

เรียนรู้การสร้างโปรเจ็ค

Google Apps Script

เล่ม 1

12 พ.ย. 2019

เรียบเรียงโดย  
วสันต์ คุณติลกเศวต



# เรียนรู้การสร้างโปรเจ็ค Google Apps Script

เรียบเรียงโดย  
วสันต์ คุณดิลกเศวต

wasankds@gmail.com  
Line ID : wasankds  
08-1459-8343  
[www.poeclub.org](http://www.poeclub.org)

# สารบัญ

คำนำ.....	6
-----------	---

## บทที่ 1

โปรเจ็ค 01 สร้าง Mail Merge จาก Sheets ไป Docs.....	9
---	---

1.1. ที่มาของโปรเจ็ค 01 .....	10
1.2. ภาพรวมโปรเจ็ค 01 .....	10
1.2.ก.) ข้อมูลในไฟล์ Template หรือไฟล์ต้นทาง (10)	
1.3. พัฒนาการที่ 1 : ก๊อปปี้ย่อหน้าจาก Doc1 ไป Doc2 .....	11
1.4. พัฒนาการที่ 2 : ก๊อปปี้ย่อหน้าจาก Doc1 ไป Doc2 + แทนที่ข้อความ .....	12
1.4.ก.) ค้นหา 1 ข้อความใน 1 ย่อหน้า (ทดสอบดูก่อน) (12)	
1.4.ข.) ค้นหา 1 ข้อความ ในทุกย่อหน้า (12)	
1.4.ค.) ค้นหาหลายข้อความ ในทุกย่อหน้า (13)	
1.4.ง.) ค้นหาหลายข้อความ ในทุกย่อหน้า – สร้างเป็นฟังก์ชัน (2 พารามิเตอร์) (13)	
1.4.จ.) เปลี่ยนข้อความคงที่เป็นพารามิเตอร์ (3+2 พารามิเตอร์) (14)	
1.5. พัฒนาการที่ 3 : เตรียมข้อมูลจากไฟล์ Google Sheet .....	14
1.5.ก.) ทดลองเอาข้อมูลจาก Google Sheet มา Log ดู (14)	
1.5.ข.) เอาข้อมูลจากชีท ไปใส่ในเอกสาร (15)	
1.6. ย่อหน้าที่เป็น List .....	15
1.6.ก.) ตรวจสอบชนิดย่อหน้า (16)	
1.6.ข.) ใช้ IF ตามชนิดของย่อหน้า (17)	

## บทที่ 2

โปรเจ็ค 02 ส่งเมลโดยใช้ MailApp.....	19
--------------------------------------	----

2.1. ที่มาของโปรเจ็ค 02 .....	20
2.2. ภาพรวมโปรเจ็ค 02 .....	20
2.3. พัฒนาการที่ 1 - จับค่าในคอลัมน์ออกมา Log ดู .....	21
2.4. พัฒนาการที่ 2 - ใช้ MailApp ส่งเมล .....	21
2.5. พัฒนาการที่ 3 - จับเนื้อหาใน Template ออกมา Log ดู .....	22
2.6. พัฒนาการที่ 4 - แทนที่ข้อความใน Template .....	23
2.7. พัฒนาการที่ 5 - ส่ง Mail Merge email .....	24
2.8. พัฒนาการที่ 6 - ส่งตาม Quota limit .....	25
2.9. พัฒนาการที่ 7 - สร้างปุ่ม .....	26

## บทที่ 3

### โปรเจกต์ 03 Mail Merge จาก Sheets โดยใช้ HTML Template.....27

3.1. ที่มา .....	28
3.2. พัฒนาการที่ 1 : เตรียมพร้อมไฟล์ Google Sheets .....	28
3.2.ก.) สร้างไฟล์ Google Sheet       (28)	
3.2.ข.) สร้างไฟล์ Google Apps Script (28)	
3.3. พัฒนาการที่ 2 : สร้าง Template ไฟล์ HTML .....	28
3.4. พัฒนาการที่ 3 : แทรกสคริปต์เตรียมเรียกใช้ฟังก์ชัน .....	29
3.5. พัฒนาการที่ 4 : ดึงข้อมูลจาก Google Sheets .....	30
3.5.ก.) ดึงข้อมูลจาก Google Sheets มา Log ดู       (30)	
3.5.ข.) ใช้ forEach เข้าไปดึงอีเมลทุกบรรทัดมา Log ดู (30)	
3.6. พัฒนาการที่ 5 : วนหลูปส่งเมลโดยใช้ Template จากไฟล์ HTML .....	31
3.7. พัฒนาการที่ 6 : สร้างปุ่มเพื่อรันสคริปต์ .....	33
3.8. พัฒนาการที่ 7 : ฝาก Subject และ Name ไว้ในชีทที่ 2 .....	34
3.9. พัฒนาการที่ 8 : ส่งใครบ้าง .....	35
3.10. พัฒนาการที่ 9 : เครื่องมือช่วยสร้างโค้ด HTML .....	37

## บทที่ 4

### โปรเจกต์ 04 ส่งอีเมลขอบคุณ เมื่อกรอกแบบฟอร์ม.....39

4.1. ที่มา .....	40
4.2. เตรียมพร้อม .....	40
4.2.ก.) สร้าง From       (40)	
4.2.ข.) สร้างโปรเจกต์ Google Apps Script สำหรับ Form       (41)	
4.2.ค.) สร้างไฟล์ email.html ในโปรเจกต์ Google Apps Script(42)	
4.3. พัฒนาการที่ 1 : ทดสอบส่งอีเมลจาก Tempalte .....	42
4.4. พัฒนาการที่ 2 : รับค่าจาก Form submit .....	44
4.5. พัฒนาการที่ 3 : สร้าง Trigger เพื่อรันฟังก์ชันตาม Event .....	45
4.6. ทดสอบกรอกแบบฟอร์ม .....	47

## บทที่ 5

โปรเจ็ค 05 สร้างข้อสไลให้ Dropdown ใน Form จากข้อมูล ใน Google Sheet....49

5.1. ที่มา .....	50
5.2. เตรียมพร้อม .....	50
5.3. พัฒนาการที่ 1 : หา ID ของ Item .....	51
5.4. พัฒนาการที่ 2 : จับคุณสมบัติต่างๆของ Item มา Log ดู .....	51
5.5. พัฒนาการที่ 3 : ใส่ข้อสไลให้กับคำถาม .....	52
5.6. พัฒนาการที่ 4 : แยกฟังก์ชัน .....	52
5.6.ก.) แยกฟังก์ชันครั้งที่ 1      (52)	
5.6.ข.) แยกฟังก์ชันครั้งที่ 2      (54)	
5.7. แก้ไข Form และ Google Sheet .....	55
5.8. พัฒนาการที่ 5 : จับข้อมูลใน Google Sheet ไปสร้าง Item ใน Form .....	55
5.8.ก.) จับแถวคำถามมา Log ดู (55)	
5.8.ข.) จับแถวคำถามมา Log ดู - ดูข้อมูลเพิ่ม (56)	
5.8.ค.) จับข้อสไลมา Log ดู      (56)	
5.8.ง.) จับข้อสไลมา Log ดู      (57)	
5.8.จ.) จับข้อมูลใน Google Sheet ไปสร้าง Item ใน Form   (57)	
5.8.ฉ.) กำจัดค่าว่างในอาร์เรย์ข้อสไล      (58)	
5.9. พัฒนาการที่ 6 : ป้องกัน Error .....	58

## บทที่ 6

โปรเจ็ค 06 สร้าง Dropdown แบบ 3 ระดับ.....61

6.1. ที่มา .....	62
6.2. พัฒนาการที่ 1 : ทดสอบสร้าง Validation .....	62
6.2.ก.) ทดสอบสร้าง Validation      (62)	
6.2.ข.) สร้างเป็นฟังก์ชันสำหรับสร้าง Validation ให้กับเซลล์   (64)	
6.3. พัฒนาการที่ 2 : สร้างฟังก์ชัน onEdit .....	64
6.3.ก.) แก้ไขที่เซลล์ไหนก็ได้ มีผลต่อเซลล์ B2      (64)	
6.3.ข.) แก้ไขที่เซลล์ไหนก็ได้ มีผลต่อเซลล์ที่อยู่ติดกัน      (66)	
6.3.ค.) แก้ไขที่เซลล์ในคอลัมน์ A เท่านั้น จึงจับค่าไปใช้   (67)	
6.3.ง.) มีผลเฉพาะชีทที่ระบุเท่านั้น      (67)	
6.4. พัฒนาการที่ 3 : จับข้อมูลในอีกชีทมาทำเป็น List ให้กับ Dropdown .....	67
6.4.ก.) จับข้อมูลในอีกชีทมาทำเป็น List ให้กับ Dropdown      (67)	
6.4.ข.) เพิ่มค่าคงที่ firstLevelColumn และ secondLevelColumn   (69)	
6.4.ค.) แยกฟังก์ชัน      (70)	
6.5. พัฒนาการที่ 4 : สร้าง Validation ให้กับระดับที่ 2 .....	71

คำนำ



ผู้เขียน เขียนหนังสือเล่มนี้ “เรียนรู้การสร้างโปรเจกต์ Google Apps Script” จุดประสงค์ดั้งเดิม ก็คือ เก็บไว้อ่านเอง

เมื่อผู้เขียนศึกษาเรื่องอะไร ก็จะไปเรียนรู้จากสื่อออนไลน์ในอินเทอร์เน็ต ทั้งคอร์สออนไลน์ วิดีโอ หรือ เอกสาร ทั้งในแบบฟรีและเสียเงิน

