pie chart
SCATTER	Enum	Scatter chart
SPARKLINE	Enum	Sparkline chart
STEPPED_AREA	Enum	Stepped area chart
TABLE	Enum	Table chart
TREEMAP	Enum	Treemap chart
WATERFALL	Enum	Waterfall chart

14.4. addRange()

addRange(range) - เมธอดในคลาส EmbeddedChartBuilder

<https://developers.google.com/apps-script/reference/spreadsheet/embedded-chart-builder.html#addrangerange>

เพิ่มเรนจ์ให้กับตัวสร้างชาร์ต โดยจะคืนค่ากลับมาเป็นคลาส EmbeddedChartBuilder

การใช้งานเช่น

```
var chart = sheet.newChart()
    .setChartType(Charts.ChartType.BAR)
    .addRange(range)
```

14.5. setNumHeaders()

setNumHeaders(headers) - เมธอดในคลาส EmbeddedChartBuilder

<https://developers.google.com/apps-script/reference/spreadsheet/embedded-chart-builder.html#setnumheadersheaders>

กำหนดจำนวนแถวหรือคอลัมน์ที่จะใช้เป็น headers ของข้อมูล โดยจะคืนค่ากลับมาเป็นคลาส

EmbeddedChartBuilder พารามิเตอร์ **headers** เป็นเลขจำนวนเต็ม

การใช้งานเช่น

```
var chart = sheet.newChart()
    .setChartType(Charts.ChartType.BAR)
    .addRange(range)
    .setNumHeaders(1)
```

14.6. setPosition()

setPosition(anchorRowPos, anchorColPos, offsetX, offsetY) - เมธอดในคลาส EmbeddedChartBuilder

<https://developers.google.com/apps-script/reference/spreadsheet/embedded-chart-builder.html#setpositionanchorrowpos,-anchorcolpos,-offsetx,-offsety>

ใช้กำหนดตำแหน่งของชาร์ต โดยอ้างอิงจากตำแหน่งของเซลล์ในชีท

พารามิเตอร์

Name	Type	Description
anchorRowPos	Integer	The chart's top side is anchored in this row.
anchorColPos	Integer	The chart's left side is anchored in this column.
offsetX	Integer	The chart's upper right-hand corner is offset by this many pixels.
offsetY	Integer	The chart's lower left-hand corner is offset by this many pixels.

การใช้งานเช่น

```
var chart = sheet.newChart()
    .setChartType(Charts.ChartType.BAR)
    .addRange(range)
    .setNumHeaders(1)
    .setPosition(5, 5, 0, 0)
```

14.7. setMergeStrategy()

setMergeStrategy(mergeStrategy) - เมธอดในคลาส EmbeddedChartBuilder

<https://developers.google.com/apps-script/reference/spreadsheet/embedded-chart-builder.html#setmergestrategymergestrategy>

ใช้ในกรณีเพิ่มเร้นจ์มากกว่า 1 เร้นจ์ ใช้กำหนดการรวมเร้นจ์ว่าจะรวมในแนวคอลัมน์ หรือแนวแถว

พารามิเตอร์ **mergeStrategy** ก็คือ Enum ChartMergeStrategy ดูได้จากตารางต่อไปนี้ ก็กำหนดคุณสมบัติเช่น Charts.ChartMergeStrategy.(Property)

Enum ChartMergeStrategy

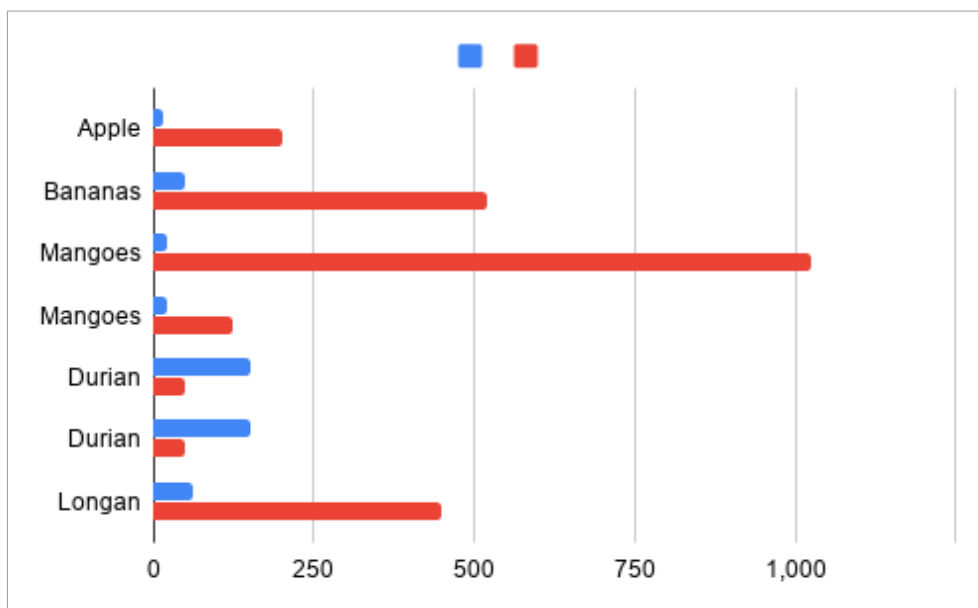
<https://developers.google.com/apps-script/reference/charts/chart-merge-strategy.html>

Property	Type	Description
MERGE_COLUMNS	Enum	Default. Charts merges the columns of multiple ranges.
MERGE_ROWS	Enum	Charts merges the rows of multiple ranges.

การใช้งานเช่น

```
var range = sheet.getRange("A2:B10");
var range2 = sheet.getRange("C2:C10");
var chart = sheet.newChart()
    .setChartType(Charts.ChartType.BAR)
    .addRange(range)
    .addRange(range2)
    .setMergeStrategy(Charts.ChartMergeStrategy.MERGE_COLUMNS)
    .setPosition(5, 5, 0, 0)
    .build() ;
```

ตัวอย่างผลลัพธ์



14.8. insertChart(), updateChart() และ getChart()

insertChart(chart) - เมธอดในคลาส Sheet

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#insertchartchart>

ใส่ชาร์ตใหม่ลงในชีท พารามิเตอร์ **chart** ก็คือ คลาส **EmbeddedChart**

คลาส EmbeddedChart

<https://developers.google.com/apps-script/reference/spreadsheet/embedded-chart.html>

เป็นวัตถุชาร์ตที่อยู่ในไฟล์ Google Sheets

getCharts() - เมธอดในคลาส Sheet

<https://developers.google.com/apps-script/reference/spreadsheet/sheet.html#getcharts>

ใช้จับชาร์ตทั้งหมดที่มีอยู่ในชีท โดยจะคืนค่ากลับมาเป็นคลาส **EmbeddedChart[]** หรือชาร์ตที่เก็บอยู่ใน
อาเรย์

modify() - เมธอดในคลาส EmbeddedChart

<https://developers.google.com/apps-script/reference/spreadsheet/embedded-chart.html#modify>

คืนค่ากลับมาเป็นวัตถุ **EmbeddedChartBuilder** ที่สามารถแก้ไข เมื่อแก้ไขแล้ว ให้อัปเดตการแก้ไขโดย

ใช้เมธอด **updateChart()**

updateChart(chart) - เมธอดในคลาส Sheet

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#updatechartchart>

ใช้อัปเดตชาร์ตในชีท

14.8.ก.) ตัวอย่างที่ 1

โค้ดต่อไปนี้เป็น อัปเดตชาร์ตตัวที่ 1 เช่น ใส่ Title เป็น Update! ย้ายไปที่ตำแหน่ง 2,2 เป็นต้น

```
function updateChartNo1() {  
    var sheet = SpreadsheetApp.getActiveSheet();  
    var range = sheet.getRange("A2:B8");  
    var chart = sheet.getCharts()[0];  
  
    chart = chart.modify()  
        .addRange(range)  
        .setOption('title', 'Updated!')  
        .setOption('animation.duration', 500)  
        .setPosition(2,2,0,0)  
        .build();  
  
    sheet.updateChart(chart);  
}
```

14.8.ข.) ตัวอย่างที่ 2

โค้ดต่อไปนี้จะอัปเดตชาร์ตทั้งหมด แล้วเปลี่ยนให้เป็นแบบ Column

```
function updateNewChart() {  
  
  var ss = SpreadsheetApp.getActiveSpreadsheet();  
  var sheet = ss.getSheets()[0];  
  var charts = sheet.getCharts(); // จัปชาร์ตทั้งหมดในชีท  
  
  for (var i in charts) {  
    var chart = charts[i];  
    var newChart = chart.modify() // โหมดไฟยชาร์ตเดิม แล้วเก็บไว้ในตัวแปรใหม่  
      .setChartType(Charts.ChartType.COLUMN)  
      .build();  
  
    sheet.updateChart(newChart);  
  } // Close - for  
} // Close - function
```

14.9. จัปชาร์ตเป็นภาพ

Export Google sheet Chart as an image

<https://stackoverflow.com/questions/28387355/export-google-sheet-chart-as-an-image>

14.9.ก.) ส่งเมลล์

ตัวอย่าง

```
function createPieChart(){  
  
  var sheet = SpreadsheetApp.openById("<ID>").getSheetByName("<SheetName>") ;  
  var chartBuilder = sheet.newChart()  
    .asPieChart()  
    .set3D()  
    .addRange(sheet.getRange("A20").getDataRegion())  
    .setPosition(16, 5, 0, 0)  
    .setOption('title', "Total hour split in 2020") ;  
  
  var blob = chartBuilder.build().getBlob() ;  
  sendMail(blob) ;  
}  
  
function sendMail(img){  
  
  MailApp.sendEmail({  
    to: "john@domain.com" ,  
    subject: "test" ,  
    htmlBody: "fair enough" ,  
    attachments: [img]}) ;  
    // If this gives you problems, replace img with img.getAs(MimeType.JPEG)  
  }  
}
```

14.9.ข.) ใส่ไว้ในชีตอื่น

ตัวอย่าง

```
function createPieChart(){  
  
    var srcSheet=SpreadsheetApp.openById("<ID>").getSheetByName("<SheetName>") ;  
    var destSheet=SpreadsheetApp.openById("<ID>").getSheetByName("<SheetName>") ;  
  
    var chartBuilder = srcSheet.newChart()  
                                .asPieChart()  
                                .set3D()  
                                .addRange(sheet.getRange("A20").getDataRegion())  
                                .setPosition(16, 5, 0, 0)  
                                .setOption('title', "Total hour split in 2020") ;  
  
    var blob = chartBuilder.build().getBlob() ;  
  
    destSheet.insertImage(blob, 1, 1) ;  
}
```


บทที่ 15
Drawing
และ
Over Grid Image



15.1. Drawing(รูปภาพ)

`getDrawings()` - เมธอดในคลาส Sheet

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#getdrawings>

`getHeight()` - เมธอดในคลาส Drawing

<https://developers.google.com/apps-script/reference/spreadsheet/drawing#getheight>

`getOnAction()` - เมธอดในคลาส Drawing

<https://developers.google.com/apps-script/reference/spreadsheet/drawing#getonaction>

`remove()` - เมธอดในคลาส Drawing

<https://developers.google.com/apps-script/reference/spreadsheet/drawing#remove>

ตัวอย่าง - วนลูปจับ Drawing แล้วจับความสูงออกมาดู

```
// Logs the height of all drawings in a sheet
var drawings = SpreadsheetApp.getActiveSheet().getDrawings();
for (var i = 0 ; i < drawings.length ; i++) {
  Logger.log(drawings[i].getHeight());
}
```

ตัวอย่าง - วนลูปจับ Drawing แล้วจับชื่อของฟังก์ชัน Apps Script ที่ Assign ให้กับรูปภาพออกมาดู

```
// Logs the macro name of all drawings on the active sheet.
var drawings = SpreadsheetApp.getActiveSheet().getDrawings();
for (var i = 0 ; i < drawings.length ; i++) {
  Logger.log(drawings[i].getOnAction());
}
```

ตัวอย่าง

```
// Deletes all drawings from the active sheet.
var drawings = SpreadsheetApp.getActiveSheet().getDrawings();
for (var i = 0 ; i < drawings.length ; i++) {
  drawings[i].remove();
}
```

15.2. Over Grid Images

ภาพมี 2 แบบ ก็คือ แบบ **ฝังในเซลล์**(In Cell Images) และ แบบ **ลอยเหนือเส้นกริด**(Over Grid Images) ตัวอย่างตามภาพ

	A	B	C	D
1				
2				
3				
4				
5				
6				
7				

`getImages()` - เมธอดในคลาส Sheet

<https://developers.google.com/apps-script/reference/spreadsheet/sheet#getimages>

`getImages()` - เมธอดในคลาส Spreadsheet

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet#getimages>

`getHeight()` - เมธอดในคลาส OverGridImage

<https://developers.google.com/apps-script/reference/spreadsheet/over-grid-image#getheight>

ตัวอย่าง - - วนลูปจับ Over Grid Images แล้วจับความสูงออกมาดู

```
// Logs the height of all images in a spreadsheet
var images = SpreadsheetApp.getActiveSpreadsheet().getImages() ;
for (var i = 0 ; i < images.length ; i++) {
  Logger.log(images[i].getHeight()) ;
}
```

ตัวอย่าง - ลบทุกภาพใน Spreadsheet

```
// Deletes all images in a spreadsheet
var images = SpreadsheetApp.getActiveSpreadsheet().getImages() ;
for (var i = 0 ; i < images.length ; i++) {
  images[i].remove() ;
}
```


บทที่ 16

User Interfaces



16.1. การสร้างเมนู

16.1.ก.) คลาสและเมธอดที่สำคัญในการสร้างเมนู

คลาส Ui

<https://developers.google.com/apps-script/reference/base/ui>

Custom Menus in G Suite

<https://developers.google.com/apps-script/guides/menus>

createMenu(caption) - เมธอดในคลาส Ui

<https://developers.google.com/apps-script/reference/base/ui#createmenucaption>

getUi() - เมธอดในคลาส SpreadsheetApp

<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet-app#getui>

ใช้จับ Ui ของ Google Sheets คืนค่ากลับมาเป็น UI (user-interface) หรือสภาพแวดล้อมของ