ในยุคปัจจุบันเราต้องเรียนรู้อะไรให้เร็ว โดยเฉพาะเรื่องของ IT ผู้เขียนจึงตั้งใจจะดูวิดีโอเพียงรอบเดียว จึงดูไปพิมพ์สรุปไปด้วย เวลาจำอะไรไม่ได้ มาดูจากที่พิมพ์สรุปไว้ง่ายกว่าการไปย้อนดูจากวิดีโอ นอกจากนี้ ก็ยังนำมาทบทวนได้ง่าย ในอนาคตสามารถเพิ่มเติมเสริมแต่งเนื้อหาได้เรื่อยๆ ด้วย

ผู้เขียน มีหนังสืออีกเล่มหนึ่งที่ต้องใช้คู่กับหนังสือเล่มนี้ ก็คือ หลักการเขียนโปรแกรมด้วย Google Apps Script เป็นหนังสือเกี่ยวกับพื้นฐานและหลักการสำคัญในการเขียนโปรแกรมด้วย Google Apps Script เมื่อผู้เขียนเรียนรู้จากโปรเจกต์ต่างๆ ในหนังสือเล่มนี้ ซึ่งก็มีที่มาจากของคนอื่น อะไรน่าสนใจ ไม่เข้าใจ ก็ไปหาข้อมูลเพิ่มเติมเป็นเรื่อยๆ แล้วจัดหมวดหมู่ใหม่แยกไปเขียนไว้ในหนังสือเล่มดังกล่าว ฉะนั้นจึงต้องใช้คู่กัน

หนังสือเล่มนี้ ความตั้งใจดั้งเดิมของผู้เขียน ก็คือ ตั้งใจเก็บไว้อ่านเองคนเดียว เหตุเพราะเนื้อหาเอามาจากของคนอื่นเป็นส่วนใหญ่ ผู้เขียนเขียนเพิ่มไปไม่มาก

อย่างไรก็ดี อดสำหรับพิมพ์ไว้เป็นหนังสือแล้ว จะเก็บไว้อ่านคนเดียวก็รู้สึกเสียดาย ผู้เขียนจึงเอามาแบ่งปัน

เนื่องด้วย ผู้เขียนให้ความสำคัญกับประเด็นด้านลิขสิทธิ์มาก ฉะนั้นจึงขอแจ้งไว้ ณ ที่นี้ตั้งแต่ต้น ก็คือ

1. เนื้อหาในหนังสือ ผู้เขียนรวบรวมมาจากแหล่งต่างๆ ในอินเทอร์เน็ต ซึ่งจะพยายามให้มากที่สุด ที่จะบอกถึงหรือแหล่งที่มาในแต่ละหัวข้อ เพราะเขียนหนังสือเล่มนี้เขียนไว้นานแล้ว บางเรื่องลืมก๊อปปี้ลิงค์มาแปะไว้

2. ผู้เขียนเป็นเพียงผู้รวบรวมเนื้อหา เขียนเพิ่มเติมเพียงเล็กน้อย

3. หนังสือเล่มนี้แจกฟรี ผู้เขียนไม่มีรายได้จากหนังสือเล่มนี้

ขอแสดงความนับถือ

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



บทที่ 1  
โปรเจ็ค 01  
สร้าง Mail Merge  
จาก Sheets ไป Docs



## 1.1. ที่มาของโปรเจ็ค 01

Mail Merge Google Sheets to Google Docs - No Addons - Apps Script Tutorial

<https://www.youtube.com/watch?v=QNPPEB64QbI&list=PL0cbWZdynvACLB4ol4gHkuXYqyynMihRj&index=1>

## 1.2. ภาพรวมโปรเจ็ค 01

โปรเจ็คนี้ เป็นการสร้าง Mail Merge โดยเอาข้อมูลรายชื่อในไฟล์ Google Sheet ไปพิมพ์เป็น Mail Merge ลงในไฟล์ Google Docs โดยเราสามารถสร้างไฟล์ที่เป็น Template ในการสร้าง Mail Merge ได้

โปรเจ็คนี้ใช้ไฟล์ 3 ไฟล์ ดังต่อไปนี้

### 1.2.ก.) ข้อมูลในไฟล์ Template หรือไฟล์ต้นทาง

ไฟล์ docTemplate (Google Docs)

เป็นไฟล์ Template มีข้อมูลดังต่อไปนี้

Dear {first}

สวัสดีที่จะบอกว่า คุณ {first} {last} เบอร์โทรศัพท์ {phone} ได้รับรางวัล ...

ขอแสดงความนับถือ

วสันต์ คุณดิลกเสวต

ไฟล์ docFinal (Google Docs)

เป็นไฟล์ว่างๆ ที่คอยรับผลจากการพิมพ์ Mail Merge

ไฟล์ sheetData (Google Sheets)

เป็นแหล่งข้อมูลสำหรับทำ Mail Merge ซึ่งมีข้อมูลดังต่อไปนี้

	A	B	C	D	E
1	ชื่อ	นามสกุล	อีเมล	เบอร์โทรศัพท์	เมือง
2	ตัวเล็ก	คุณดิลกเสวต	tualek@gmail.com	091-399-6766	กรุงเทพฯ
3	นภาพร	ศิลาพันธ์	napa.por@gmail.com	095-768-0670	พิษณุโลก
4	ภูคัลป์	สวรรณดิน	poecclub@gmail.com	081-459-8343	เพชรบูรณ์
5					
6					

### 1.3. พัฒนาการที่ 1 : ก๊อปปีย่อหน้าจาก Doc1 ไป Doc2

โค้ดต่อไปนี้จะก๊อปปีย่อหน้าจาก ไฟล์ docTemplate ไปที่ ไฟล์ docFinal

```
function pDoc1_to_Doc2_R1() {  
  // ID ของไฟล์ Google Docs ที่เป็น Template  
  var docTemplatedID = "1Gc8MC7o4O68dVekknOPLGcmdcWI82wUjplH-x8HwXQc" ;  
  // ID ของไฟล์ Google Docs ที่รอรับผลจากการพิมพ์ MailMerge  
  var docFinalID = "1aqAv5bzhIvxWPheMpaQJu2vPTWZQQKd3qIsyGPlq-qM" ;  
  // ID ของไฟล์ Google Sheet ไฟล์ที่เก็บตารางข้อมูล  
  var wsID = "1IKcuYYWBZGgKvKLF3SNCvTM-OA7gyYjvTxzhEMmVjzE" ;  
  
  // เปิดไฟล์ Google Docs ทั้ง 2 แล้วเก็บไว้ในตัวแปร  
  
  var docTemplate = DocumentApp.openById(docTemplatedID) ;  
  var docFinal = DocumentApp.openById(docFinalID) ;  
  
  // จับย่อหน้าทั้งหมดในไฟล์ Template มาเก็บไว้ในตัวแปร  
  var docTemplateParagraphs = docTemplate.getBody().getParagraphs();  
  
  // Logger.log(docTemplateParagraphs); - log ดูตัวแปร จะพบว่าเป็นอาเรย์  
  
  docTemplateParagraphs.forEach(function(p){  
    // วนเข้าไปในสมาชิกอาเรย์แต่ละตัว เพื่อก๊อปปีย่อหน้าไปไว้ในอีกไฟล์  
    docFinal.getBody().appendParagraph(p.copy());  
  });  
  
  // Logger.log(docTemplateParagraphs); - log ดูย่อหน้าจะพบว่าเป็นอาเรย์  
}
```

เมื่อรันโค้ด ย่อหน้าทั้งหมดในไฟล์ Template จะถูกก๊อปปี้ไปที่ไฟล์ Final

#### หมายเหตุ 1

appendParagraph() ก็ตามความหมาย ก็คือ ต่อท้ายด้วยย่อหน้า เช่น

appendParagraph("ย่อหน้าที่ 1")

#### หมายเหตุ 2

ถ้าใช้เป็นบรรทัดด้านล่างจะปรากฏ Error “Element must be detached.” (detach = แยกออก)

```
docFinal.getBody().appendParagraph(p);
```

ย่อหน้าต้นทางต้องถูกแยกออกมาก่อน โดยการก๊อปปี้ จากนั้นจึงจะวางลงไปในไฟล์ปลายทาง ฉะนั้นแก้  
เป็น

```
docFinal.getBody().appendParagraph(p.copy());
```

## 1.4. พัฒนาการที่ 2 : ก๊อปปีย่อหน้าจาก Doc1 ไป Doc2 + แทนที่ข้อความ

จากไฟล์ Template ที่เป็นต้นทาง ย่อหน้าที่ 1 มีข้อความเช่น {name} เราจะค้นหาข้อความ {name} แล้วแทนที่ด้วยข้อความ เช่น “วสันต์” จากนั้นจึงก๊อปปี้ไปไว้ในไฟล์ Final ที่เป็นปลายทาง

### 1.4.ก.) ค้นหา 1 ข้อความใน 1 ย่อหน้า (ทดสอบดูก่อน)

ทดสอบ ก๊อปปีย่อหน้า ค้นหาและแทนที่ข้อความ จากนั้นวางไปยังไฟล์ปลายทาง เพียง 1 ย่อน้าก่อน

```
function pDoc1_to_Doc2_R2_1() {  
    var docTemplatedID = "1Gc8MC7o4O68dVekknOPLGCmdcWI82wUjplH-x8HwXQc" ;  
    var docFinalID = "1aqAv5bzhIvxWPheMpaQJu2vPTWZQQKd3qIsyGPlq-qM" ;  
    var wsID = "1IKcuYYWBZGgKvkLF3SNCvTM-OA7gyYjvTxzhEMmVjzE";  
  
    var docTemplate = DocumentApp.openById(docTemplatedID) ;  
    var docFinal = DocumentApp.openById(docFinalID) ;  
  
    // ไฟล์ต้นทาง  
    var docTemplateParagraphs_1= docTemplate.getBody().getParagraphs()[0] // จับย่อหน้าที่ 1  
        .copy() // Copy  
        .replaceText("{first}", "วสันต์") ; // ค้นหาและแทนที่  
  
    // ก๊อปปีย่อหน้าที่ค้นหาและแทนที่แล้ว ไปยังไฟล์ปลายทาง  
    docFinal.getBody().appendParagraph(docTemplateParagraphs_1.copy());  
}
```

### 1.4.ข.) ค้นหา 1 ข้อความ ในทุกย่อหน้า

เหมือนโค้ดในข้อก่อนหน้า คราวนี้จะทำกับทุกย่อหน้าในไฟล์ต้นทาง

```
function pDoc1_to_Doc2_R2_2() {  
    var docTemplatedID = "1Gc8MC7o4O68dVekknOPLGCmdcWI82wUjplH-x8HwXQc" ;  
    var docFinalID = "1aqAv5bzhIvxWPheMpaQJu2vPTWZQQKd3qIsyGPlq-qM" ;  
    var wsID = "1IKcuYYWBZGgKvkLF3SNCvTM-OA7gyYjvTxzhEMmVjzE";  
  
    var docTemplate = DocumentApp.openById(docTemplatedID) ;  
    var docFinal = DocumentApp.openById(docFinalID) ;  
  
    // จับทุกย่อหน้าในไฟล์ต้นทาง  
    var docTemplateParagraphs = docTemplate.getBody().getParagraphs();  
  
    // ก๊อปปี้+ค้นหาและแทนที่+วางไปที่ไฟล์ปลายทาง โดยทำทีละย่อหน้าโดยใช้ forEach  
    docTemplateParagraphs.forEach(function(p){  
        docFinal.getBody().appendParagraph(p.copy().replaceText("{first}", "วสันต์"));  
    });  
}
```

#### 1.4.ค.) ค้นหาหลายข้อความ ในทุกย่อหน้า

โค้ดขี้อก่อนหน้า ค้นหาและแทนที่เพียง 1 ข้อความ แต่โค้ดต่อไปนี้จะ ค้นหาและแทนที่หลายข้อความ

```
function pDoc1_to_Doc2_R2_3() {  
  
    var docTemplatedID = "1Gc8MC7o4O68dVekknOPLGcmdcWI82wUjplH-x8HwXQc" ;  
    var docFinalID = "1aqAv5bzhIvxWPheMpaQJu2vPTWZQQKd3qIsyGPlq-qM" ;  
    var wsID = "1IKcuYYWBZGgKvklF3SNCvTM-OA7gyYjvTxzhEMmVjzE";  
  
    var docTemplate = DocumentApp.openById(docTemplatedID) ;  
    var docFinal = DocumentApp.openById(docFinalID) ;  
  
    // Get ย่อหน้าทั้งหมดในไฟล์ต้นทาง  
    var docTemplateParagraphs = docTemplate.getBody().getParagraphs();  
  
    // ก๊อปปี้ + แทนที่หลายข้อความ + วางลงไฟล์ปลายทาง โดยทำทีละย่อหน้า โดยใช้ forEachLoop  
    docTemplateParagraphs.forEach(function(p){  
        docFinal.getBody().appendParagraph(p.copy()  
                                            .replaceText("{first}", "มะนาว")  
                                            .replaceText("{last}", "เปรี้ยวจัด")  
                                            .replaceText("{phone}", "080-000-0000")  
                                            );  
    }); //จบ forEach  
}
```

#### 1.4.ง.) ค้นหาหลายข้อความ ในทุกย่อหน้า – สร้างเป็นฟังก์ชัน (2 พารามิเตอร์)

โค้ดต่อไปนี้ ให้ผลเหมือนกับโค้ดก่อนหน้า แต่ครั้งนี้จะสร้างเป็นฟังก์ชันแยกออกมา ใช้การส่งผ่านพารามิเตอร์ เพื่อให้โค้ดอ่านง่าย

```
function pDoc1_to_Doc2_R2_4() {  
  
    var docTemplatedID = "1Gc8MC7o4O68dVekknOPLGcmdcWI82wUjplH-x8HwXQc" ;  
    var docFinalID = "1aqAv5bzhIvxWPheMpaQJu2vPTWZQQKd3qIsyGPlq-qM" ;  
    var wsID = "1IKcuYYWBZGgKvklF3SNCvTM-OA7gyYjvTxzhEMmVjzE";  
  
    var docTemplate = DocumentApp.openById(docTemplatedID) ;  
    var docFinal = DocumentApp.openById(docFinalID) ;  
  
    var docTemplateParagraphs = docTemplate.getBody().getParagraphs();  
    docFinal.getBody().clear(); // ล้างทุกอย่างออกไปจาก Body ในเอกสารก่อน  
  
    // เรียกใช้ฟังก์ชัน – โดยส่ง 1.ไฟล์ต้นทาง และ 2.ไฟล์ปลายทาง ไปประมวลผล  
    createMailMerge(docTemplateParagraphs, docFinal);  
  
    function createMailMerge(docTemplateParagraphs, docFinal){  
        docTemplateParagraphs.forEach(function(p) {  
            docFinal.getBody().appendParagraph(p.copy()  
                                                .replaceText("{first}", "มะนาว")  
                                                .replaceText("{last}", "เปรี้ยวจัด")  
                                                .replaceText("{phone}", "080-000-0000")  
                                                );  
        }); //จบ forEach  
    }  
}
```

### 1.4.จ.) เปลี่ยนข้อความคงที่เป็นพารามิเตอร์ (3+2 พาราร)

ก่อนหน้านี้ ฟังก์ชันที่ส่งค่ามาแทนที่อยู่ในฟังก์ชันภายนอก แต่ตอนนี้ จะส่งข้อความที่จะมาแทนที่เป็นพารามิเตอร์ไปที่ฟังก์ชัน เตรียมไว้สำหรับการไปดึงข้อมูลมาจากไฟล์ Google Sheet

```
function pDoc1_to_Doc2_R2_4() {

  var docTemplatedID = "1Gc8MC7o4O68dVekknOPLGCmdcWI82wUjplH-x8HwXQc" ;
  var docFinalID = "1aqAv5bzhIvxWPheMpaQJu2vPTWZQQKd3qIsyGPlq-qM" ;
  var wsID = "1IKcuYYWBZGgKvkLF3SNCvTM-0A7gyYjvTxzhEMmVjzE";

  var docTemplate = DocumentApp.openById(docTemplatedID) ;
  var docFinal = DocumentApp.openById(docFinalID) ;

  var docTemplateParagraphs = docTemplate.getBody().getParagraphs();
  docFinal.getBody().clear();

  // เรียกใช้ฟังก์ชัน – โดยส่งพารามิเตอร์ 5 ตัว
  // 1.ชื่อ 2.นามสกุล 3.เบอร์โทรฯ 4.ไฟต้นทาง และ 5.ไฟล์ปลายทาง
  createMailMerge("ชื่อ","นามสกุล","เบอร์โทรศัพท์",docTemplateParagraphs,docFinal);
}

function createMailMerge(first,last,phone,docTemplateParagraphs,docFinal){
  docTemplateParagraphs.forEach(function(p) {
    docFinal.getBody().appendParagraph(p.copy()
                                     .replaceText("{first}",first)
                                     .replaceText("{last}",last)
                                     .replaceText("{phone}",phone)
    );
  }); //จบ forEach
}
```

### 1.5. พัฒนาการที่ 3 : เตรียมข้อมูลจากไฟล์ Google Sheet

#### 1.5.ก.) ทดลองเอาข้อมูลจาก Google Sheet มา Log ดู

โค้ดต่อไปนี้จะดึงข้อมูลจากรายในไฟล์ Google Sheet มา Log ดู

```
function Sheet2Docs_1() {

  // ข้อมูลสำหรับเปิดไฟล์ และ ชีท
  var wsID = "1IKcuYYWBZGgKvkLF3SNCvTM-0A7gyYjvTxzhEMmVjzE" ;
  var wsSheetName = "data" ;

  // เอาชีทในไฟล์ มาใส่ไว้ในตัวแปร ws
  var ws = SpreadsheetApp.openById(wsID).getSheetByName(wsSheetName) ;

  // เอาข้อมูล มาใส่ไว้ในตัวแปร data
  var data = ws.getRange(2, 1, ws.getLastRow()-1,ws.getLastColumn()).getValues() ;

  Logger.log(data);
}
```

#### Logs

```
[ 01 ] [ [1.0, ตัวเล็ก, คุณติลกเศวต, tualek@gmail.com, 091-399-6766, กรุงเทพฯ],  
        [2.0, นภาพร, ติลาพันธ์, napa.por@gmail.com, 095-768-0670, พิษณุโลก],  
        [3.0, ภูคลับ, สวรรค์บนดิน, poeclub@gmail.com, 081-459-8343, เพชรบูรณ์] ]
```