โปรแกรมที่ผู้ใช้งานกำลังทำงานด้วย ซึ่งเราสามารถเพิ่มเมนู, กรอบโต้ตอบ(Dialogs) หรือ กรอบด้านข้าง(Sidebars) ได้

createMenu(caption) - เมธอดในคลาส Ui

<https://developers.google.com/apps-script/reference/base/ui#createmenucaption>

ใช้สร้างตัวสร้างเมนู โดยจะคืนค่ากลับมาเป็นคลาส **Menu** จากนั้นก็จะสามารถใส่เมนูต่างๆลงไปได้ แต่ว่าเมนูจะยังไม่ถูกเพิ่มเข้าไปใน Ui ของ Google Sheets จนกว่า ใช้ Method ชื่อ **addToUi()**

พารามิเตอร์ **caption** ก็คือ ชื่อของเมนูในเมนูบาร์ที่จะสร้าง

createAddonMenu() - เมธอดในคลาส Ui

<https://developers.google.com/apps-script/reference/base/ui#createaddonmenu>

คล้ายกับ Method **createMenu()** ใช้สร้างเมนูเหมือนกัน แต่จะสร้างเป็นเมนูย่อย ของเมนู Addon โดยจะคืนค่ากลับมาเป็นคลาส **Menu**

addItem(caption, functionName)

<https://developers.google.com/apps-script/reference/base/menu.html#additemcaption,-functionname>

เพิ่มรายการ(item หรือ เมนูย่อย) ลงในเมนู โดยจะคืนค่ากลับมาเป็นคลาส **Menu** (**caption** = ชื่อเมนูย่อย **functionName** = ชื่อฟังก์ชัน)

ตัวอย่าง

Dialogs and Sidebars in G Suite Documents

<https://developers.google.com/apps-script/guides/dialogs>

```
function onOpen(e) { // e คืออะไรดูเรื่อง Simple trigger
  SpreadsheetApp.getUi()
    .createMenu('Custom Menu')
    // ชื่อเมนู ชื่อฟังก์ชัน
    .addItem('Show alert', 'showAlert')
    .addToUi() ;

  Logger.log(e); // ดูต่อ ที่ผลการ Logs ----- > [ 01 ]
}

function showAlert() {
  var ui = SpreadsheetApp.getUi();
  var result = ui.alert( 'Please confirm' ,
    'Are you sure you want to continue?' ,
    ui.ButtonSet.YES_NO );

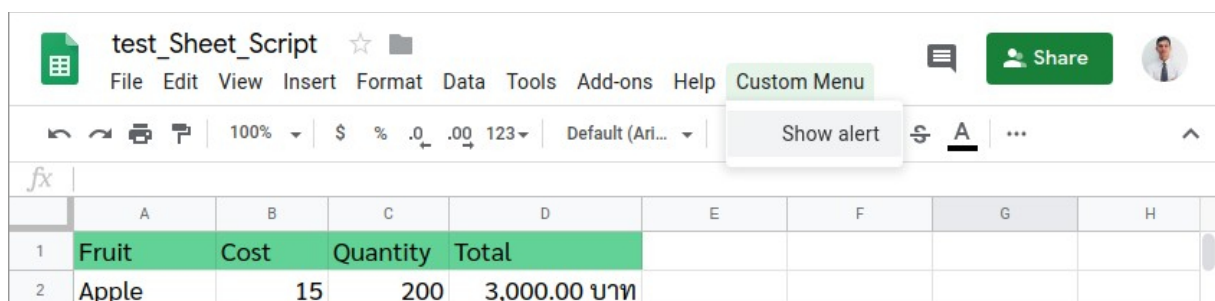
  // ถ้ามีผู้สเชอร์ตอบสนอง
  if (result == ui.Button.YES) {
    ui.alert('Confirmation received.');// User clicked "Yes".
  } else {
    ui.alert('Permission denied.');// User clicked "No" or X in the title bar.
  }
}
```

เมื่อปิดเปิดไฟล์ Google Sheet โค้ดจะรันอัตโนมัติ

Logs

```
[ 01 ] { authMode=LIMITED,
  range=Range,
  source=Spreadsheet,
  user=wasankds@gmail.com }
```

ผล - มีเมนู Custom Menu ปรากฏเพิ่มเติมเมื่อเปิดไฟล์ Google Sheet



16.2. alert()

`alert(title, prompt, buttons)` - เมธอดในคลาส Ui

[https://developers.google.com/apps-script/reference/base/ui#alert\(title,-prompt,-buttons\)](https://developers.google.com/apps-script/reference/base/ui#alert(title,-prompt,-buttons))

Alert dialogs

https://developers.google.com/apps-script/guides/dialogs#alert_dialogs

`alert()` ใช้เปิดกรอบ Dialogue ที่มีข้อความ หัวเรื่อง และ ปุ่มต่างๆ ที่เรากำหนดได้ เมื่อ Dialogue ถูกเปิดสรีปต์จะหยุดการรัน และจะทำงานต่อเมื่อผู้ใช้งานตอบสนองต่อ Dialogue

พารามิเตอร์

Name	Type	Description
<code>title</code>	String	The <code>title</code> to display above the dialog box.
<code>prompt</code>	String	The <code>message</code> to display in the dialog box.
<code>buttons</code>	ButtonSet	The <code>button set</code> to display in the dialog box. (ดูตาราง Enum ButtonSet)

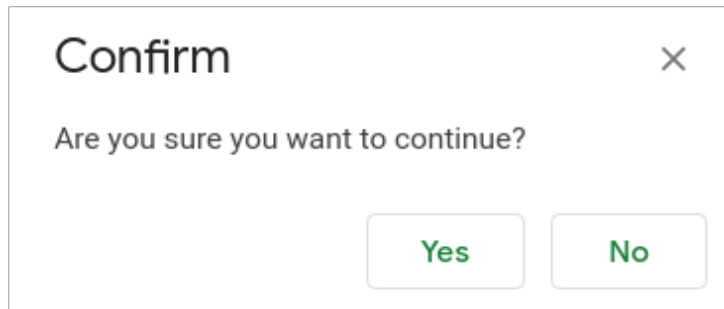
Enum ButtonSet

<https://developers.google.com/apps-script/reference/base/button-set.html>

Property	Type	Description
<code>OK</code>	Enum	A single <code>"OK"</code> button, indicating an informational message that can only be dismissed.
<code>OK_CANCEL</code>	Enum	An <code>"OK"</code> button and a <code>"Cancel"</code> button, allowing the user to either proceed with or halt an operation.
<code>YES_NO</code>	Enum	A <code>"Yes"</code> button and a <code>"No"</code> button, allowing the user to answer a yes/no question.
<code>YES_NO_CANCEL</code>	Enum	A <code>"Yes"</code> button, a <code>"No"</code> button, and a <code>"Cancel"</code> button, allowing the user to either answer a yes/no question or halt an operation.

ตัวอย่าง

```
function main() {  
  var ui = SpreadsheetApp.getUi() ;  
  var response = ui.alert( 'Confirm'  
    , 'Are you sure you want to continue?'  
    , ui.ButtonSet.YES_NO);  
  
  // Process the user's response.  
  if (response == ui.Button.YES) {  
    Logger.log('The user clicked "Yes."') ;  
  } else {  
    Logger.log('The user clicked "No" or the close button in the dialog\'s title bar.') ;  
  }  
}
```



หาก Log คุณจะผลลัพธ์ดังนี้ เมื่อผู้ใช้งานคลิกปุ่ม Yes

Logs

[] The user clicked "Yes."

16.3. prompt()

`prompt(title, prompt, buttons)` - เมธอดในคลาส Ui

<https://developers.google.com/apps-script/reference/base/ui#prompttitle,-prompt,-buttons>

Prompt dialogs

https://developers.google.com/apps-script/guides/dialogs#prompt_dialogs

`prompt()` ใช้เปิดกรอบ Input Dialogue ที่ผู้ใช้งานสามารถกรอกข้อความ(input) เพื่อจะนำไปประมวลผลต่อได้ มีปุ่ม มีหัวเรื่อง มีข้อความ เหมือนๆกับ Method `alert()`

พารามิเตอร์

Name	Type	Description
<code>title</code>	String	The title to display above the dialog box.
<code>prompt</code>	String	The message to display in the dialog box.
<code>buttons</code>	ButtonSet	The button set to display in the dialog box. (ดูตาราง Enum ButtonSet)

`prompt()` คืนค่ากลับมาเป็นคลาส `PromptResponse` ซึ่งเราสามารถใช้ เมธอดในคลาสนี้ จับคำตอบที่ผู้ตอบกลับมาประมวลผลต่อได้ เช่น `getResponseText()`, `getSelectedButton()`

คลาส `PromptResponse`

<https://developers.google.com/apps-script/reference/base/prompt-response.html>

`getResponseText()` - เมธอดในคลาส `PromptResponse`

<https://developers.google.com/apps-script/reference/base/prompt-response.html#getresponsetext>

ใช้จับข้อความที่กรอกลงใน Input box โดยจะคืนค่ากลับมาเป็นข้อความ(Straing)

`getSelectedButton()` - เมธอดในคลาส `PromptResponse`

<https://developers.google.com/apps-script/reference/base/prompt-response.html#getselectedbutton>

ใช้จับปุ่มที่คลิก โดยจะคืนค่ากลับมาเป็น Enum Button

Enum Button

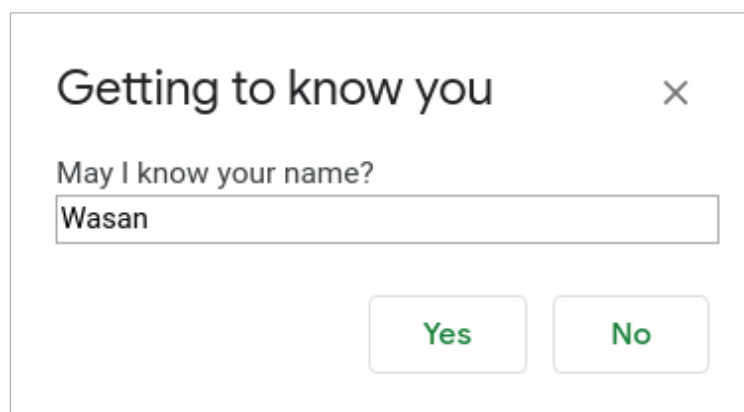
<https://developers.google.com/apps-script/reference/base/button.html>

Property	Type	Description
CLOSE	Enum	The standard close button displayed in every dialog's title bar. This button is not explicitly added to a dialog, and it cannot be removed.
OK	Enum	An "OK" button, indicating that an operation should proceed.
CANCEL	Enum	A "Cancel" button, indicating that an operation should not proceed.
YES	Enum	A "Yes" button, indicating a positive response to a question.
NO	Enum	A "No" button, indicating a negative response to a question.

ตัวอย่าง

```
function main() {  
  var ui = SpreadsheetApp.getUi();  
  var response = ui.prompt( 'Getting to know you'  
    , 'May I know your name?'  
    , ui.ButtonSet.YES_NO );  
  
  if (response.getSelectedButton() == ui.Button.YES) {           // คลิก Yes  
    Logger.log('The user\'s name is %s.', response.getResponseText());  
  } else if (response.getSelectedButton() == ui.Button.NO) {     // คลิก No  
    Logger.log('The user didn\'t want to provide a name.');  } else {                                                       // คลิกอย่างอื่น  
    Logger.log('The user clicked the close button in the dialog\'s title bar.');  }  
}
```

ผลหลังรันโค้ด



ถ้า Logs คุณจะได้ผลลัพธ์ดังนี้

```
Logs  
[ 01 ] The user's name is Wasan.
```

16.4. Custom dialogs (Modal Dialog)

Custom dialogs

https://developers.google.com/apps-script/guides/dialogs#custom_dialogs

Custom dialogs เป็น Dialog ที่เราออกแบบเอง โดยใช้เซอวิส HTML ในการสร้าง Dialog ในรูปแบบหน้าเว็บ HTML ฉะนั้นเวลารันสคริปต์ สคริปต์จะไม่หยุดรัน(ฝั่ง Server-side) เมื่อ Dialog ถูกเปิด

HTML Service: Communicate with Server Functions

<https://developers.google.com/apps-script/guides/html/communication>

คลาส google.script.run (Client-side API)

<https://developers.google.com/apps-script/guides/html/reference/run>

google.script.run เป็นการสื่อสารทางเดียวในฝั่ง Client-side เป็น JavaScript API และเป็นสิ่งที่มีในเซอวิส HTML ที่สามารถฟังก์ชัน Google Apps Script ในฝั่ง Server-side เพื่อที่จะมีปฏิสัมพันธ์กับกรอบโต้ตอบต่างๆ เช่น Sidebar เป็นต้น

showModalDialog(userInterface, title) – เมธอดในคลาส Ui

<https://developers.google.com/apps-script/reference/base/ui#showmodaldialoguserinterface,-title>

ใช้เปิด Dialog โดยไม่หยุดการรันโค้ด การสื่อสารกับสคริปต์ฝั่ง Server-side ฝั่ง Client-side จะต้องเรียกใช้ API google.script เพื่อเรียกใช้คลาส HtmlService เช่น เวลาจะปิด Dialog เรียกใช้ Method google.script.host.close() ในฝั่ง Client-side

พารามิเตอร์

Name	Type	Description
userInterface	Object	An HtmlOutput representing the interface to display.
title	String	The title of the dialog; overrides any title set by calling setTitle() on the userInterface object.

16.4.ก.) ตัวอย่าง : แสดง Dialog

โค้ดในตัวอย่างนี้มี 2 ไฟล์ ก็คือ ไฟล์ .gs และ ไฟล์ .html การรันโค้ดทำได้ 2 วิธี ก็คือ รันฟังก์ชัน onOpen หรือ เปิดไฟล์ Google Sheet (ใช้วิธีรีเฟรช) ก็ได้

ไฟล์ Page.html มีโค้ดดังต่อไปนี้

```
<!DOCTYPE html>
<html>
  <head>
    <base target="_top">
  </head>
  <body>
    <!-- ปุ่ม                                เมื่อคลิกให้ปิด Dialog -->
    Hello, world! <input type="button" value="Close" onclick="google.script.host.close()" />
    <!-- google.script.host.close() ใช้ปิด Custom dialog -->
  </body>
</html>
```

ไฟล์ Code.gs มีโค้ดดังต่อไปนี้

```
function onOpen() { // รันอัตโนมัติเมื่อเปิดไฟล์ Google Sheets

    SpreadsheetApp.getUi() // สร้างเมนู Custom Menu → Show dialog
        .createMenu('Custom Menu')
        .addItem('Show dialog', 'showDialog')
        .addToUi() ;

}

function showDialog() {
    var html = HtmlService.createHtmlOutputFromFile('Page')
        .setWidth(400) // ขนาดของกรอบ Dialog
        .setHeight(300) ; // ขนาดของกรอบ Dialog

    SpreadsheetApp.getUi() // Or DocumentApp or SlidesApp or FormApp.
        .showModalDialog(html, 'My custom dialog') ;
}
```

เมื่อรันฟังก์ชัน onOpen หรือ เปิดไฟล์ จากนั้นไปที่เมนู Custom Menu → Show dialog จะได้ผลลัพธ์ดังต่อไปนี้ - เนื้อหาและรูปแบบของกรอบ Dialoge ออกแบบจากไฟล์ Page.html



16.5. Custom sidebars

Custom sidebars

https://developers.google.com/apps-script/guides/dialogs#custom_sidebars

Custom sidebars เป็น Sidebar ที่เราออกแบบเอง โดยใช้เซอว์วิส HTML ในการสร้าง ในรูปแบบหน้าเว็บ HTML ฉะนั้นเวลารันสคริปต์ สคริปต์จะไม่หยุดรันเมื่อ Dialog ถูกเปิด

`showSidebar(userInterface)` – เมธอดในคลาส Ui

<https://developers.google.com/apps-script/reference/base/ui#showsidebaruserinterface>

ใช้เปิด Sidebar ที่มีเนื้อหาข้างในเป็นแบบ Client-side ก็คือ เนื้อหาที่กำหนดเองได้ และทำงานในฝั่ง Client

พารามิเตอร์

Name	Type	Description
<code>userInterface</code>	Object	An HtmlOutput representing the interface to display.