### 1.5.ข.) เอาข้อมูลจากชีท ไปใส่ในเอกสาร

คราวนี้ จะเอาข้อมูลจากชีทแต่ละบรรทัด ไปวนลูปเพื่อพิมพ์ค่าลงในไฟล์ Google Docs ปลายทาง

```
function Sheet2Docs_2() {  
  
    var docTemplatedID = "1Gc8MC7o4O68dVekknoPLGCmdcWI82wUjplH-x8HwXQc" ;  
    var docFinalID = "1aqAv5bzhIvxWPheMpaQJu2vPTWZQQKd3qIsyGPlq-qM" ;  
  
    // ข้อมูลสำหรับเปิดไฟล์ และ ชีท  
    var wsID = "1IKcuYYWBZGgKvKLF3SNCvTM-0A7gyYjvTxzhEMmVjzE" ;  
    var wsSheetName = "data" ;  
  
    var docTemplate = DocumentApp.openById(docTemplatedID) ;  
    var docFinal = DocumentApp.openById(docFinalID) ;  
  
    // เอาชีทในไฟล์ มาใส่ไว้ในตัวแปร ws  
    var ws = SpreadsheetApp.openById(wsID).getSheetByName(wsSheetName) ;  
  
    // เอาข้อมูล มาใส่ไว้ในตัวแปร data  
    var data = ws.getRange(2, 1, ws.getLastRow()-1,ws.getLastColumn()).getValues() ;  
  
    var docTemplateParagraphs = docTemplate.getBody().getParagraphs();  
  
    docFinal.getBody().clear();  
  
    // วนลูปเอาอาเรย์ทีละ 1 แถว(คอลัมน์ไหนระบุได้) ไปใส่ฟังก์ชัน  
    data.forEach(function(r){  
        createMailMerge(r[0], r[1], r[3] , docTemplateParagraphs ,docFinal) ;  
    }); // End - forEach  
}  
  
function createMailMerge(first,last,phone,docTemplateParagraphs,docFinal){  
    docTemplateParagraphs.forEach(function(p) {  
        docFinal.getBody().appendParagraph(p.copy()  
                                            .replaceText("{first}",first)  
                                            .replaceText("{last}",last)  
                                            .replaceText("{phone}",phone)  
                                            );  
    }); //จบ forEach  
    docFinal.getBody().appendPageBreak() ; // ขึ้นหน้าใหม่  
}
```

### 1.6. ย่อหน้าที่เป็น List

กรณีเจอย่อหน้าที่เป็น List เช่น Bullets หรือ Numbering จะพบเจอปัญหา ซึ่งเราสามารถแก้ได้ดังต่อไปนี้

## 1.6.ก.) ตรวจสอบชนิดย่อหน้า

ย่อหน้าที่เป็น List เมื่อถูกก๊อปปี้ไปที่ไฟล์ปลายทาง จะถูกลอด List ออก กลายเป็นย่อหน้าธรรมดา หากต้องการเก็บ List เอาไว้ ให้ทำดังนี้

เขียนโค้ด ตรวจสอบชนิดของ 1 ย่อน้าก่อน

```
var docTemplateParagraphs_1 = docTemplate.getBody().getParagraphs()[0].getType();
```

```
function MM_Doc2Sheet() {  
  
    var docTemplatedID = "1Gc8MC7o4O68dVekknOPLGcmdcWI82wUjplH-x8HwXQc" ;  
    var docFinalID = "1aqAv5bzhIvxWPheMpaQJu2vPTWZQQKd3qIsyGPlq-qM" ;  
    var wsID = "1IKcuYYWBZGgKvkLF3SNCvTM-OA7gyYjvTxzhEMmVjzE" ;  
    var wsSheetName = "data" ;  
  
    var docTemplate = DocumentApp.openById(docTemplatedID) ;  
    var docFinal = DocumentApp.openById(docFinalID) ;  
  
    var ws = SpreadsheetApp.openById(wsID).getSheetByName(wsSheetName) ;  
  
    var data = ws.getRange(2, 1, ws.getLastRow()-1,ws.getLastColumn()).getValues() ;  
  
    var docTemplateParagraphs = docTemplate.getBody().getParagraphs();  
  
    // ตรวจสอบชนิดข้อมูลของย่อหน้าที่ 1 *****  
    var docTemplateParagraphs_1 = docTemplate.getBody().getParagraphs()[0].getType();  
  
    docFinal.getBody().clear();  
  
    // วนหลูปเอาอาเรย์ทีละ 1 แถว(คอลัมน์ไหนระบุได้) ไปใส่ฟังก์ชัน  
    data.forEach(function(r){  
        createMailMerge(r[0], r[1], r[3] , docTemplateParagraphs ,docFinal);  
    }); // End – forEach
```

ย้ายไปใส่ในฟังก์ชัน แล้ว Log ดูก่อน ว่าโค้ดใช้ได้ จะพบ Type ของย่อหน้าที่เป็น List มีชนิดเป็น LIST\_ITEM ส่วนที่เป็นย่อหน้าธรรมดาเป็น PARAGRAPH

```
function createMailMerge(first,last,phone,docTemplateParagraphs,docFinal){  
  
    // Log ดูชนิดของย่อหน้า  
    Logger.log(p.getType());  
  
    docTemplateParagraphs.forEach(function(p) {  
        docFinal.getBody().appendParagraph(p.copy()  
                                                .replaceText("{first}",first)  
                                                .replaceText("{last}",last)  
                                                .replaceText("{phone}",phone)  
                                                );  
    }); //จบ forEach  
    docFinal.getBody().appendPageBreak() ; // ขึ้นหน้าใหม่  
}
```



## 1.6.ข.) ใช้ IF ตามชนิดของย่อหน้า

ส่วนของฟังก์ชันหลัก ไม่มีอะไรเปลี่ยนแปลง

```
function MM_Doc2Sheet() {  
  
    var docTemplatedID = "1Gc8MC7o4O68dVekknOPLGCmdcWI82wUjplH-x8HwXQc" ;  
    var docFinalID = "1aqAv5bzhIvxWPheMpaQJu2vPTWZQQKd3qIsyGPlq-qM" ;  
    var wsID = "IIKcuYYWBZGgKvklF3SNCvTM-OA7gyYjvTxzhEMmVjzE" ;  
    var wsSheetName = "data" ;  
  
    var docTemplate = DocumentApp.openById(docTemplatedID) ;  
    var docFinal = DocumentApp.openById(docFinalID) ;  
  
    var ws = SpreadsheetApp.openById(wsID).getSheetByName(wsSheetName) ;  
  
    var data = ws.getRange(2, 1, ws.getLastRow()-1,ws.getLastColumn()).getValues() ;  
  
    var docTemplateParagraphs = docTemplate.getBody().getParagraphs();  
  
    // ตรวจสอบชนิดข้อมูลของย่อหน้าที่ 1 *****  
    var docTemplateParagraphs_1 = docTemplate.getBody().getParagraphs()[0].getType();  
  
    docFinal.getBody().clear();  
  
    // วนลูปเอาอาเรย์ทีละ 1 แถว(คอลัมน์ไหนระบุได้) ไปใส่ฟังก์ชัน  
    data.forEach(function(r){  
        createMailMerge(r[0], r[1], r[3] , docTemplateParagraphs ,docFinal) ;  
    }); // End – forEach
```

ส่วนของฟังก์ชัน createMailMerge ที่ใช้ก๊อปปี้ข้อมูลไปที่ไฟล์ปลายทาง ใส่ if เพื่อทำตามเงื่อนไข

ถ้าเป็น **PARAGRAPH** ก็ให้ทำตามปกติ อย่างที่ทำมา

แต่ถ้าเป็น **LIST\_ITEM** หลังจาก getBody() แล้ว ให้ต่อด้วย appendListItem แก่จากของเดิมที่เป็น appendParagraph จากนั้นแทนที่ข้อความใน LIST\_ITEM จากนั้นอีก ให้เซ็่รูปแบบของ LIST\_ITEM โดยใช้ setGlyphType() โดยเลือกชนิดของ LIST\_ITEM จาก DocumentApp.GlyphType.BULLET

```

function createMailMerge(first,last,phone,docTemplateParagraphs,docFinal){
    docTemplateParagraphs.forEach(function(p) {
// ตรวจสอบชนิดของย่อหน้า แล้วจับใส่ไว้ในตัวแปร
var elType = p.getType();

// ถ้าเป็น PARAGRAPH ก็ให้ทำตามปกติ อย่างที่ทำมา
if (elType == "PARAGRAPH" ) {
    docFinal.getBody().appendParagraph(p.copy()
                                        .replaceText("{first}",first)
                                        .replaceText("{last}",last)
                                        .replaceText("{phone}",phone)
                                        );
} // จบ if

// ถ้าเป็น LIST_ITEM ก็ให้ทำต่อไปนี้
if (elType == "LIST_ITEM" ) {
    docFinal.getBody().appendListItem(p.copy()
                                        .replaceText("{first}",first)
                                        .replaceText("{last}",last)
                                        .replaceText("{phone}",phone)
                                        ).setGlyphType(DocumentApp.GlyphType.BULLET);
} // จบ if

}); //จบ forEach

docFinal.getBody().appendPageBreak() ; // ขึ้นหน้าใหม่
}

```

บทที่ 2  
โปรเจ็ค 02  
ส่งเมลโดยใช้  
MailApp



## 2.1. ที่มาของโปรเจ็ค 02

Google Sheets - Send Emails Using Apps Script JavaScript MailApp Tutorial - Part 12

<https://www.youtube.com/watch?v=ZcNmur6xiX4>

## 2.2. ภาพรวมโปรเจ็ค 02

ไฟล์ **Data\_Email** (Google Sheets) มีข้อมูลใน 2 ชีท ก็คือ

**ชีทที่ 1** : ชื่อ **sEmail** เป็นข้อมูลที่จะใช้ส่งอีเมล

	A	B	C	D
1	Email	Name	Class	
2	wasankds@gmail.com	Wasan	Google Sheet	
3	wasan@snpfood.com	SnP	LibreOffice Calc	
4	info.poeclub@gmail.com	Poeclub	Gimp	

**ชีทที่ 2** : ชื่อ **sTemplate** เป็น Template สำหรับส่งอีเมล โดยพิมพ์ข้อความที่จะใช้เป็น Template ลงในเซลล์ A1 เซลล์เดียว

ไฟล์โปรเจ็ค Google Apps Script สร้างฝังไว้ที่ไฟล์ Google Sheet ไฟล์นี้

## 2.3. พัฒนาการที่ 1 - จับค่าในคอลัมน์ออกมา Log ดู

จับค่าในคอลัมน์ออกมา Log ดู

```
function sendMails() {  
    // จับไปที่ชีทที่เปิดอยู่  
    var ss = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
    // จับไปที่แถวสุดท้าย - ได้เลขจำนวนเต็มเช่น 4  
    var lr = ss.getLastRow();  
    // จับไปที่คอลัมน์สุดท้าย - ได้เลขจำนวนเต็มเช่น 3 (ไม่ได้ใช้ในโค้ดนี้)  
    var lc = ss.getLastColumn();  
    // วนloopเพื่อ ดึงค่าในคอลัมน์ที่ระบุ ทุกแถวออกมาใช้งาน  
    for(var i = 2 ; i <= lr ; i++) {  
        // จับค่าในแถว i คอลัมน์ที่ 1  
        var currentEmail = ss.getRange(i, 1).getValue();  
        Logger.log(currentEmail); // Log ดู  
    }; // จบ for  
}
```

ผล - Log ค่าในคอลัมน์ที่เป็นอีเมลออกมาที่ละบรรทัด

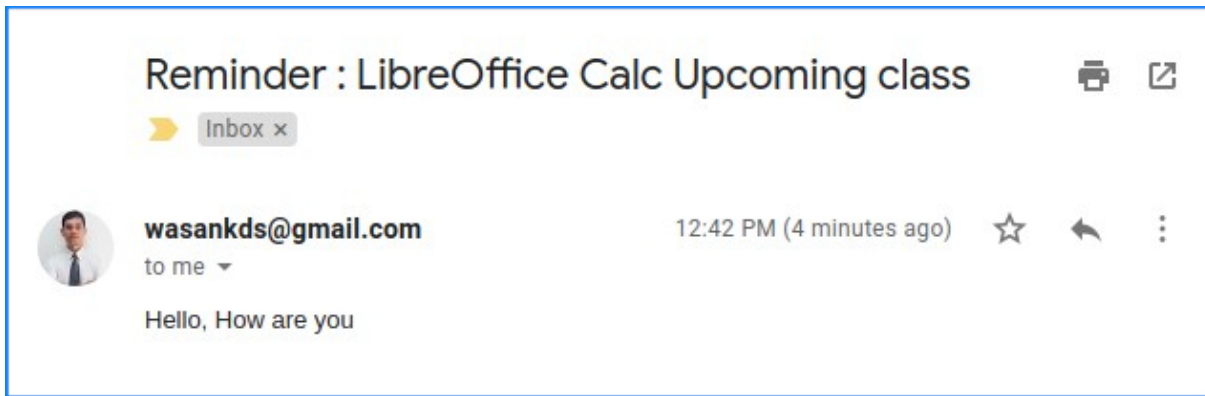
```
Logs  
[ 01 ] wasankds@gmail.com  
[ 02 ] wasan@snpfood.com  
[ 03 ] info.poeclub@gmail.com
```

## 2.4. พัฒนาการที่ 2 - ใช้ MailApp ส่งเมล

โค้ดต่อไปนี้จะใช้คลาส MailApp ส่งเมล โดยดึงผู้รับเมลมาจาก Google Sheet

```
function sendMails() {  
    var ss = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
    var lr = ss.getLastRow();  
    // เนื้อหาในอีเมล- ทดลองส่งสั้นๆไปก่อน  
    var body = "Hello, How are you";  
    for(var i = 2 ; i <= lr ; i++) {  
        var currentEmail = ss.getRange(i, 1).getValue();  
        var currentClass = ss.getRange(i, 3).getValue();  
        MailApp.sendEmail(currentEmail, // ผู้รับ  
                           "Reminder : " + currentClass + " Upcoming class" , // หัวเรื่อง  
                           body // เนื้อหา  
                        )  
    }; // จบ for  
}
```

ผล - ผู้รับเมลได้รับเมลดังต่อไปนี้



## 2.5. พัฒนาการที่ 3 – จับเนื้อหาใน Template ออกมา Log ดู

โค้ดนี้ จะจับเนื้อหาในเซลล์ A1 ในชีท sTemplate ออกมา Log ดูก่อน ซึ่งจะใช้เป็น Template ในการเขียนอีเมลต่อไป

เนื่องจากเรามี 2 ชีท จึงต้องเขียนโค้ดให้เลือกชีท ที่เราจะทำงานด้วย

```
function sendMails() {  
  // เลือกชีท Email – คล้ายการคลิกเลือกที่ป้ายชื่อ Email  
  SpreadsheetApp.getActiveSpreadsheet().getSheetByName("Email").activate();  
  
  var ss = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet(); // ชีท Email แอ็คทีฟอยู่  
  var lr = ss.getLastRow();  
  
  // มีหรือไม่มีก็ได้  
  // SpreadsheetApp.getActiveSpreadsheet().getSheetByName("sTemplate").activate();  
  
  // จับค่าในเซลล์ A1 ในชีท sTemplate  
  var templateText = SpreadsheetApp.getActiveSpreadsheet().  
    getSheetByName("sTemplate").getRange("A1").getValue();  
  
  Logger.log(templateText); // ดูผลได้ที่ Logs  
}
```

ผล

### Logs

[ 01 ] สวัสดี คุณ {name}

หลักสูตร {class} กำลังจะเปิดอบรม ในเร็วนี้ๆ คุณสามารถดูรายละเอียดได้ตามลิงค์ต่อไปนี้  
[www.poeclub.org](http://www.poeclub.org)

ขอแสดงความนับถือ  
วสันต์ คุณดิลกเศวต  
(สัน เซฟคอสต์)

## 2.6. พัฒนาการที่ 4 – แทนที่ข้อความใน Template

แทนที่ข้อความในเซลล์ A1 ในชีต sTemplate ในส่วนที่เป็น Mail Merge เช่น {name} เป็นต้น

```
function sendMails() {  
    // เลือกที่ชีต Email – คล้ายการคลิกเลือกที่ป้ายชื่อ Email  
    SpreadsheetApp.getActiveSpreadsheet().getSheetByName("Email").activate();  
  
    var ss = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet(); // ชีต Email แอ็คทีฟอยู่  
    var lr = ss.getLastRow();  
  
    // มีหรือไม่มีก็ได้  
    // SpreadsheetApp.getActiveSpreadsheet().getSheetByName("sTemplate").activate();  
  
    // จับค่าในเซลล์ A1 ในชีต sTemplate  
    var templateText = SpreadsheetApp.getActiveSpreadsheet().  
        getSheetByName("sTemplate").getRange("A1").getValue();  
  
    var templateTextReplaced = templateText  
        .replace("{name}", "วสันต์")  
        .replace("{class}", "LibreOffice Base") ;  
  
    Logger.log(templateTextReplaced);  
}
```

ผล

### Logs

[บรรทัดที่ 1] สวัสดิ์ คุณ วสันต์

หลักสูตร LibreOffice Base กำลังจะเปิดอบรม ในเร็วนี้ๆ คุณสามารถดูรายละเอียดได้ตามลิงค์ต่อไปนี้  
[www.poeclub.org](http://www.poeclub.org)

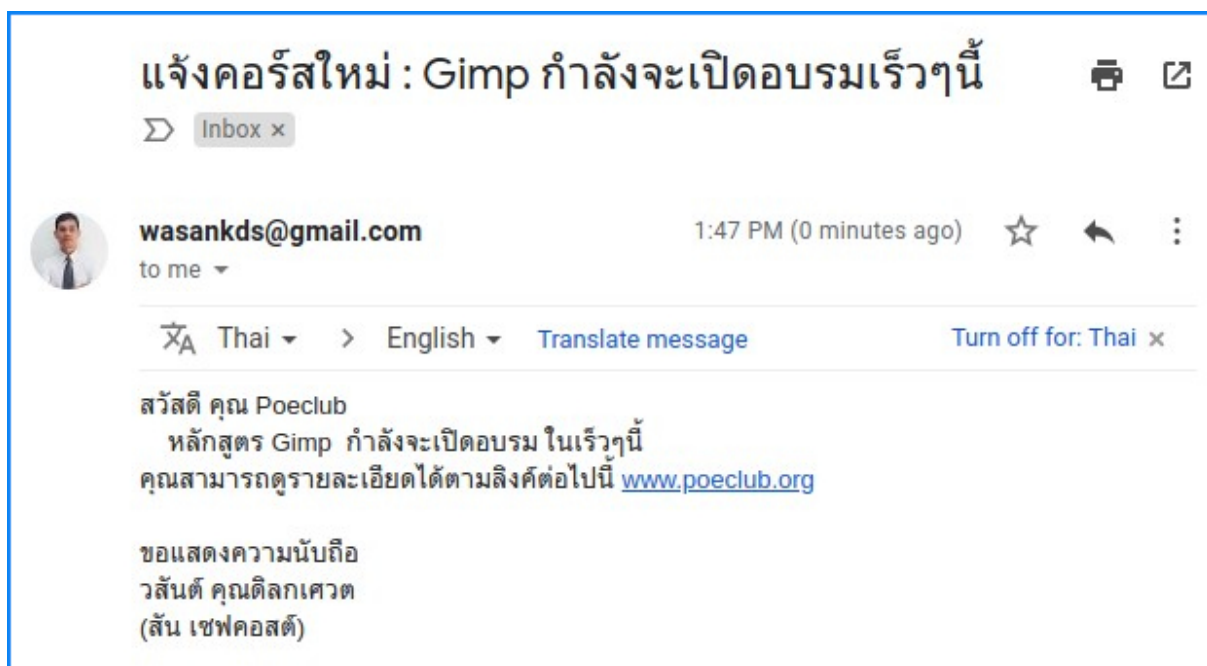
ขอแสดงความนับถือ  
วสันต์ คุณดิลกเสวต  
(สัน เชฟคอสต์)