ตัวอย่าง โค้ดในตัวอย่างนี้มี 2 ไฟล์ ก็คือ ไฟล์ .gs และไฟล์ .html การรันโค้ดทำได้ 2 วิธี ก็คือ รันฟังก์ชัน onOpen หรือ เปิดไฟล์ Google Sheet (ใช้วิธีเพรช) ก็ได้

ไฟล์ Page.html มีโค้ดดังต่อไปนี้

```
<!DOCTYPE html>
<html>
  <head>
    <base target="_top">
  </head>
  <body>
    <!-- ปุ่ม เมื่อคลิกให้ปิด Dialog -->
    Hello, world! <input type="button" value="Close" onclick="google.script.host.close()" />
    <!-- google.script.host.close() ใช้ปิด Custom sidebars -->
  </body>
</html>
```

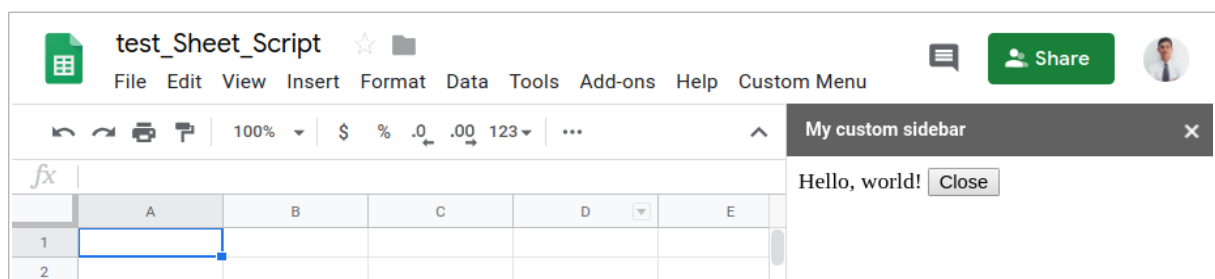
ไฟล์ Code.gs มีโค้ดดังต่อไปนี้

```
function onOpen() { // รันอัตโนมัติเมื่อเปิดไฟล์ Google Sheets
  SpreadsheetApp.getUi() // สร้างเมนู Custom Menu → Show sidebar
    .createMenu('Custom Menu')
    .addItem('Show sidebar', 'showSidebar')
    .addToUi();
}
```

```
function showSidebar() {
  var html = HtmlService.createHtmlOutputFromFile('Page')
    .setTitle('My custom sidebar')
    .setWidth(300) // กำหนดความกว้าง Sidebar

  SpreadsheetApp.getUi().showSidebar(html) ;
}
```

เมื่อรันฟังก์ชัน onOpen หรือ เปิดไฟล์ จากนั้นไปที่เมนู Custom Menu → Show sidebar จะได้ผลลัพธ์ดังต่อไปนี้ - เนื้อหาและรูปแบบของกรอบ Dialog ออกแบบจากไฟล์ Page.html



บทที่ 17

คลาส Browser



17.1. msgBox()

คลาส Browser

<https://developers.google.com/apps-script/reference/base/browser>

ใช้เข้าถึง Dialog boxes ของ Google Sheets เท่านั้น

`msgBox(prompt)` - เมธอดในคลาส Browser

<https://developers.google.com/apps-script/reference/base/browser#msgboxprompt>

`msgBox(prompt, buttons)` - เมธอดในคลาส Browser

<https://developers.google.com/apps-script/reference/base/browser#msgboxprompt,-buttons>

`msgBox(title, prompt, buttons)` - เมธอดในคลาส Browser

<https://developers.google.com/apps-script/reference/base/browser#msgboxtitle,-prompt,-buttons>

`msgBox()` ใช้แสดง Message box ที่มีข้อความและปุ่มต่างๆ

เนื่องจากเมธอด `msgBox()` เปิด Message box ที่รันในฝั่ง Client-side จึงอาจทำให้ Google Apps Script ซึ่งรันในฝั่ง Server-side ค้างได้ จึงแนะนำให้ใช้ Ui Prompt จะดีกว่า

พารามิเตอร์ของ `msgBox()`

Name	Type	Description
<code>title</code>	String	The title of the dialog box.
<code>prompt</code>	String	The text to be displayed in the dialog box.
<code>buttons</code>	ButtonSet	The type of button set to use. (ดูตาราง Enum ButtonSet)

Enum ButtonSet

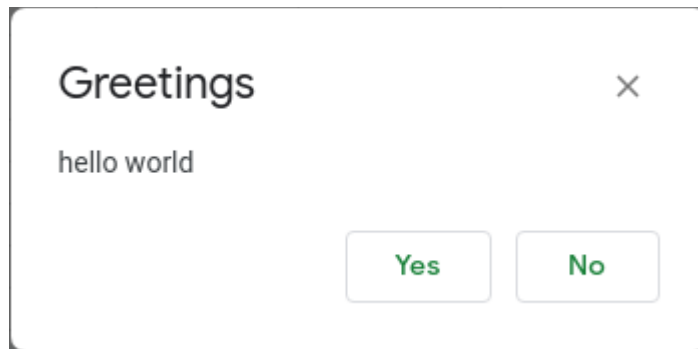
<https://developers.google.com/apps-script/reference/base/button-set.html>

Property	Type	Description
<code>OK</code>	Enum	A single "OK" button, indicating an informational message that can only be dismissed.
<code>OK_CANCEL</code>	Enum	An "OK" button and a "Cancel" button, allowing the user to either proceed with or halt an operation.
<code>YES_NO</code>	Enum	A "Yes" button and a "No" button, allowing the user to answer a yes/no question.
<code>YES_NO_CANCEL</code>	Enum	A "Yes" button, a "No" button, and a "Cancel" button, allowing the user to either answer a yes/no question or halt an operation.

ตัวอย่าง

```
// แสดง Dialog box - มี Title + ปุ่ม Yes + No  
Browser.msgBox('Greetings', 'hello world', Browser.Buttons.YES_NO) ;
```

ผล



17.2. inputBox()

`inputBox(prompt)` - เมธอดในคลาส Browser

<https://developers.google.com/apps-script/reference/base/browser#inputboxprompt>

`inputBox(prompt, buttons)` - เมธอดในคลาส Browser

<https://developers.google.com/apps-script/reference/base/browser#inputboxprompt,-buttons>

`inputBox(title, prompt, buttons)`

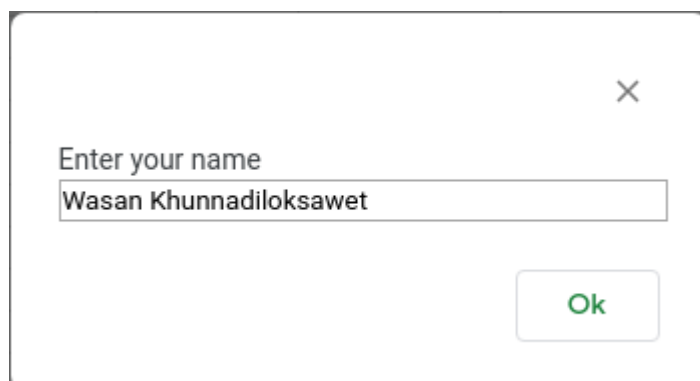
<https://developers.google.com/apps-script/reference/base/browser#inputboxtitle,-prompt,-buttons>

`inputBox` ใช้งานคล้ายกับ `msgBox` พารามิเตอร์ก็เหมือนกัน เพียงแต่ใช้รับค่าเข้าไป

ตัวอย่างการใช้งาน

```
function myFunction() {  
  var name = Browser.inputBox('Enter your name'); // จับค่าที่กรอกลงใน inputBox  
  
  SpreadsheetApp.getActiveSpreadsheet()  
    .getActiveSheet()  
    .getRange('A1')  
    .setValue(name);  
}
```

ผล - หลังจากรันสคริปต์แล้ว ให้ไปดูที่ไฟล์ Google Sheet จะปรากฏ Input box



หลังกรอกข้อมูล และ คลิกปุ่ม OK จะกรอกชื่อที่พิมพ์ลงใน inputBox ลงในเซลล์ A1

	A	
1	Wasan Khunnadiloksawet	
2		

บทที่ 18

การประยุกต์ใช้งาน



18.1. การกรองแถวซ้ำ

ที่มา https://developers.google.com/apps-script/articles/removing_duplicates

ตัวอย่างข้อมูลที่มีแถวซ้ำ(ทั้งแถว) ตัวอย่างตามภาพ

	A	B	C	D
1	Fruit	Cost	Quantity	Total
2	Apple	15	200	3,000.00 บาท
3	Bananas	50	520	26,000.00 บาท
4	Mangoes	20	1,025	20,500.00 บาท
5	Mangoes	20	123	20,500.00 บาท
6	Durian	150	50	7,500.00 บาท
7	Durian	150	50	7,500.00 บาท
8	Longan	60	450	27,000.00 บาท
9	Total			132,500.00 บาท

ให้สร้างโปรเจ็ค Google Apps Script แบบฝังในไฟล์ Google Sheet จากนั้น ใส่โค้ดดังต่อไปนี้ เมื่อรันโค้ด แถวซ้ำใน Google Sheet จะหายไป

```
function removeDuplicates() {  
  
    var sheet = SpreadsheetApp.getActiveSheet() ;  
    var data = sheet.getDataRange().getValues() ;           // จับ Data ทั้งหมดในชีท  
  
    var newData = [] ;           // สร้างอาร์เรย์ว่างๆ ไว้รอเก็บผลลัพธ์จากการกรอง  
  
    for (var i in data) {           // for – ชั้นที่ 1  
        var row = data[i] ;           // จับทีละแถว  
        // Logger.log(row);           // พิมพ์ : [Apple, 15.0, 200.0, 3000.0]  
  
        var duplicate = false ; // เริ่มต้น Set เป็น false ไว้ก่อน  
  
        for (var j in newData) {           // for – ชั้นที่ 2  
            if (row.join() == newData[j].join()) {  
                // Apple,15,200,3000 == ,, -----> แถวไม่ซ้ำ ซ้ำมไป  
                // ถ้าแถวซ้ำกัน  
                duplicate = true ; // เจอแถวซ้ำ Set เป็น true  
            } // if  
        } // for – ชั้นที่ 2  
  
        // ถ้าเป็น !false(true) หรือ แถวไม่ซ้ำ -----> จับใส่ newData  
        if (!duplicate) {  
            newData.push(row);  
        } // if  
    } // for – ชั้นที่ 1  
  
    // จับอาร์เรย์ผลลัพธ์ วางกลับไปแทนที่  
    sheet.clearContents();  
    sheet.getRange(1, 1, newData.length, newData[0].length).setValues(newData);  
}
```


อธิบายการทำงานของโค้ดข้างต้น

สร้างอาร์เรย์ผลลัพธ์ว่างๆ ไว้รอรับผลจากการกรอง ก็คือ `var newData = []` ;

จับข้อมูลทั้งหมดในชีทมาก่อน `var data = sheet.getDataRange().getValues()` จากนั้นจับมาทีละแถวโดยใช้ลูป `for/in` (ขั้นที่1) สิ่งที่ได้จับมาได้ เช่น `[Apple, 15.0, 200.0, 3000.0]`

จากนั้นนำไปประมวลผลต่อ ในลูป `for/in` (ขั้นที่2) โดยรวมข้อมูลในแถวที่จับมาได้โดยใช้ Method `join` ก็คือ `row.join()` ผลที่ได้เช่น `Apple,15,200,3000`

เมื่อรวมแล้ว นำไปเปรียบเทียบกับ สมาชิกในแถวของอาร์เรย์ผลลัพธ์ที่ Join แล้ว ก็คือ `newData[j].join()` โดยเปรียบเทียบทีละแถว ไปจนครบทุกแถว (เหตุนี้จึงต้องใช้ `for/in` ขั้นที่ 2)

ตัวอย่างการเปรียบเทียบ

`Apple,15,200,3000 == ,,,` // คำนวณค่า false

ถ้าคำนวณค่า false หมายถึง ข้อมูลไม่ซ้ำ ก็ให้จับแถวนั้น ยัดเข้าไปในอาร์เรย์ว่างโดยใช้ Method `push` ก็คือ `newData.push(row)`

ตัวอย่างการเปรียบเทียบครั้งที่ 2 - หลังจากยัดแถวข้อมูลเข้าไปในอาร์เรย์ผลลัพธ์แล้ว