## 2.7. พัฒนาการที่ 5 – ส่ง Mail Merge email

ข้อนี้ จะเป็นการส่ง Mail Merge email โดยจับตัวแปรที่ใช้เก็บ Template และ ข้อมูลรายบุคคล สำหรับการส่งอีเมล ไปใส่ใน for เพื่อนวนลูปสำหรับส่งอีเมลให้กับแต่ละคน

```
function sendMails() {  
  // เลือกที่ชี้ Email – คล้ายการคลิกเลือกที่ป้ายชื่อ Email  
  SpreadsheetApp.getActiveSpreadsheet().getSheetByName("Email").activate();  
  
  var ss = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet(); // ชี้ Email แอ็คทีฟอยู่  
  var lr = ss.getLastRow();  
  
  var templateText = SpreadsheetApp.getActiveSpreadsheet().  
    getSheetByName("sTemplate").getRange("A1").getValue();  
  
  for(var i = 2 ; i <= lr ; i++) {  
    var currentEmail = ss.getRange(i, 1).getValue();  
    var currentName = ss.getRange(i, 2).getValue(); // จับชื่อ  
    var currentClass = ss.getRange(i, 3).getValue();  
  
    var templateTextReplaced = templateText  
      .replace("{name}",currentName)  
      .replace("{class}",currentClass) ;  
  
    var subjectLine = "แจ้งคอร์สใหม่ : " + currentClass + " กำลังจะเปิดอบรมเร็วๆนี้" ;  
  
    MailApp.sendEmail(currentEmail, subjectLine, templateTextReplaced) ;  
  } ; // จบ for  
}
```

ผล





## 2.8. พัฒนาการที่ 6 – ส่งตาม Quota limit

เราส่งอีเมลต่อวันได้จำกัด เราเรียกว่า Quota limit แล้วแต่ประเภทของ Google Account โดยบัญชีแบบฟรีส่งได้ 100 เมลต์ต่อวัน

ตรวจสอบ Quota limit ได้ดังนี้

```
var quotaLimit = MailApp.getRemainingDailyQuota();
Logger.log(quotaLimit);
```

ปัญหามันจะเกิด ถ้าจำนวนอีเมลที่เราจะส่งเหลือน้อยกว่า Quota limit ฉะนั้นให้ตรวจสอบ Quota ก่อน ถ้าพอก็ส่ง ไม่พอก็ให้ขึ้น Message Box แจ้ง

```
function sendMails() {

    SpreadsheetApp.getActiveSpreadsheet().getSheetByName("Email").activate();

    var ss = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
    var lr = ss.getLastRow();

    var templateText = SpreadsheetApp.getActiveSpreadsheet().
        getSheetByName("sTemplate").getRange("A1").getValue();

    // ตอนตรวจสอบเหลือ 84 ก็เลยใส่ -83 เพื่อทดสอบ Qouta ไม่พอ - ใช้งานจริงให้ลบตรงนี้ออก
    var quotaLeft = MailApp.getRemainingDailyQuota() - 83;

    // Logger.log(quotaLeft);

    // ถ้ามี Quota ไม่พอให้ขึ้น Message Box
    if(lr-1 > quotaLeft){
        Browser.msgBox("คุณมี Quota limit เหลือเพียง " + quotaLeft +
            " แต่คุณกำลังจะส่งอีเมลจำนวน " + (lr-1) + " ฉบับ" +
            " ฉะนั้นอีเมลจะไม่ถูกส่ง");
    } /* จบ if */

    // ถ้า Quota พอให้ส่งอีเมล
    else {
        for(var i = 2 ; i <= lr ; i++) {
            var currentEmail = ss.getRange(i, 1).getValue();
            var currentName = ss.getRange(i, 2).getValue();
            var currentClass = ss.getRange(i, 3).getValue();

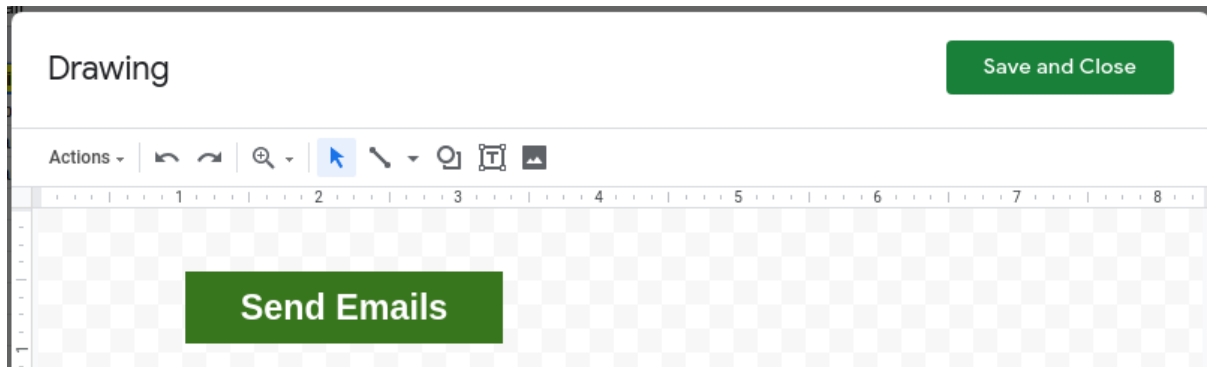
            var templateTextReplaced = templateText
                .replace("{name}",currentName)
                .replace("{class}",currentClass) ;

            var subjectLine = "แจ้งคอร์สใหม่ : " + currentClass + " กำลังจะเปิดอบรมเร็วๆนี้" ;

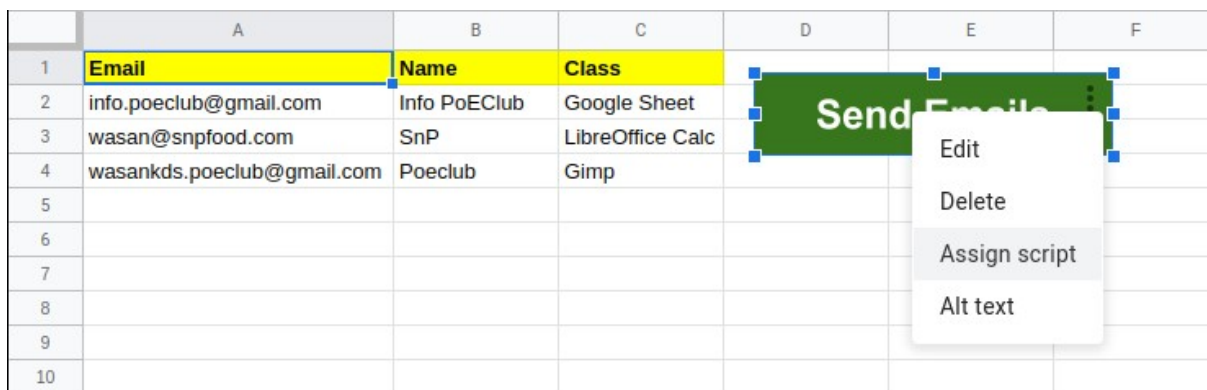
            MailApp.sendEmail(currentEmail, subjectLine, templateTextReplaced) ;
        } ; // จบ for
        Browser.msgBox("ส่งอีเมลไป " + (lr-1) + " ฉบับ");
    } ; // จบ for
}
```

## 2.9. พัฒนาการที่ 7 – สร้างปุ่ม

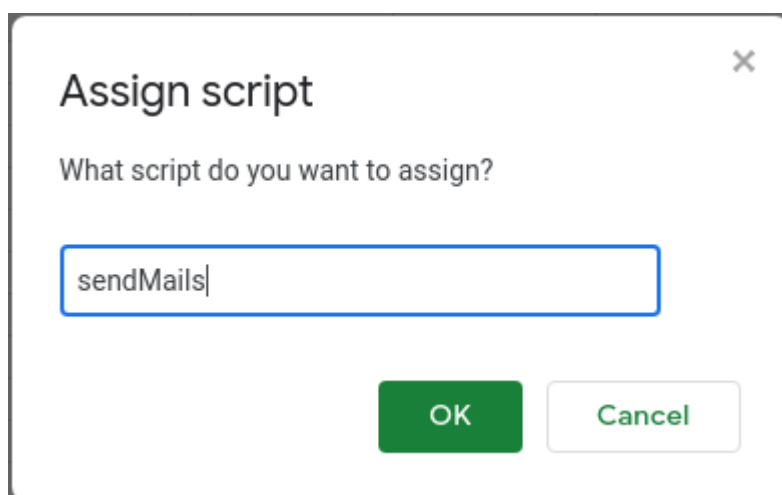
ที่ Google Sheet ไปที่เมนู **Insert** → **Drawing** จะปรากฏหน้าต่างให้วาดรูปวาด หรือ Drawing Object ให้เราสร้าง Drawing Object ที่จะใช้เป็นปุ่ม ตัวอย่างตามภาพ



ที่ Drawing Object ที่สร้างไป คลิกที่ ปุ่ม 3 จุด → **Assign Script** ตามภาพ



ที่หน้าต่าง **Assign Script** พิมพ์ชื่อฟังก์ชันที่เราเขียน Script ลงไป เช่น **sendMails**



จากนั้น เมื่อเราคลิกที่ Drawing Object ตัวนี้ก็จะเป็นการรันสคริปต์

บทที่ 3  
โปรเจ็ค 03  
Mail Merge จาก Sheets  
โดยใช้ HTML Template



### 3.1. ที่มา

Google Sheets Mail Merge - Email - No Addons – Tutorial

<https://www.youtube.com/watch?v=h2z13YE3kJg>

โปรเจกต์นี้ ส่งอีเมลแบบ Mail Merge โดยดึงข้อมูลอีเมล ชื่อผู้ส่ง และอื่นๆ มาจากไฟล์ Google Sheet โดยเนื้อหาของอีเมลมาจาก Template ในไฟล์ HTML

### 3.2. พัฒนาการที่ 1 : เตรียมพร้อมไฟล์ Google Sheets

#### 3.2.ก.) สร้างไฟล์ Google Sheet

เริ่มต้นให้สร้างไฟล์ Google Sheet จากนั้นใส่ข้อมูลชื่อ นามสกุล และ เบอร์โทรศัพท์ ลงไป ตัวอย่างข้อมูลตามภาพ

	A	B	C	D
1	<b>Email</b>	<b>Name</b>	<b>Last Name</b>	<b>Phone</b>
2	info.poeclub@gmail.com	นภาพร	คุณติลกเสวต	081-111-1111
3	wasan@snpfood.com	wasan	snp	095-222-2222
4	wasankds.poeclub@gmail.com	PoE	Club	031-333-3333

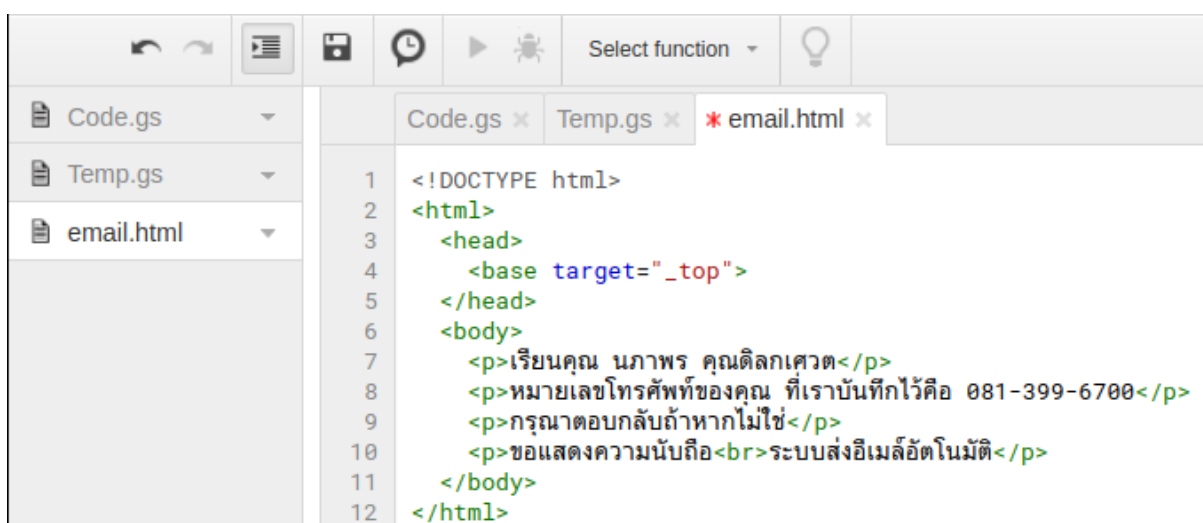
#### 3.2.ข.) สร้างไฟล์ Google Apps Script

สร้างโปรเจกต์ Google Apps Script แบบฝังกับไฟล์ Google Sheet ไว้

### 3.3. พัฒนาการที่ 2 : สร้าง Template ไฟล์ HTML

ที่โปรเจกต์ Google Apps Script สร้างไฟล์ HTML โดยไปที่เมนู File → New → HTML

จากนั้นเขียนโค้ด HTML เพื่อสร้างเป็น Template สำหรับส่งอีเมล



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <base target="_top">
5   </head>
6   <body>
7     <p>เรียนคุณ นภาพร คุณติลกเสวต</p>
8     <p>หมายเลขโทรศัพท์ของคุณ ที่เรบันทึกไว้คือ 081-399-6700</p>
9     <p>กรุณาตอบกลับถ้าหากไม่ใช่</p>
10    <p>ขอแสดงความนับถือ<br>ระบบส่งอีเมลอัตโนมัติ</p>
11  </body>
12 </html>
```

```

<!DOCTYPE html>

<html>

  <head>
    <base target="_top">
  </head>

  <body>
    <p>เรียนคุณ นภาพร คุณดิลกเศวต</p>
    <p>หมายเลขโทรศัพท์ของคุณ ที่เรานัดกันไว้คือ 081-399-6700</p>
    <p>กรุณาตอบกลับถ้าหากไม่ใช่</p>
    <p>ขอแสดงความนับถือ<br>ระบบส่งอีเมลอัตโนมัติ</p>
  </body>

</html>

```

### 3.4. พัฒนาการที่ 3 : แทรกสคริปต์เตรียมเรียกใช้ฟังก์ชัน

ข้อความหรือจุด ที่จะเปลี่ยนเป็นข้อมูลที่ดึงมาจาก Google Sheet เช่น ชื่อ หรือ เบอร์โทรศัพท์ ให้เราแทรกบล็อก `<? ?>` เพื่อจะแทรกสคริปต์(Script

let) เตรียมไว้สำหรับเรียกใช้ฟังก์ชันจากไฟล์ Google Apps Script (ไฟล์ .gs)

โค้ดในบล็อก `<? ?>` เป็นสคริปต์ที่รันในฝั่ง Server side แล้วคืนค่ากลับมาเป็น HTML แล้วส่งต่อให้ Browser รับไปแสดงผล

```

<!DOCTYPE html>

<html>

  <head>
    <base target="_top">
  </head>

  <body>
    <p>เรียนคุณ <?= fn ?> <?= ln ?></p>
    <p>หมายเลขโทรศัพท์ของคุณ ที่เรานัดกันไว้คือ <?= phone ?></p>
    <p>กรุณาตอบกลับถ้าหากไม่ใช่</p>
    <p>ขอแสดงความนับถือ<br>ระบบส่งอีเมลอัตโนมัติ</p>
  </body>

</html>

```

#### หมายเหตุ

`<?=` เขียนติดกัน เว้นแบบนี้ไม่ได้ `<? =`

### 3.5. พัฒนาการที่ 4 : ดึงข้อมูลจาก Google Sheets

#### 3.5.ก.) ดึงข้อมูลจาก Google Sheets มา Log ดู

จับข้อมูลในชีทก่อน โดยจับมาทุกบรรทัดที่มีข้อมูล

```
function myFunction() {  
  var ws = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("DataEmail");  
  var data = ws.getRange("A2:D"+ws.getLastRow()).getValues();  
  Logger.log(data);  
}
```

ผล

##### Logs

```
[ ] [[info.poeclub@gmail.com, นภาพร, คุณติลกเสวต, 081-111-1111], [wasan@snpfood.com, wasan,  
snp, 095-222-2222], [wasankds.poeclub@gmail.com, PoE, Club, 031-333-3333]]
```

#### 3.5.ข.) ใช้ forEach เข้าไปดึงอีเมลทุกบรรทัดมา Log ดู

ข้อมูลที่จับมาจาก Google Sheet เราจะใช้ forEach วนลูปเข้าไปในอาร์เรย์ โดยจะดึงข้อมูลมาได้ที่ละบรรทัด เพราะเป็นอาร์เรย์ 2 มิติ อย่างไรก็ตามเราสามารถจับข้อมูลในคอลัมน์ที่ต้องการได้โดยระบุสมาชิกในแถว เช่น row[2] หมายถึง คอลัมน์ที่ 3(ดรรชนีลำดับที่ 2) ในแถวข้อมูล