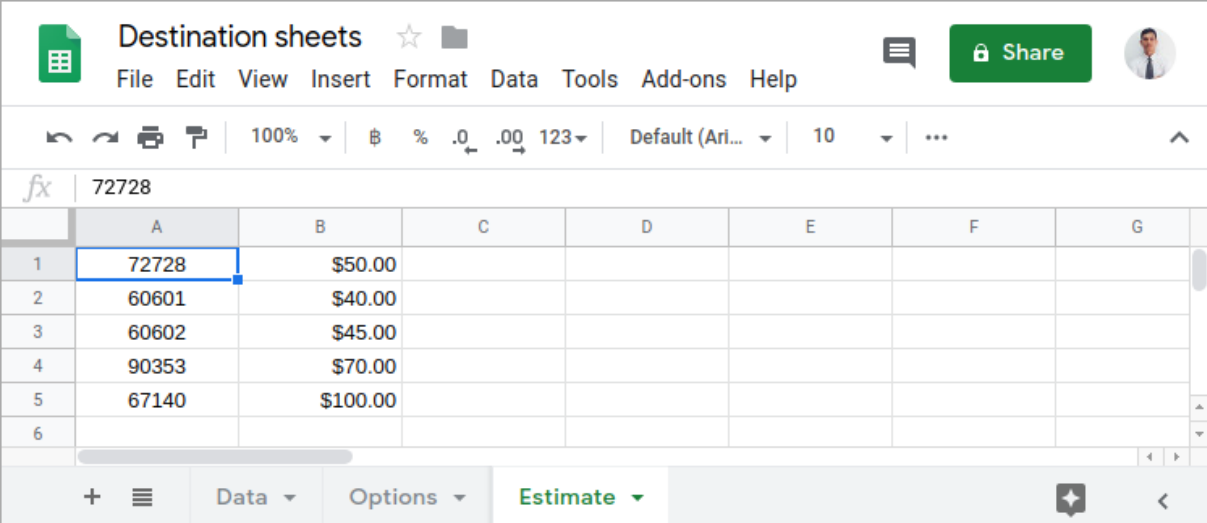
`Apple,15,200,3000 == Apple,15,200,3000` // คำนวณค่า true

ถ้าคำนวณค่า true ก็คือ แถวข้อมูลต้นทาง กับ แถวข้อมูลในอาร์เรย์ผลลัพธ์ ซ้ำกัน ก็ให้ข้ามไปจับแถวใหม่มาเปรียบเทียบต่อ ไม่มีการยัดแถวซ้ำ เข้าไปในอาร์เรย์ผลลัพธ์

18.2. Data Lookup

ตัวอย่างนี้ ทำคล้ายกับ VLOOKUP โดยส่งข้อมูลที่เป็นคีย์ไปค้นหาในตารางข้อมูล หากเจอที่ตรงกัน ก็ให้คืนค่าในคอลัมน์อื่น ในแถวเดียวกันกลับมา

ตัวอย่างข้อมูลในชีท มีดังต่อไปนี้



	A	B	C	D	E	F	G
1	72728	\$50.00					
2	60601	\$40.00					
3	60602	\$45.00					
4	90353	\$70.00					
5	67140	\$100.00					
6							

```

function LookUp(){
    var cost = getCost(60602) ;
    Logger.log(cost) ; // ฿45.00
}

function getCost(zipCode){

    zipCode = Number(zipCode) ; // แปลงค่าที่จับมาเป็นตัวเลข เพื่อความมั่นใจ

    var id = "<< Spreadsheet Id >>" ;
    var ss = SpreadsheetApp.openById(id) ;
    var ws = ss.getSheetByName("Estimate") ;
    var data = ws.getRange(1,1,ws.getLastRow(),2).getValues() ; // จับข้อมูล 2 คอลัมน์

    var zipCodesList = data.map(function(r){ return r[0]; }); // คอลัมน์ที่ 1
    // [72728, 60601, 60602, 90353, 67140]

    var costList = data.map(function(r){ return r[1]; }); // คอลัมน์ที่ 2
    // [50.0, 40.0, 45.0, 70.0, 100.0]

    var position = zipCodesList.indexOf(zipCode) ; // ***
    // 2 <-- [72728, 60601, 60602, 90353, 67140] <---- 60602

    if(position > -1){ // กรณีหาเจอ
        return "฿"+costList[position].toFixed(2) ;
    } else { // กรณีหาไม่เจอ
        return "Unavailable" ;
    } ;
}

```

บทที่ 19

วันและเวลา



19.1. สร้างวัตถุวันที่และเวลา

Working with Dates and Times

<https://developers.google.com/google-ads/scripts/docs/features/dates>

JavaScript Date Output

https://www.w3schools.com/js/js_dates.asp

JavaScript Date Formats

https://www.w3schools.com/js/js_date_formats.asp

19.1.ก.) new Date()

new Date() ใช้สร้างวัตถุวันที่ตัวใหม่ ซึ่งสามารถสร้างได้ 4 วิธี

```
new Date()  
new Date(year, month, day, hours, minutes, seconds, milliseconds)  
new Date(milliseconds)  
new Date(date string)
```

ตัวอย่าง

```
var d1 = new Date(2018, 11, 24, 10, 33, 30, 0) ;  
        // Mon Dec 24 2018 10:33:30 GMT+0700 (Indochina Time)  
  
var d2 = new Date(2018, 11, 24, 10, 33) ;  
        // Mon Dec 24 2018 10:33:00 GMT+0700 (Indochina Time)  
  
var d3 = new Date(2018, 11, 24, 10) ;  
        // Mon Dec 24 2018 10:00:00 GMT+0700 (Indochina Time)  
  
var d4 = new Date("October 13, 2014 11:13:00") ;  
        // Mon Oct 13 2014 11:13:00 GMT+0700 (Indochina Time)  
  
var d5 = new Date("2015-03-25") ;  
        // Wed Mar 25 2015 07:00:00 GMT+0700 (Indochina Time)  
  
var d6 = new Date(99, 11, 24) ;  
        // Fri Dec 24 1999 00:00:00 GMT+0700 (Indochina Time)  
  
var d7 = new Date(9, 11, 24) ;  
        // Fri Dec 24 1909 00:00:00 GMT+0642 (Indochina Time)  
  
var d8 = new Date(0) ;  
        // Thu Jan 01 1970 07:00:00 GMT+0700 (Indochina Time)  
        // วันและเวลาอ้างอิง  
  
var d9 = new Date(1000000000000) ;  
        // Sat Mar 03 1973 16:46:40 GMT+0700 (Indochina Time)  
        // ผ่านมา 1000000000000 มิลลิวินาที นับจากวันและเวลาอ้างอิง  
  
var d10 = new Date(-1000000000000) ;  
        // Mon Oct 31 1966 21:13:20 GMT+0700 (Indochina Time)  
        // ย้อนไป 1000000000000 มิลลิวินาที นับจากวันและเวลาอ้างอิง
```

19.1.ข.) +new Date()

ใส่เครื่องหมาย + หน้า new จะได้ค่ามิลินาทีของวันที่ เช่น

```
var d1 = new Date() ;    // เช่น Sun May 03 2020 09:52:43 GMT+0700 (Indochina Time)
var d2 = +new Date() ;   // เช่น 1588474363200
```

19.2. การทำงานกับวันที่และเวลา

JavaScript Get Date Methods

https://www.w3schools.com/js/js_date_methods.asp

JavaScript Set Date Methods

https://www.w3schools.com/js/js_date_methods_set.asp

JavaScript Date Formats

https://www.w3schools.com/js/js_date_formats.asp

คลาส Utilities

<https://developers.google.com/apps-script/reference/utilities/utilities#formatdatedate-timezone-format>

setNumberFormat([numberFormat](#)) – เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setnumberformatnumberformat>

setNumberFormats([numberFormats](#)) – เมธอดในคลาส Range

<https://developers.google.com/apps-script/reference/spreadsheet/range#setnumberformatsnumberformats>

formatDate([date](#), [timeZone](#), [format](#)) – เมธอดในคลาส Utilities

<https://developers.google.com/apps-script/reference/utilities/utilities#formatdatedate,-timezone,-format>

<http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html> - Java SE SimpleDateFormat

formatDate() ใช้จัดรูปแบบวันที่ โดยใช้แพทเทิร์นตามที่อธิบายไว้ในคลาส Java SE

SimpleDateFormat (ตามมลิ่งข้างบน) โดยจะคืนค่ากลับมาเป็นข้อความ(String)

พารามิเตอร์

Name	Type	Description
date	Date	a Date to format as a String
timeZone	String	the output timezone of the result
format	String	a format per the SimpleDateFormat specification

ตัวอย่าง

```
var formattedDate = Utilities.formatDate( new Date()
                                          , "GMT"      // "GMT+7" for Thailand
                                          , "yyyy-MM-dd' T 'HH:mm:ss' Z '" );
Logger.log(formattedDate) ;
```

ตัวอย่าง

ตัวอย่างโค้ดที่เกี่ยวข้องกับวันที่หรือเวลา

```
function setDate() {  
    var sheet = SpreadsheetApp.getActiveSheet() ;  
    var A1 = sheet.getRange("A1").clearContent() ;  
    var A2 = sheet.getRange("A2").clearContent() ;  
  
    var t_date = Utilities.formatDate(new Date(), "GMT+7", "dd MMM yyyy เวลา HH:mm:ss") ;  
  
    A1.setValue(t_date) ;  
    // พิมพ์วันที่ลงในเซลล์ A1 – Type เป็น TEXT  
  
    var n_Date = new Date() ; // Sun Nov 17 08:40:57 GMT+07:00 2019  
  
    A2.setValue(n_Date).setNumberFormat("yyyy MMM dd HH:mm:ss") ;  
    // พิมพ์วันที่ลงในเซลล์ A2 – Type เป็น NUMBER  
  
    Logger.log(n_Date.getDate()) ;           // พิมพ์ : 20.0  
    Logger.log(n_Date.getMonth()) ;          // พิมพ์ : 10.0  
    Logger.log(n_Date.getYear()) ;           // พิมพ์ : 2019.0  
    Logger.log(n_Date.getFullYear()) ;       // พิมพ์ : 2019.0  
    Logger.log(n_Date.getTime().toString()) ; // พิมพ์ : 1574219076765  
    Logger.log(n_Date.getHours()) ;          // พิมพ์ : 4.0  
    Logger.log(n_Date.getMinutes()) ;        // พิมพ์ : 36.0  
    Logger.log(n_Date.getSeconds()) ;        // พิมพ์ : 10.0  
}
```

ผล

	A	B
1	20 Nov 2019 เวลา 10:04:36	(Text)
2	2019 พ.ย. 20 10:04:37	(Number)

อธิบายโค้ด

เริ่มจาก สร้างข้อมูลเวลาขึ้นมาก่อน โดยใช้โค้ด `new date()` ถ้า Log ดู จะได้ผลลัพธ์ตามตัวอย่าง

```
Logger.log(new date())  
// พิมพ์ : Wed Nov 20 09:19:08 GMT+07:00 2019
```

เมื่อได้ข้อมูลเวลามาแล้ว จากนั้นก็สามารถใช้ Method อย่างเช่น `getYear()`, `getMonth()`, `getDate()` เข้าไปจับข้อมูลส่วนต่างๆในเวลา เช่น ปี เดือน วัน ออกมาใช้ได้ เป็นต้น

หากต้องการจัดรูปแบบวันที่หรือเวลา สามารถทำได้โดยใช้ Method `formatDate` ในคลาส `Utilities` ผลลัพธ์ที่ได้มา มีชนิดข้อมูลเป็น Text เช่น

```
Utilities.formatDate(new Date(), "GMT+7", "dd MMM yyyy เวลา HH:mm:ss")
// ถ้า Log จะได้ผลลัพธ์เช่น 20 Nov 2019 เวลา 09:19:28
```

หากเรากรอกเวลาลงไปในเซลล์ ในไฟล์ Google Sheets และต้องการจัดรูปแบบตัวเลข วันที่หรือเวลา สามารถทำได้โดยใช้ Method `setNumberFormat()` ในคลาส `Range` เช่น

```
var n_Date = new Date(); // Sun Nov 17 08:40:57 GMT+07:00 2019
A2.setValue(n_Date).setNumberFormat("yyyy MMM dd HH:mm:ss");
```

Method `getTime()` ใช้จับข้อมูลเวลาที่สร้างขึ้นมา ให้ออกมาเป็นตัวเลขมิลลิวินาที นับจากวันที่ January 1, 1970 UTC. ซึ่งเป็นวันที่อ้างอิง เช่น

```
var myDate = new Date();
Logger.log(myDate.getTime().toString()); // พิมพ์ : 1574217590425
```

`getTime()` เป็น Method ของเวลาที่เป็นพื้นฐานสำคัญในการคำนวณเกี่ยวกับเวลา

19.3. การคำนวณเกี่ยวกับเวลา

Date Math

https://developers.google.com/google-ads/scripts/docs/features/dates#date_math

Method `getTime()` ใช้จับข้อมูลเวลาที่สร้างขึ้นมา ให้ออกมาเป็นตัวเลขมิลลิวินาที โดยนับจากวันที่ January 1, 1970 UTC.

(1 วัน มี $1000 * 60 * 60 * 24 = 86,400,000$ วินาที)

ตัวอย่างการคำนวณความแตกต่างระหว่าง 2 วันที่

```
function dateMath(){
    var date1 = new Date('November 18, 2019 19:30:00 -0000');
    var date2 = new Date('November 21, 2019 21:00:00 -0000');
    var diffTime = (fDate.getTime() - sDate.getTime())/1000/60/60 ;
    Logger.log(fDate);
    Logger.log(sDate);
    Logger.log(diffTime);
}
```

ผล

```
[ 01 ] Mon Nov 18 19:30:00 GMT+07:00 2019
[ 02 ] Thu Nov 21 21:00:00 GMT+07:00 2019
[ 03 ] 73.5 // หน่วยเป็น ชม.
```

19.4. setHours()

setHours()

https://www.w3schools.com/jsref/jsref_sethours.asp

Method ที่ใช้กำหนดเวลาให้กับวันที่

สัญลักษณ์

```
Date.setHours(hour, min, sec, millisec)
```

พารามิเตอร์

Parameter	Description
hour	Required. An integer representing the hour. Expected values are 0-23, but other values are allowed <ul style="list-style-type: none">1 will result in the last hour of the previous day24 will result in the first hour of the next day
min	Optional. An integer representing the minutes. Expected values are 0-59, but other values are allowed: <ul style="list-style-type: none">1 will result in the last minute of the previous hour60 will result in the first minute of the next hour
sec	Optional. An integer representing the seconds Expected values are 0-59, but other values are allowed: <ul style="list-style-type: none">1 will result in the last second of the previous minute60 will result in the first second of the next minute
millisec	Optional. An integer representing the milliseconds Expected values are 0-999, but other values are allowed: <ul style="list-style-type: none">1 will result in the last millisecond of the previous second1000 will result in the first millisecond of the next second

ตัวอย่าง

```
var d = new Date();

Logger.log(d); // Tue Feb 04 12:13:23 GMT+07:00 2020
Logger.log(d.setHours(15, 35, 1).toString()) ; // 1580805301061
Logger.log(d); // Tue Feb 04 15:35:01 GMT+07:00 2020

var formattedDate = Utilities.formatDate(d
    , "GMT+7"
    , "yyyy-MM-dd' T 'HH:mm:ss' Z ' ");

Logger.log(formattedDate) ; // 2020-02-04 T 15:35:01 Z
```


บทที่ 20
ฟังก์ชัน, เมทอด และ
คุณสมบัติที่น่าสนใจ



20.1. toString()

JavaScript Number toString() Method

https://www.w3schools.com/jsref/jsref_tostring_number.asp

toString() เป็น Method ของคลาส Number(ตัวเลข) ใช้แปลงจากตัวเลขเป็นข้อความ

```
var num = 15 ;

Logger.log(num) ;           // พิมพ์ : 15.0
Logger.log(typeof num) ;    // พิมพ์ : number

var n = num.toString() ;

Logger.log(n) ;             // พิมพ์ : 15
Logger.log(typeof n) ;      // พิมพ์ : string
```

สามารถใส่ Argument ให้กับ toString() ได้ หมายถึง แปลงเป็นเลขฐานตามที่ระบุ เช่น toString(2) ก็คือ แปลงเป็นเลขฐาน 2

```
var num1 = 155;

Logger.log(num1.toString(2));           // พิมพ์ : 10011011
Logger.log(num1.toString(16));          // พิมพ์ : 9b

var num2 = -2051;

Logger.log(num2.toString(2));           // พิมพ์ : -1000000000011
Logger.log(num2.toString(16));          // พิมพ์ : -803
```

20.2. String()

JavaScript String() Function

https://www.w3schools.com/jsref/jsref_string.asp

ใช้แปลงค่าของอ็อบเจกต์เป็นสตริง

โครงสร้างการใช้งาน

```
String(object)
```

พารามิเตอร์

Parameter	Description
object	(Required) A JavaScript object

ตัวอย่าง

```
var x1 = Boolean(0);
var x2 = Boolean(1);
var x3 = new Date();
var x4 = "12345" ;
var x5 = 12345 ;

var res = String(x1) + "\n" +
          String(x2) + "\n" +
          String(x3) + "\n" +
          String(x4) + "\n" +
          String(x5) ;

Logger.log(res) ;
```

ผล

```
Logs
[ ]
false
true
Sun May 10 2020 12:34:08 GMT+0700 (Indochina Time)
12345
12345
```

20.3. toLowerCase() และ toUpperCase()

JavaScript String toLowerCase()

https://www.w3schools.com/jsref/jsref_tolowercase.asp

JavaScript String toUpperCase()

https://www.w3schools.com/jsref/jsref_toupper.asp

toLowerCase() และ toUpperCase() เป็น เมธอดในคลาส String(ข้อความ) ใช้แปลงตัวอักษรในข้อความให้เป็นตัวพิมพ์เล็กทั้งหมด หรือ พิมพ์ใหญ่ทั้งหมด ตามลำดับ

```
var str = "Hello World!" ;

Logger.log(str.toLowerCase()) ;           // พิมพ์ : hello world!
Logger.log(str.toUpperCase()) ;           // พิมพ์ : HELLO WORLD!
```

20.4. PI

JavaScript PI Property

https://www.w3schools.com/jsref/jsref_pi.asp

PI เป็นคุณสมบัติ(Property) ในคลาส Math เป็นตัวเลขค่าคงที่ของ π ก็คือ 3.14159...

```
function myFunction() {  
    Logger.log(Math.PI);    // พิมพ์ : 3.141592653589793  
}
```

20.5. isNaN()

JavaScript isNaN() Function

https://www.w3schools.com/jsref/jsref_isnan.asp

isNaN() เป็นฟังก์ชันที่ใช้ทดสอบตัวแปรหรือค่า ว่าเป็น NaN (Not-a-Number) หรือไม่

ตัวอย่าง

```
isNaN(123)           // false  
isNaN(-1.23)         // false  
isNaN(5-2)           // false  
isNaN(0)              // false  
isNaN('123')          // false – ใส่เป็นข้อความแต่ isNaN มองเป็น Number  
isNaN('Hello')        // true  
isNaN('2005/12/12')   // true  
isNaN("")             // false  
isNaN(true)           // false  
isNaN(undefined)     // true  
isNaN('NaN')          // true  
isNaN(NaN)            // true  
isNaN(0 / 0)          // true
```

ตัวอย่าง

```
// var iAge = Browser.inputBox("กรอกอายุ") ;  
var iAge = 12 ;  
var Age = Number(iAge) ;  
if (isNaN(Age)) {  
    Logger.log("Input is not a number") ;  
} else {  
    Logger.log((Age < 18) ? "Too young" : "Old enough") ;  
}
```

20.6. Number()

Number() - JavaScript Function

https://www.w3schools.com/jsref/jsref_number.asp

`number()` เป็นฟังก์ชันที่ใช้แปลงค่าให้เป็นตัวเลข หากแปลงไม่ได้จะคืนค่า NaN กลับมา ก็คือ ไม่ใช่ทั้งตัวเลขทั้งข้อมูลและหน้าตา

```
var x1 = true ;
var x2 = false ;
var x3 = new Date() ;
var x4 = "999" ;
var x5 = "999 888" ;

Logger.log(Number(x1)) ; // พิมพ์ : 1.0
Logger.log(Number(x2)) ; // พิมพ์ : 0.0
Logger.log(Number(x3)) ; // พิมพ์ : 1.572579859377E12
Logger.log(Number(x4)) ; // พิมพ์ : 999
Logger.log(Number(x5)) ; // พิมพ์ : NaN
```

20.7. split()

JavaScript String split() Method

https://www.w3schools.com/jsref/jsref_split.asp

Method ของข้อความ(String) ใช้แยกข้อความ แล้วเก็บข้อความที่แยกแล้วไว้ในอาร์เรย์

โครงสร้างการใช้งาน Split

```
string.split(separator, limit)
```

พารามิเตอร์

Parameter	Description
<code>separator</code>	(Optional) Specifies the character, or the regular expression, to use for splitting the string. If omitted, the entire string will be returned (an array with only one item) (ตัวแยกข้อความเช่น ,(Comma) หรือ - (Hyphen) เป็นต้น)
<code>limit</code>	(Optional) An integer that specifies the number of splits, items after the split limit will not be included in the array (จำนวนที่จะเก็บเป็นสมาชิกในอาร์เรย์)

`separator` หากไม่ระบุพารามิเตอร์ ,(Comma) จะถูกใช้เป็นตัวแยกข้อความโดยปริยาย และ แยกข้อความได้เท่าไร ก็จะเก็บไว้ในอาร์เรย์ทั้งหมด

ตัวอย่าง

```
var str = "How are you doing today?" ;
var res = str.split(" ") ;
Logger.log(res) ;
Logger.log(res.length) ;
```

ผล

```
Logs
[ ] [How, are, you, doing, today?]
[ ] 5.0
```

20.8. parseInt()

parseInt() - JavaScript Global Functions

https://www.w3schools.com/jsref/jsref_parseint.asp

ใช้วิเคราะห์และแปลงจากข้อความ(String) ไปเป็นเลขจำนวนเต็ม(Integer) และปัดลงอย่างเดียว

โครงสร้างการใช้งาน

```
parseInt(string, radix)
```

พารามิเตอร์

Parameter	Description
string	(ต้องการ) String(หรือตัวเลข) ที่จะแปลง
radix	(เว้นได้) ตัวเลข 2 ถึง 36 แสดงถึงเลขฐานที่จะแปลงไปเป็น หากไม่ระบุจะหมายถึงเลขฐาน 10

ถ้ามีการใช้พารามิเตอร์ **radix**

ถ้าข้อความ(String) ขึ้นต้นด้วย "0x" ความหมายก็คือ ใช้ **radix = 16** (hexadecimal)

ถ้าข้อความ(String) ขึ้นต้นด้วย "0" ความหมายก็คือ ใช้ **radix = 8** (octal)

ถ้าข้อความ(String) ขึ้นต้นด้วยตัวอื่นๆ **radix = 10** (decival)

ตัวอย่าง

```
var a = parseInt("10") ;           // 10.0
var b = parseInt("10.00") ;        // 10.0
var c = parseInt("10.33") ;        // 10.0
var d = parseInt("34 45 66") ;     // 34.0 - เฉพาะเลขตัวหน้าเท่านั้น
var e = parseInt(" 60 ") ;         // 60.0 - ช่องว่างหัวและท้ายมีได้ แต่จะถูกตัด
var f = parseInt("40 years") ;     // 40.0
var g = parseInt("He was 40") ;    // NaN - ขึ้นด้วยอักษร
var h = parseInt("10", 10) ;       // 10.0
var i = parseInt("010") ;          // 8.0 - ขึ้นด้วย 0 ,radix = 8
var j = parseInt("10", 8) ;        // 8.0
var k = parseInt("0x10") ;         // 16.0 - ขึ้นด้วย 0x ,radix = 16
var l = parseInt("10", 16) ;       // 16.0
```

ตัวอย่างการใช้ parseInt() แปลงเลขฐานต่างๆ เป็นเลขฐาน 10

```
function bi3dec() {  
  var a = parseInt("10111", 2); // 23 (เลขฐาน 10) = 10111 (เลขฐาน 2)  
  var b = parseInt("10001", 2); // 17 (เลขฐาน 10) = 10001 (เลขฐาน 2)  
  var c = parseInt("10111", 16); // 65809 (เลขฐาน 10) = 10111 (เลขฐาน 16)  
  var d = parseInt("10001", 16); // 65537 (เลขฐาน 10) = 10001 (เลขฐาน 16)  
  var e = parseInt("244F", 16); // 9295 (เลขฐาน 10) = 244F (เลขฐาน 16)  
}
```

20.9. charAt()

JavaScript String charAt() Method

https://www.w3schools.com/jsref/jsref_charat.asp

คืนค่าเป็นอักขระ ณ ตำแหน่งที่ระบุ

โครงสร้างการใช้งาน

```
string.charAt(index)
```

พารามิเตอร์

Parameter	Description
index	(Required) ตัวเลข ระบุตำแหน่งของตัวอักษรที่ต้องการคืนค่ากลับมา

ตัวอย่าง

```
var str = "HELLO WORLD" ;  
var res = str.charAt(str.length-1) ; // D
```

20.10. charCodeAt()

JavaScript String charCodeAt() Method

https://www.w3schools.com/jsref/jsref_charcodeat.asp

HTML Unicode (UTF-8) Reference

https://www.w3schools.com/charsets/ref_html_utf8.asp

charAt() คืนค่ากลับมาเป็นรหัส Unicode ของตัวอักษร ที่อยู่ในสตริง ที่ระบุตำแหน่งโดย

พารามิเตอร์ [index](#)

โครงสร้างการใช้งาน

```
string.charCodeAt(index)
```

พารามิเตอร์

Parameter	Description
index	(Required) ตัวเลข ระบุตำแหน่งของตัวอักษรที่ต้องการคืนค่ากลับมา

ตัวอย่าง

```
var str = "HELLO WORLD" ;  
var res = str.charCodeAt(str.length-1) ; // 68
```

20.11. fromCharCode()

JavaScript String fromCharCode() Method

https://www.w3schools.com/jsref/jsref_fromcharcode.asp

แปลงรหัส Unicode ไปเป็นตัวอักษร เมธอดนี้ใช้กับวัตถุ String โครงสร้างการใช้งาน ใช้งานแบบนี้เสมอ

ก็คือ **String.fromCharCode()**

โครงสร้างการใช้งาน

```
String.fromCharCode(n1, n2, ..., nX)
```

พารามิเตอร์

Parameter	Description
n1, n2, ..., nX	(Required) ตัวเลข Unicode ใส่เรียงกันไปขึ้นด้วย ,

ตัวอย่าง

```
var res = String.fromCharCode(72, 69, 76, 76, 79) ;  
Logger.log(res) ; // HELLO
```

20.12. trim()

JavaScript String trim() Method

https://www.w3schools.com/jsref/jsref_trim_string.asp

โครงสร้างการใช้งาน

```
string.trim()
```

*** ไม่ Trim ตรงกลาง เช่น ตรงกลางมี 3 Spaces ก็จะปล่อยไว้อย่างนั้น

ตัวอย่าง

```
var str = "    Hello World!    " ;  
Logger.log(str.trim()) ; // Hello World!
```

หรือมีค่าเท่ากับ การใช้ replace() ต่อไปนี้

```
function myTrim(x) {  
    return x.replace(/^\s+|\s+$/gm, "");  
}
```


20.13. substr() และ substring()

JavaScript String substr() Method

https://www.w3schools.com/jsref/jsref_substr.asp

ใช้ตัดตัดอักษรจากข้อความออกมา โดยระบุจุดเริ่มต้นและความยาวที่จะตัด

โครงสร้างการใช้งาน

```
string.substr(start, length)
```

พารามิเตอร์

Parameter	Description
start	(Required) ตำแหน่งเริ่มต้นตัด โดยตำแหน่งแรกคือ 0 ถ้าค่าเป็นบวก และมากกว่า หรือเท่ากับ ความยาวของ string จะคืนค่า string ว่าง ถ้าค่าเป็นลบ จะเริ่มจากด้านหลัง ถ้าค่าเป็นลบ และมากกว่า ความยาวของ string ค่าของ start จะถูกเซตเป็น 0
length	(Optional) จำนวนตัวอักษรที่จะตัด ถ้าเว้นไว้จะตัดตัวอักษรทั้งหมดที่เหลือ

ตัวอย่าง

```
var str = "Hello world!" ;  
var res = str.substr(2) ;           // llo world!  
var res = str.substr(0, 1) ;       // H  
var res = str.substr(11, 1) ;      // !
```

JavaScript String substring() Method

https://www.w3schools.com/jsref/jsref_substring.asp

ใช้ตัดตัดอักษรจากข้อความออกมา โดยระบุจุดเริ่มต้นและจุดจบ

โครงสร้างการใช้งาน

```
string.substring(start,end)
```

พารามิเตอร์

Parameter	Description
start	(Required) The position where to start the extraction. First character is at index 0
end	(Optional) The position (up to, but not including) where to end the extraction. If omitted, it extracts the rest of the string

20.14. search()

JavaScript String search() Method

https://www.w3schools.com/jsref/jsref_search.asp

ใช้ค้นหาในข้อความด้วยค่าที่กำหนด โดยจะคืนค่ากลับมาเป็นตำแหน่งของค่าที่พบ แต่ถ้าไม่พบเลย จะคืนค่า -1 กลับมา ค่าที่ค้นหาสามารถเป็น String หรือ Regular Expression (ดูเพิ่มเติมบทที่ 22 : Regular Expression หน้า 191) ก็ได้

โครงสร้างการใช้งาน

```
string.search(searchvalue)
```

พารามิเตอร์

Parameter	Description
searchvalue	(Required) Regular expression หรือ String(จะถูกแปลงเป็น RE โดยอัตโนมัติ)

ตัวอย่างที่ 1 – ค้นหาโดยใช้ String

```
var str = "Mr. Blue has a blue house" ;  
var n = str.search("blue") ;
```

ตัวอย่างที่ 2 – ค้นหาโดยใช้ Regular Expression

```
var str = "Mr. Blue has a blue house" ;  
var n = str.search(/blue/i) ;
```

ตัวอย่างที่ 3 – Regular Expression ใส่ตัวแปรลงไปได้ หากต้องการใช้ ตัวแปรในการสร้าง Regular Expression ให้ใช้ RegExp ช่วย

```
var textToSearch = "Hello world" ;  
var regPatText = 'w'+ '.*' ;  
var regPatNew = new RegExp(regPatText, "gi") ;  
  
var posFound = textToSearch.search(regPat) ;  
Logger.log(posFound) ; // 6
```

20.15. replace()

JavaScript String replace() Method

https://www.w3schools.com/jsref/jsref_replace.asp

JavaScript Regular Expressions

https://www.w3schools.com/js/js_regexp.asp

JavaScript RegExp Reference

https://www.w3schools.com/jsref/jsref_obj_regexp.asp

`replace()` ใช้ค้นหาและแทนที่ด้วยข้อความ ในข้อความตั้งต้น และคืนค่ากลับมาเป็นอีกความที่ถูกแทนที่แล้ว ตัวที่ใช้ค้นหาสามารถใช้เป็น Regular Expression ได้

กรณีใช้ Regular Expression หากพบตัวที่ค้นหา หลายตัวจะคืนค่ากลับมาเป็นตัวแรก หากต้องการคืนค่าหลายตัวให้ จบด้วย `/g` (global modifier)

โครงสร้างการใช้งาน

```
string.replace(searchvalue, newvalue)
```

พารามิเตอร์

Parameter	Description
<code>searchvalue</code>	(Required) The value, or regular expression, that will be replaced by the new value
<code>newvalue</code>	(Required) The value to replace the search value with

ตัวอย่างที่ 1

ตัวค้นหาเป็น Regular Expression และเป็นแบบ Case-sensitive

```
var str = "Mr Blue has a blue house and a blue car" ;
var res = str.replace(/blue/g, "red") ;
Logger.log(res) ;           // Mr Blue has a red house and a red car
```

ตัวอย่างที่ 2

ตัวค้นหาเป็น Regular Expression และเป็นแบบ Case-insensitive

```
var str = "Mr Blue has a blue house and a blue car" ;
var res = str.replace(/blue/g, "red") ;
Logger.log(res) ;           // Mr red has a red house and a red car
```

ตัวอย่างที่ 3

พารามิเตอร์ที่ 2 `newvalue` ใช้เป็นฟังก์ชัน เพื่อนำสิ่งที่ค้นพบไปประมวลผลในฟังก์ชันต่อ

```
var str = "Mr Blue has a blue house and a blue car" ;
var res = str.replace(/bluehousecar/gi, function (x) {
    return x.toUpperCase() ; // แปลงตัวที่พบให้เป็นพิมพ์ใหญ่ทั้งหมด
});
Logger.log(res);           // Mr BLUE has a BLUE HOUSE and a BLUE CAR
```

20.16. match()

JavaScript String match() Method

https://www.w3schools.com/jsref/jsref_match.asp

match() ใช้ค้นหาข้อความที่สอดคล้องกับ Regular Expression และคืนค่ากลับมาเป็น ค่าที่แมทกัน โดยเก็บไว้ในอาร์เรย์ แต่ถ้าไม่แมทจะคืนค่าเป็น **null**

โครงสร้างการใช้งาน

```
string.match(regex)
```

พารามิเตอร์

Parameter	Description
regex	(ต้องการ) Regular Expression

ตัวอย่าง

```
var str = "The rain in SPAIN stays mainly in the plain" ;  
var res = str.match(/ain/gi) ;      // ain,AIN,ain,ain
```

20.17. includes() - Chrome V8

JavaScript String includes() Method

https://www.w3schools.com/jsref/jsref_includes.asp

ใช้ตรวจสอบว่ามีข้อความที่ระบุอยู่ในข้อความตั้งต้นหรือไม่

โครงสร้างการใช้งาน

```
string.includes(searchvalue, start)
```

พารามิเตอร์

Parameter	Description
searchvalue	Required. The string to search for
start	Optional. Default 0. At which position to start the search

ตัวอย่าง

```
var str = "Hello world, welcome to the universe." ;  
var n = str.includes("world") ;  
  
Logger.log(n) ; // true // Chrome V8 ถึงจะใช้ได้
```

20.18. IndexOf()

indexOf()

https://www.w3schools.com/jsref/jsref_indexof.asp

คืนค่ากลับมาเป็นตำแหน่งแรกที่พบในข้อความตั้งต้น โดยจะส่งข้อความค้นหาเข้าไป ค้นหาแบบ case sensitive โดยตำแหน่งเริ่มจาก 0 และจะคืนค่า -1 หากไม่เจอ

โครงสร้างการใช้งาน

```
string.indexOf(searchvalue, start)
```

พารามิเตอร์

Parameter	Description
searchvalue	Required. The string to search for
start	Optional. Default 0. At which position to start the search

ตัวอย่าง

```
var str = "Hello planet earth, you are a great planet." ;  
var n1 = str.indexOf("planet") ;  
Logger.log(n1) ;           // 6.0  
var n2 = str.indexOf("a") ;  
Logger.log(n2) ;           // 8.0  
var n3 = str.indexOf("a",10) ;  
Logger.log(n3) ;           // 14.0
```

20.19. lastIndexOf()

JavaScript String lastIndexOf() Method

https://www.w3schools.com/jsref/jsref_lastindexof.asp

คืนค่ากลับมาเป็นตำแหน่งสุดท้ายที่พบในข้อความตั้งต้น โดยจะส่งข้อความค้นหาเข้าไป ค้นหาแบบ case sensitive โดยตำแหน่งเริ่มจาก 0 และจะคืนค่า -1 หากไม่เจอ

ตัวอย่าง

```
var str = "Hello planet earth, you are a great planet." ;  
var n1 = str.lastIndexOf("planet") ;  
Logger.log(n1) ;           // 36.0  
var n2 = str.lastIndexOf ("a") ;  
Logger.log(n2) ;           // 38.0  
var n3 = str.lastIndexOf ("a",10) ;  
Logger.log(n3) ;           // 8.0
```

20.20. toFixed

JavaScript toFixed() Method

https://www.w3schools.com/jsref/jsref_tofixed.asp

toFixed() ใช้แปลงตัวเลขเป็นข้อความ และเก็บทศนิยมตามที่กำหนด

โครงสร้างการใช้งาน

```
number.toFixed(x)
```

พารามิเตอร์

Parameter	Description
x	(Optional) The number of digits after the decimal point. Default is 0 (no digits after the decimal point)

ตัวอย่าง

```
var num = 5.56789 ;  
var n = num.toFixed(2) ;    // 5.56  
  
var num2 = 5.56789 ;  
var n2 = num2.toFixed() ;   // 6  
  
var num3 = 5.56789 ;  
var n3 = num3.toFixed(10) ; // 5.6789000000
```

20.21. toPrecision()

JavaScript toPrecision() Method

https://www.w3schools.com/jsref/jsref_toprecision.asp

ใช้จัดรูปแบบตัวเลข ให้มีความยาวตามที่กำหนด

โครงสร้างการใช้งาน

```
number.toPrecision(x)
```

พารามิเตอร์

Parameter	Description
x	(Optional) จำนวนหลักของตัวเลข ถ้าเว้นว่างไว้จะคืนค่าเป็นเลขตั้งเดิม ที่ไม่มีการจัดรูปแบบ

ตัวอย่าง

```
var num = 13.3714 ;
var a = num.toPrecision() ;           // 13.3714
var b = num.toPrecision(2) ;           // 13
var c = num.toPrecision(3) ;           // 13.4
var d = num.toPrecision(10) ;          // 13.37140000

var num2 = 0.001658853 ;
var e = num2.toPrecision() ;           // 0.001658853
var f = num2.toPrecision(2) ;           // 0.0017
var g = num2.toPrecision(3) ;           // 0.00166
var h = num2.toPrecision(10) ;          // 0.001658853000
```

20.22. valueOf()

JavaScript Number valueOf() Method

https://www.w3schools.com/jsref/jsref_valueof_number.asp

คืนค่าตัวเลขดั้งเดิมกลับมา ใช้กับข้อมูลเวลาจะเห็นชัดเจน

```
var num = 15 ;
var n = num.valueOf() ; // 15

var str = "Hello World!" ;
var res = str.valueOf() ; // Hello World!

// *****
var date = new Date("Tue Feb 11 2020 00:00:00 GMT+0700 (Indochina Time)")
var res = date.valueOf() ; // 1581354000000
```


บทที่ 21

Script Properties



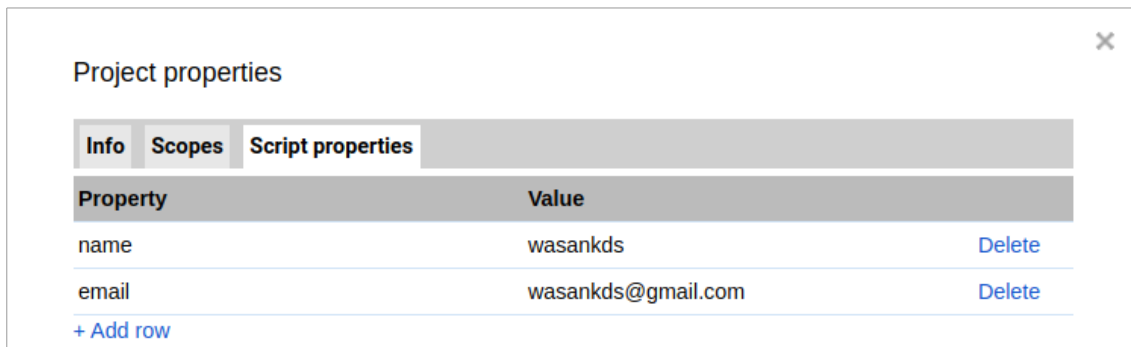
21.1. Script Properties คืออะไร ?

Script Properties ซึ่งเป็นคู่ของ Key และ Value มีลักษณะเหมือนกับตัวแปรวัตถุ เราสามารถสร้างคู่ดังกล่าว เก็บไว้ในโปรเจ็ค Apps Script ของเราได้ การใช้งานคล้ายกับการประกาศตัวแปร Global ไว้ในไฟล์ Code.gs แต่ Script Properties เป็นการสร้างฝังไว้ในโปรเจ็คเลย

ตัวอย่างการใช้งาน Script Properties เช่น รันสคริปต์เพื่อตั้งชื่อ Named Range แล้วเก็บชื่อ Named Range ไว้ใน Script Properties เพื่อให้สคริปต์อื่นๆนำไปใช้ต่อ เป็นต้น

21.2. การสร้าง Script Properties แบบกรอกค่าเอง

การสร้าง Script Properties ที่โปรเจ็ค Apps Script ให้ไปที่เมนู [File → Project Properties](#) จากนั้นดูแท็บ [Script properties](#) กรอกค่าเป็นคู่ของ Key และ Value ตัวอย่างตามภาพ เช่น name:wasankds เป็นต้น



21.3. การจับ Script Properties

คลาส `PropertiesService`

<https://developers.google.com/apps-script/reference/properties/properties-service>

`getScriptProperties()` - เมธอดในคลาส `PropertiesService`

<https://developers.google.com/apps-script/reference/properties/properties-service#getscriptproperties>

จับคุณสมบัติ ที่ทุกยูสเซอร์สามารถเข้าถึงได้ และอยู่ในโปรเจ็คนี้เท่านั้น

คลาส `Properties`

<https://developers.google.com/apps-script/reference/properties/properties>

`getProperties()` - เมธอดในคลาส `Properties`

<https://developers.google.com/apps-script/reference/properties/properties#getproperties>

ใช้จับ Script Properties ทั้งหมดในโปรเจ็ค โดยจะคืนค่ากลับมาเป็น **Object**

`getProperty(key)` - เมธอดในคลาส `Properties`

<https://developers.google.com/apps-script/reference/properties/properties#getpropertykey>

ใช้จับ Value ของ Key ที่ระบุ หากไม่มี Key ที่ระบุจะคืนค่า Null หากมีจะคืนค่าเป็น Value ในรูปแบบ

String

`getKeys()` - เมธอดในคลาส `Properties`

<https://developers.google.com/apps-script/reference/properties/properties#getkeys>

ใช้จับ Keys ทั้งหมดในคลังของ Script Properties

โค้ดต่อไปนี้จะจับ Script Properties ของโปรเจกต์มา Logs ดู

```
// จับเฉพาะ Keys โดยเก็บอยู่ในอาร์เรย์ ...
//
var KEYS = PropertiesService.getScriptProperties().getKeys() ;

// จับมาหมดทั้งยวง ทั้ง Keys และ Values โดยเก็บอยู่ใน Object
//
var PROPS = PropertiesService.getScriptProperties().getProperties() ;

Logger.log(KEYS) ;           // ----- > ดูผลที่ Logs [ 01 ]
Logger.log(PROPS) ;         // ----- > ดูผลที่ Logs [ 02 ]

var keyName = PropertiesService.getScriptProperties().getProperty('name') ;
var keyEmail = PropertiesService.getScriptProperties().getProperty('email') ;

Logger.log('name : ' + keyName) ;           // ----- > ดูผลที่ Logs [ 03 ]
Logger.log('email : ' + keyEmail) ;         // ----- > ดูผลที่ Logs [ 04 ]
```

ผล

```
Logs
[ 01 ] [name, email]
[ 02 ] {name=wasankds, email=wasankds@gmail.com}

[ 03 ] name : wasankds
[ 04 ] email : wasankds@gmail.com
```

21.4. การสร้าง Script Properties โดยการเขียนสคริปต์

21.4.ก.) setProperty()

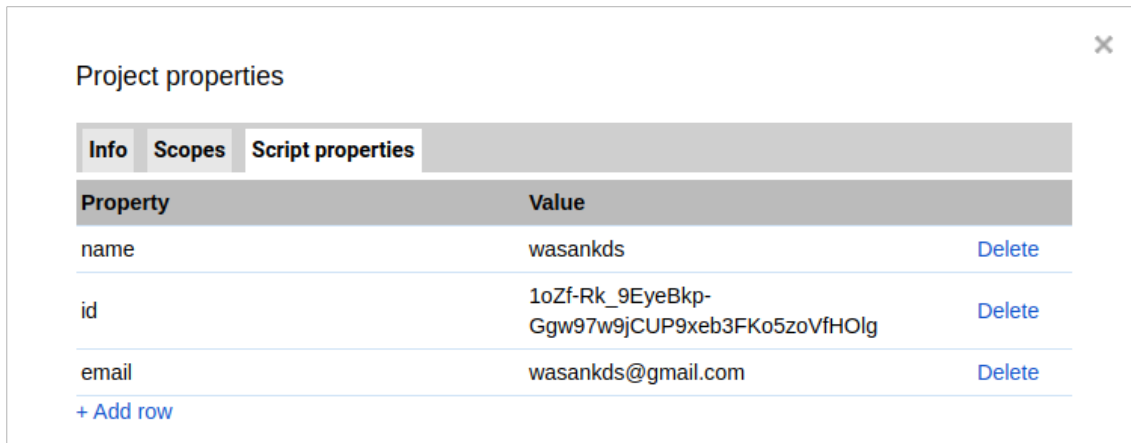
setProperty(key, value) - เมธอดในคลาส Properties

ใช้สร้างคู่ของ Key และ Value ลงในคลัง Script Properties ของโปรเจกต์

ตัวอย่าง - โค้ดต่อไปนี้จะจับ id ของไฟล์ Google Sheets ไปสร้างเป็น Script Property

```
const id = SpreadsheetApp.getActiveSpreadsheet().getId() ;
PropertiesService.getScriptProperties().setProperty('id', id) ;
```

ผลเมื่อรันฟังก์ชัน จะได้ Script Property ตามภาพ



Project properties		
Info	Scopes	Script properties
Property	Value	
name	wasankds	Delete
id	1oZf-Rk_9EyeBkp-Ggw97w9jCUP9xeb3FKo5zoVfHOlg	Delete
email	wasankds@gmail.com	Delete
+ Add row		

21.4.ข.) setProperties()

setProperties(properties) - เมธอดในคลาส Properties

<https://developers.google.com/apps-script/reference/properties/properties#setpropertiesproperties>

setProperties(properties, deleteAllOthers) - เมธอดในคลาส Properties

<https://developers.google.com/apps-script/reference/properties/properties#setpropertiesproperties,-deleteallothers>

setProperties() ใช้สร้างคู่ของ Key และ Value หลายๆคู่ ลงในคลัง Script Properties ของโปรเจ็ค

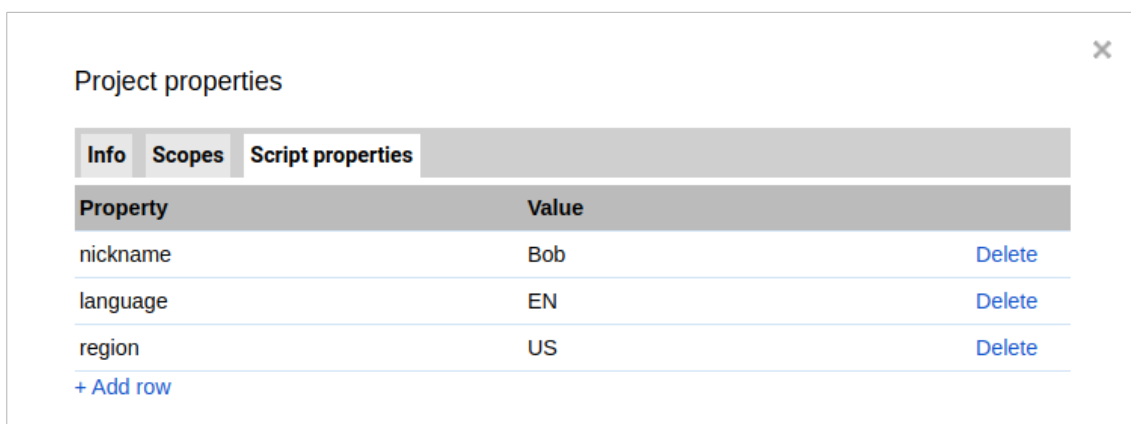
พารามิเตอร์ properties เป็นตัวแปรวัตถุที่ระบุคู่ของ Key-Value จำนวนตามต้องการ

พารามิเตอร์ deleteAllOthers เป็นบูลีนที่ใช้ระบุว่าลบ Key-Value ที่มีอยู่แล้วในคลัง Script Properties หรือไม่

ตัวอย่าง - สร้าง Script Properties 3 คู่ โดยลบทั้งหมดที่มีอยู่ในคลังทิ้งไปก่อน

```
var scriptProperties = PropertiesService.getScriptProperties() ;  
var newProperties = { nickname: 'Bob', region: 'US', language: 'EN'} ;  
scriptProperties.setProperties(newProperties,true) ;
```

ผล



Project properties		
Info	Scopes	Script properties
Property	Value	
nickname	Bob	Delete
language	EN	Delete
region	US	Delete
+ Add row		

21.5. deleteAllProperties() และ deleteProperty()

deleteProperty(key) - เมธอดในคลาส Properties

<https://developers.google.com/apps-script/reference/properties/properties#deletepropertykey>

ใช้ลบ Script Property ที่ระบุ

deleteAllProperties() - เมธอดในคลาส Properties

<https://developers.google.com/apps-script/reference/properties/properties#deleteallproperties>

ใช้ลบ Script Properties ทั้งหมดในคลังของโปรเจ็ค

ตัวอย่าง - ลบ Script Property ตัวเดียวแบบเจาะจง

```
var scriptProperties = PropertiesService.getScriptProperties () ;  
scriptProperties.deleteProperty('nickname') ;
```

ตัวอย่าง - ลบ Script Properties ทั้งหมดที่มีอยู่ในคลัง

```
var scriptProperties = PropertiesService.getScriptProperties () ;  
scriptProperties.deleteAllProperties() ;
```


บทที่ 22

Regular Expression



22.1. Regular Expression คืออะไร ?

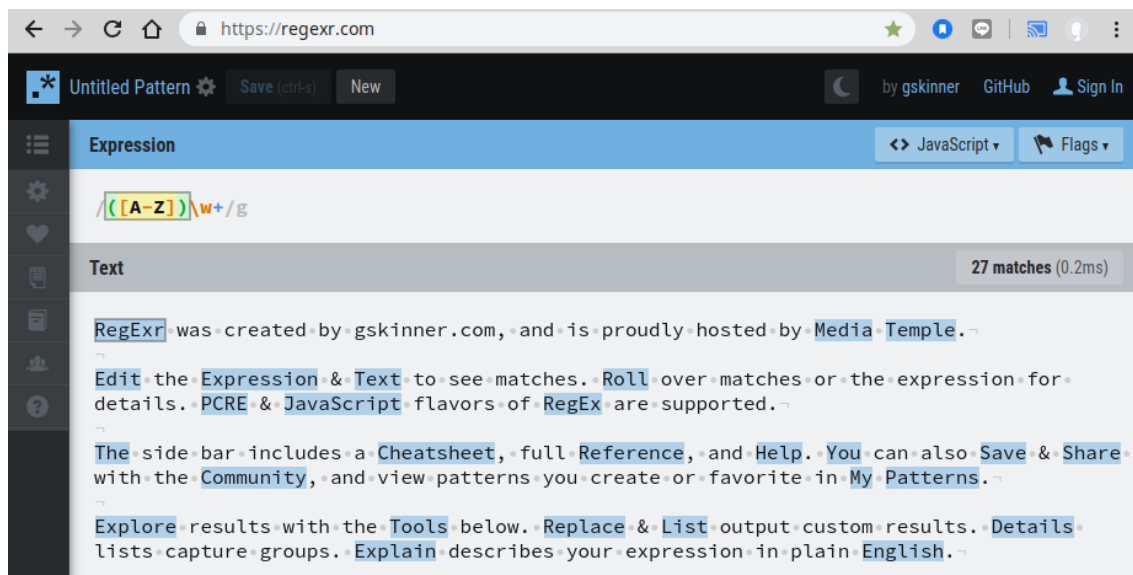
22.1.ก.) Regular Expression คืออะไร ?

Regular Expression เป็นอักขระที่สามารถนำมาผสมกัน เพื่อใช้เป็นแพทเทิร์น ในการค้นหา ตรวจสอบ ทดสอบ หรือ ประมวลผลใดๆ กับข้อความตั้งต้น

เครื่องมือต่างๆใน Google Sheets เช่น การค้นหา, ฟังก์ชันตระกูล RegEx Regular Expression ถูกใช้ในนั้นด้วย นอกจากนี้ การเขียนโปรแกรมที่เกี่ยวข้องกับข้อความ ไม่ว่าจะภาษาใดๆ ก็สามารถใช้งาน Regular Expression ได้ด้วย ซึ่ง Regular Expression ถือเป็นเครื่องมือที่ใช้งานกันอย่างกว้างขวาง และมีประโยชน์มาก

มีหลายเว็บไซต์ที่ทดสอบ Regular Expressions Syntax เช่น <https://regexr.com/>

ตามภาพ กรอบ Expression เป็น Regular Expression Syntax กรอบล่าง Text เป็นข้อความตั้งต้นที่จะใช้ทดสอบ Regular Expression



22.1.ข.) การใช้งาน Regular Expression ใน Apps Script

Regular Expression จัดเป็นวัตถุอีกชนิดหนึ่ง เวลาใช้งานจะเขียนไว้ในเครื่องหมายสแลช 2 อัน **/(RE)/** เช่น **/ABC[0-9]{5}/g** (ขึ้นต้นด้วย ABC ต่อด้วยตัวเลขอะไรก็ได้ 5 ตัว) เป็นต้น Regular Expression มักจะมี **Expression flags** ต่อท้ายเช่น **/(RE)/g** ซึ่งจะเป็นตัวที่บอกว่า Regular Expression จะถูกแปลงอย่างไร **g=global** จะแมท Regular Expression กับข้อความย่อย ภายในของข้อความตั้งต้นทั้งหมด

ตัวอย่าง รูปแบบการเขียน Regular Expression ใน Apps Script

```
var regPat = /Hello.+/g ; // ประกาศตัวแปรที่เก็บ Regular Expression
var regPatInText = ".*" + "wasan" + ".*"
var regPatNew = new RegExp(regPatInText,"g") ; // สร้างวัตถุ RegExp ใหม่จากข้อความ
```


22.2. Regular Expression Syntax ที่สำคัญ

อักขระที่ใช้เป็น **Regular Expressions** มีเยอะมาก และสามารถนำมาผสมเพื่อใช้งานได้หลากหลาย ในข้อนี้ ขอยกตัวอย่างบางตัวที่สำคัญๆ หรือใช้บ่อยๆ

JavaScript Regular Expressions

https://www.w3schools.com/js/js_regexp.asp

JavaScript RegExp Reference

https://www.w3schools.com/jsref/jsref_obj_regexp.asp

การค้นหาและแทนที่ใน Google Apps

https://support.google.com/docs/answer/62754?p=spreadsheets_find_replace&visit_id=636910790117519891-447857832&rd=1

Regular Expressions Syntax

<https://github.com/google/re2/blob/master/doc/syntax.txt>

Regular Expressions (Java Script)

https://developer.mozilla.org/th/docs/Web/JavaScript/Guide/Regular_Expressions

Google Sheets - RegEx REGEXTRACT, Functions Extract, Replace, Match Tutorial - Part 1

<https://www.youtube.com/watch?v=w5B43VsJqOs>

Google Sheets - Find and Replace with Functions or Without SUBSTITUTE, RegEx, Wildcards Tutorial

<https://www.youtube.com/watch?v=OVzXpLJRcQk&t=544s>

22.2.ก.) Regular Expression Syntax ที่สำคัญ

xy อักขร xy ผลก็คือ xy

. (Any char) อักขรอะไรก็ได้ในตำแหน่งที่ระบุ เช่น **d**. ผล do, dog, dg, ads

***** (zero or more) อยู่หลังตัวอักขรตัวไหน หมายถึง ตัวนั้นกี่ตัวก็ได้

เช่น **do*g** ผล dog, dg, dooog (dOg ไม่ได้)

+ (one or more) อยู่หลังตัวอักขรตัวไหน หมายถึง ตัวอักขรนั้นอย่างน้อย 1 ตัว

(คล้าย * แต่ต้องมีอย่างน้อย 1 ตัว) เช่น **do+g** ผล dog, dooog (dg, dOg ไม่ได้)

? (optional) อยู่หลังตัวอักขรตัวไหน หมายถึง มีหรือไม่มี อักขรตัวนั้นก็ได้

เช่น **do?g** (มีหรือไม่มี o ก็ได้) ผล dog, dg (dOg, doug ไม่ได้)

^ (start with) อยู่เริ่มต้น Regular Expression หมายถึง ประโยคเริ่มต้นด้วย

(เฉพาะใน Google Sheet เท่านั้น) เช่น **^[dh]og** ผล dog, hog (A dog, his hog ไม่ได้)

\$ (end with) อยู่จุดสุดท้ายของ Regular Expression หมายถึง ประโยคลงท้ายด้วย

เช่น **[dh]og\$** ผล dog, hog, hot dog (dogs, hogs, doggy ไม่ได้)

**** (escape) อยู่ก่อนอักขรหมายถึง อักขรตัวนั้น มักใช้กับอักขรพิเศษ เช่น \.

[A,B]	อยู่หลังตัวอักษรตัวไหน หมายถึง ซ้ำอักษรตัวนั้น A ถึง B ครั้ง เช่น d(o{1,2})g ผล dog, doog (dg, dOg, doog ไม่ได้) [0-9]{1,} - ตัวเลขยาวติดกันกี่ตัวก็ได้ เช่น 12, 123, 4567
[x], [xa], [xa5]	(match) ตัวใดตัวหนึ่งที่อยู่ใน [] เช่น d[ou]g ผล dog, dug (dg, dOg, doog, doug ไม่ได้)
[a-z]	ตัวใดตัวหนึ่งที่อยู่ระหว่าง ช่วงที่ระบุใน [] เช่น d[o-u]g ผล dog, dug, dpg, drg (dg, dOg, dag ไม่ได้)
[^a-fDEF]	ต้องไม่มีตัวอักษรตัวใดตัวหนึ่งที่ระบุใน [] เช่น d[^aeu]g ผล dog, dOg, dig, d\$g (dg, dag, deg, dug ไม่ได้)
 	(or) หรือ
()	จับกลุ่มที่อยู่ใน ()
(?:re) = ?:	อยู่ใน () ก็คือ ไม่เอากลุ่มนี้
(re)? = ()?	อยู่หลังกลุ่ม ก็คือ Optional มีก็เอา ไม่มีก็ไม่เอา

Escape sequences (\)

\d	digits (== [0-9]) ตัวเลข
\D	not digits (== [^0-9]) ไม่ใช่ตัวเลข
\s	whitespace (== [\t\n\r]) สเปซ เช่น d\s g ผล <u>d g, d[tab]g</u> (dg, dog ไม่ได้)
\S	not whitespace (== [^\t\n\r]) ไม่ใช่สเปซ
\w	word characters (== [0-9A-Za-z_]) Alphanumeric หรือ ตัวอักษรและตัวเลข
\W	not word characters (== [^0-9A-Za-z_])

22.2.ข.) การผสม Regular Expression ที่น่าสนใจ

aaa

ไม่มี Regular Expression ค้นหา **aaa**

..

อักษร 2 ตัว

...

อักษร 3 ตัว ผลเช่น **Abc, Axx**

\.

เครื่องหมายจุด ถ้าใช้โดดๆหมายถึงอักขระอะไรก็ได้ 1 ตัว แต่ถ้าใช้กับ \ หมายถึง อักขระที่เป็น

เครื่องหมายจุด นอกจากนี้ก็ยังมี \ (, \) , \ ? , \ + , \ \ , \ ^ , \ \$, \ [, \] , \ | , ซึ่งเหล่านี้เป็น Special characters

A+

A ก็ได้ตัวก็ได้ ผลเช่น A, AA, AAA

.+

อักขระอะไรก็ได้ ที่ตัวก็ได้ แต่อย่างน้อย 1 ตัว ผลก็คือ ข้อความทั้งหมดแม้มีช่องว่างก็ตาม

*.

อักขระอะไรก็ได้ ที่ตัวก็ได้ 0 ตัวก็ได้ ผลก็คือ ข้อความทั้งหมดแม้มีช่องว่างก็ตาม

A.+

ข้อความที่ขึ้นต้นด้วย A ผลเช่น Amily Checking, Alaska Airline

A.+B.+

ข้อความที่ขึ้นต้นด้วย A หรือ B ผลเช่น Alpha Channel, Ben Tema,

(A.+)(B.+)

ข้อความที่ขึ้นต้นด้วย A หรือ B ผลที่ได้จะแยกคอลัมน์

ผลเช่น	คอลัมน์ที่ 1 (กลุ่มที่1)	คอลัมน์ที่ 2 (กลุ่มที่2)
	(ว่าง)	Ben Tema
	Alpha Channal	(ว่าง)

((A.+)(B.+))

ข้อความที่ขึ้นต้นด้วย A หรือ B ผลที่ได้จะแยกคอลัมน์ตามจำนวนกลุ่ม มี 3 กลุ่ม ตามที่ระบุ ก็คือ (())()

ผลเช่น	คอลัมน์ที่ 1 (กลุ่มที่1)	คอลัมน์ที่ 2 (กลุ่มที่2)	คอลัมน์ที่ 3 (กลุ่มที่2)
	Ben Tema	(ว่าง)	Ben Tema
	Alpha Channal	Alpha Channal	(ว่าง)

((?:A.+)(?:B.+))

เหมือนก่อนหน้า แต่ไม่เอากลุ่มที่มี ? อยู่ข้างหน้า ฉะนั้นจะเหลือเฉพาะกลุ่ม ข้างนอกหรือกลุ่มที่ 1 อย่าง

เดียว

ผลเช่น	คอลัมน์ที่ 1 (กลุ่มที่1)
	Ben Tema
	Alpha Channal

[A-Z]+

A-Z ตัวได้ก็ได้ ติดกันก็ได้ ตัวก็ได้ ผลเช่น A, F, AA, BA, CFD, ABCD (ตัวเล็กไม่ได้)

[a-zA-Z._-]+

อักขระ a-z, A-Z, . , _ , - ก็ได้ติดกันก็ได้

เช่น ถ้าใช้กับอีเมล จะตัดชื่อจากอีเมล ผลเช่น wasan, kds.wasan, suwaj, wannee, Family

\w

อักขร Alphanumeric 1 ตัว เทียบเท่า [0-9A-Za-z_] ผลเช่น A, B, a, 5, _

\w+

อักขร Alphanumeric ก็ได้ตัวก็ได้ที่ติดกัน ผลก็คือ ข้อความ 1 คำ (ตัดที่ช่องว่าง)

`\w+$`

อักษร Alphanumeric ก็ได้ที่ติดกัน ทำจากข้างหลัง ผลก็คือ ข้อความ 1 คำด้านหลัง

`\w+\s\w`

อักษร Alphanumeric 2 คำที่ติดกัน (เว้นด้วยช่องว่าง)

`@[A-Za-z0-9-]+\.[A-Za-z]+`

ถ้าใช้กับอีเมล จะตัดโดเมนข้างหลังออกมา ใช้ได้กับแบบจุดเดียว

ผลเช่น `@gmail.com`, `@snp.gmail` (มี . 2 อันหลัง @ มาแค่ตัวแรกตัวเดียว)

`@[A-Za-z0-9-]+\.[A-Za-z]+\.[A-Za-z]`

ถ้าใช้กับอีเมล จะตัดโดเมนข้างหลังออกมา ใช้ได้กับแบบ 2 จุด

ผลเช่น `@snp.gmail.com` (`@gmail.com` ไม่ได้)

`(@[A-Za-z]+\.[A-Za-z]+\.[A-Za-z]+)(@[A-Za-z]+\.[A-Za-z]+)`

Regex ชุดแรก ตัดโดเมนข้างหลังออกมา ใช้ได้กับแบบ 2 จุด Regex ชุดหลัง ตัดโดเมนข้างหลังออกมา ใช้ได้กับแบบจุดเดียว ลำดับมีความสำคัญ ถ้าเจอ Regex ชุดแรกแล้ว จะไม่ทำชุดหลัง จะทำ Regex ชุดหลังเมื่อตัวแรกไม่เจอ

ผลเช่น	คอลัมน์ที่ 1 (กลุ่มที่1)	คอลัมน์ที่ 2 (กลุ่มที่2)
	(ว่าง)	<code>@gmail.com</code>
	<code>@snp.gmail.com</code>	(ว่าง)
	(ว่าง)	<code>@snpfood.com</code>
	<code>@yahoo.thai.com</code>	(ว่าง)

`(?:@[A-Za-z]+\.[A-Za-z]+\.[A-Za-z]+)|(?:@[A-Za-z]+\.[A-Za-z]+)`

หรือ

`@[A-Za-z]+\.[A-Za-z]+\.[A-Za-z]+|@[A-Za-z]+\.[A-Za-z]+`

แบบไม่ใช้วงเล็บ

เหมือนข้างบน แต่เอามาอันใดอันหนึ่ง เฉพาะอันที่เจอ

ผลเช่น	คอลัมน์ที่ 1
	<code>@gmail.com</code>
	<code>@snp.gmail.com</code>
	<code>@snpfood.com</code>
	<code>@yahoo.thai.com</code>

`@[A-Za-z]+(?:\.[A-Za-z]+\.[A-Za-z]+)?(?:\.[A-Za-z]+)?`

เหมือนข้างบน แต่ `((?:RE)(?:RE))` มี 3 กลุ่ม ไม่เอากลุ่มที่มี `?:` จึงเหลือกลุ่มนอกสุดอันเดียว ผลที่ได้ก็คือ

เฉพาะ `.com` หรือ `.co.th` (อยู่หลัง 1 หรือ 2 จุด)

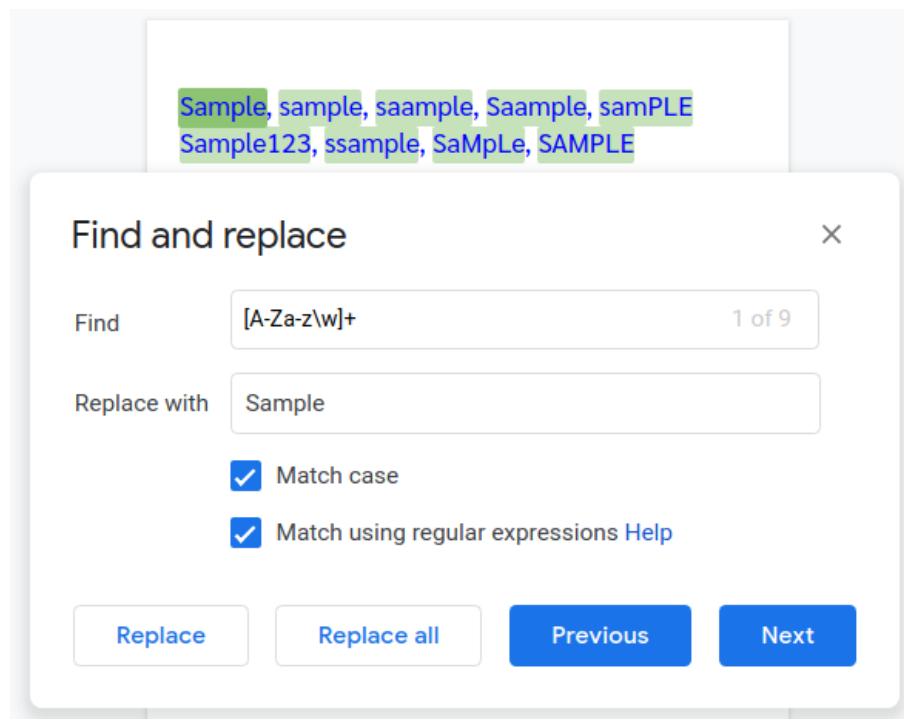
ผลเช่น	คอลัมน์ที่ 1
	<code>.com</code>
	<code>.gmail.com</code>
	<code>.com</code>
	<code>.thai.com</code>

22.3. การค้นหาและแทนที่โดยใช้ Regular Expression

ข้อนี้ไม่เกี่ยวกับ Google Apps Script แต่เกี่ยวกับ Google Apps ที่สำคัญๆ เช่น Google Sheets, Google Docs

เราสามารถใช้เครื่องมือ Find and replace ในการค้นหาและแทนที่ โดยใช้ Regular Express ได้ด้วย

ยกตัวอย่างใน Google Docs ไปที่เมนู Edit → Find and replace (**<Ctrl><H>**) จะปรากฏหน้าต่าง Find and replace ตามภาพ จากนั้นให้ ดับเบิลคลิกที่ Match using regular expressions



22.4. RegExp()

JavaScript RegExp Reference

https://www.w3schools.com/jsref/jsref_obj_regexp.asp

เราไม่สามารถ ใส่ตัวแปรรวมกับ Regular expression syntax ได้ วิธีแก้ก็คือ สร้าง Regular expression syntax ตัวใหม่จากวัตถุ **RegExp**

ตัวอย่าง

```
var textToSearch = "Hello world" ;
var regPatText = 'w+ .*' ;

// สร้าง RegExp ตัวใหม่ จากตัวแปร
var regPatNew = new RegExp(regPatText, "gi") ;

var posFound = textToSearch.search(regPat) ;
Logger.log(posFound) ; // 6
```

22.5. exec()

JavaScript exec() Method

https://www.w3schools.com/jsref/jsref_regexp_exec.asp

RegExp.prototype.exec()

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/RegExp/exec

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/RegExp

exec(str) เป็น Built-in ฟังก์ชัน ใช้จัดการเมธอดของ RegExp กับ String (ข้อความ) โดยจะคืนค่า

String[]-ก้อนอาเรย์ของ String ที่แมท หากไม่เจอค่าที่แมท จะคืนค่า null

โครงสร้าง

```
RegExpObject.exec(string)
```

ตัวอย่างที่ 1

```
var str = "The best things in life are free" ;
var patt = new RegExp("e") ;
var res = patt.exec(str) ;
Logger.log(res) ;           // [e]
```

ตัวอย่างที่ 2

```
// The string :
var str = "Hello world!" ;

// Look for "Hello"
var patt = /Hello/g ;
var result = patt.exec(str) ;
Logger.log(result) ;           // [Hello]

// Look for "W3Schools"
var patt2 = /W3Schools/g ;
var result2 = patt2.exec(str) ;
Logger.log(result2) ;          // null
```

ตัวอย่างที่ 3 - โค้ดต่อไปนี้จะดูชนิดของ Regular Expression

```
var regex1 = /\w+/ ;
var regex2 = new RegExp('\w+') ; // RegExp ใช้สร้าง regular expression

Logger.log(typeof regex1) ;      // ดูผลที่ Logs -----> [1]
Logger.log(regex1) ;             // ดูผลที่ Logs -----> [2]

Logger.log(typeof regex2) ;      // ดูผลที่ Logs -----> [3]
Logger.log(regex2) ;             // ดูผลที่ Logs -----> [4]
```

ผล

Logs

```
[01] object
[02] ^w+/
[03] object
[04] ^w+/
```

ตัวอย่างที่ 4 - โค้ดต่อไปนี้จะ Regular Expression เป็นแพทเทิร์นในการค้นหาข้อความ

```
function myFunction() {  
    var text = 'wasankds@gmail.com';    // ข้อความเริ่มต้น  
    var regex = /[@].+;/                // RE  
    var match = regex.exec(text);        // ใช้ RE เป็นแพทเทิร์นในการค้นหาข้อความ  
    Logger.log(typeof match);  
    Logger.log(match);  
}
```

ผล

Logs

```
[01] object
[02] [@gmail.com]
```

22.6. test()

RegExp.prototype.test()

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/RegExp/test

เมธอด `test()` ใช้ทดสอบ Regular Expression กับ String ที่ระบุโดยจะคืนค่า `true` หรือ `false`

โครงสร้างการใช้งาน

```
regexObj.test(str)
```

พารามิเตอร์ **str** ก็คือ String ที่ใช้ทดสอบ Regular Expression

ตัวอย่างที่ 1

```
const str = 'hello world!' ;
const result = /^hello/.test(str) ;
console.log(result) ; // true – จับ /^hello/ ใน String ได้
```

ตัวอย่างที่ 2

```
var regex = /^[\\w-\\.]+@[\\w-]+\\.+[\\w-]{2,6})?$/ ; // ใช้เป็น /m
return regex.test(email) ; // ถ้าเป็น email คืนค่า true
```