```
function myFunction() {  
  var email = 0;  
  var first = 1 ;  
  var last = 2;  
  var phone = 3;  
  
  var ws = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("DataEmail");  
  var data = ws.getRange("A2:D"+ws.getLastRow()).getValues();  
  
  data.forEach(function(row){  
    Logger.log(row[email]); // Log สมาชิกอาร์เรย์ในคอลัมน์ที่ 1 (ที่ละแถว)  
  }); // End – forEach  
}
```

ผล

##### Logs

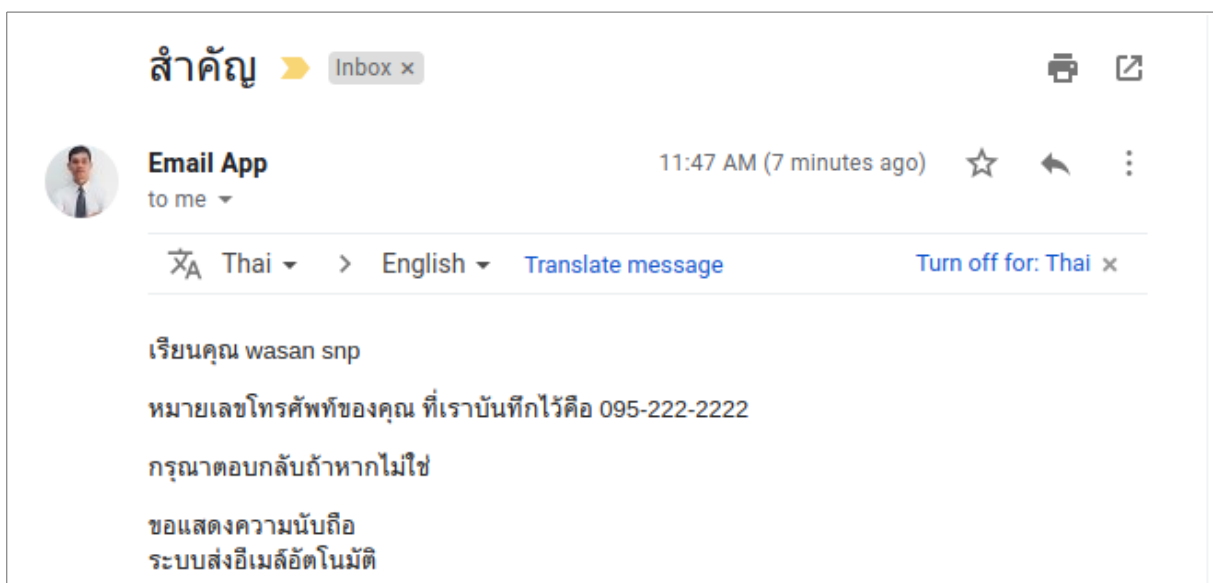
```
[ 01 ] info.poeclub@gmail.com  
[ 02 ] wasan@snpfood.com  
[ 03 ] wasankds.poeclub@gmail.com
```

### 3.6. พัฒนาการที่ 5 : วนหลูบส่งเมลล์โดยใช้ Template จากไฟล์ HTML

ขั้นตอนนี้ จะเป็นการวนหลูบเพื่อส่งเมลล์ โดยใช้ Template จากไฟล์ HTML

```
function myFunction() {  
  var email = 0;  
  var first = 1 ;  
  var last = 2;  
  var phone = 3;  
  
  // สร้าง Template จากไฟล์ HTML ในโปรเจ็ค  
  var emailTemp = HtmlService.createTemplateFromFile("email");  
  
  var ws = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("DataEmail");  
  var data = ws.getRange("A2:D"+ws.getLastRow()).getValues();  
  
  // วนหลูบส่งอีเมลล์  
  data.forEach(function(row){  
  
    emailTemp.fn = row[first] ;           // จับค่าของ ชื่อ ไปใส่  
    emailTemp.ln = row[last] ;           // จับค่าของ นามสกุล ไปใส่  
    emailTemp.phone = row[phone] ;       // จับค่าของ เบอร์โทร ไปใส่  
  
    // จับเนื้อหาในไฟล์ HTML ที่เป็น Template  
    var htmlMessage = emailTemp.evaluate().getContent();  
    // Logger.log(htmlMessage); // ----- ดูต่อที่ผลการ Logs  
  
    GmailApp.sendEmail(  
      row[email] ,  
      "สำคัญ" ,  
      "อีเมลล์ของคุณไม่สนับสนุน HTML" ,  
      {name : "Email App", htmlBody:htmlMessage}  
    ); // End - sendMail  
  }); // End - forEach  
}
```

ผล - อีเมลล์ถูกส่งออกไป ผู้รับได้รับอีเมลล์ลักษณะตามภาพ



[ 01 ]

```
<!DOCTYPE html>
<html>
  <head>
    <base target="_top">
  </head>

  <body>
    <p>เรียนคุณ นภาพร คุณดิลกเศวต</p>
    <p>หมายเลขโทรศัพท์ของคุณ ที่เราบันทึกไว้คือ 081-111-1111</p>
    <p>กรุณาตอบกลับถ้าหากไม่ใช่</p>
    <p>ขอแสดงความนับถือ<br>ระบบส่งอีเมลอัตโนมัติ</p>
  </body>
</html>
```

[ 02 ]

```
<!DOCTYPE html>
<html>
  <head>
    <base target="_top">
  </head>

  <body>
    <p>เรียนคุณ wasan snp</p>
    <p>หมายเลขโทรศัพท์ของคุณ ที่เราบันทึกไว้คือ 095-222-2222</p>
    <p>กรุณาตอบกลับถ้าหากไม่ใช่</p>
    <p>ขอแสดงความนับถือ<br>ระบบส่งอีเมลอัตโนมัติ</p>
  </body>
</html>
```

[ 03 ]

```
<!DOCTYPE html>
<html>
  <head>
    <base target="_top">
  </head>

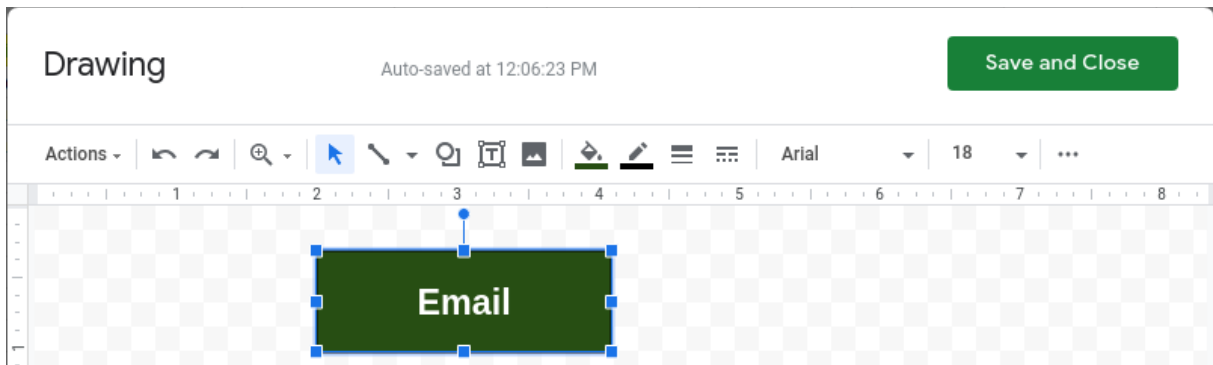
  <body>
    <p>เรียนคุณ PoE Club</p>
    <p>หมายเลขโทรศัพท์ของคุณ ที่เราบันทึกไว้คือ 031-333-3333</p>
    <p>กรุณาตอบกลับถ้าหากไม่ใช่</p>
    <p>ขอแสดงความนับถือ<br>ระบบส่งอีเมลอัตโนมัติ</p>
  </body>
</html>
```



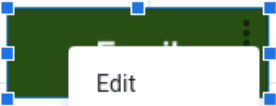
### 3.7. พัฒนาการที่ 6 : สร้างปุ่มเพื่อรันสคริปต์

สร้างปุ่มเพื่อรันสคริปต์ โดยสร้างจาก Drawing Object แล้ว Assign ชื่อฟังก์ชันลงไป

ไปที่ Insert → Drawing



ที่ Drawing object ทำการ Assign script โดย คลิกที่ปุ่ม 3 จุด → Assign script จะปรากฏหน้าต่างมาให้ พิมพ์ชื่อของฟังก์ชันที่จะผูกกับ Drawing Object

	A	B	C	D	E	F
1	Email	Name	Last Name	Phone		
2	info.poeclub@gmail.com	นภาพร	คุณดิลกเศวต	081-111-1111		
3	wasan@snpfood.com	wasan	snp	095-222-2222		
4	wasankds.poeclub@gmail.com	PoE	Club	031-333-3333		
5						
6						
7						
8						
9						
10						
11						
12						

×

Assign script

What script do you want to assign?

OK

Cancel

จากนี้ไป เวลาจะรันสคริปต์เพื่อส่งอีเมล ให้คลิกที่ Drawing Object ได้เลย

### 3.8. พัฒนาการที่ 7 : ฝาก Subject และ Name ไว้ในชีทที่ 2

ในขั้นตอนนี้ เราจะฝาก Subject และ Name(ผู้ส่ง) ไว้ในชีทที่ 2 (ชื่อ Settings) ตัวอย่างตามภาพ เพื่อที่จะได้แก้ไขหัวข้อการส่งได้ง่าย

	A	B
1	<b>Subject</b>	เมลล์สำคัญอีกฉบับ
2	<b>Name</b>	แอฟของเรา

```
function myFunction() {
  var email = 0;
  var first = 1 ;
  var last = 2;
  var phone = 3;

  var emailTemp = HtmlService.createTemplateFromFile("email");

  var ws = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("DataEmail");
  var wsSettings = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("Settings");

  // จับค่าในเซลล์ ในชีทที่ 2
  var name = wsSettings.getRange("B2").getValue();
  var subject = wsSettings.getRange("B1").getValue();

  var data = ws.getRange("A2:D"+ws.getLastRow()).getValues();

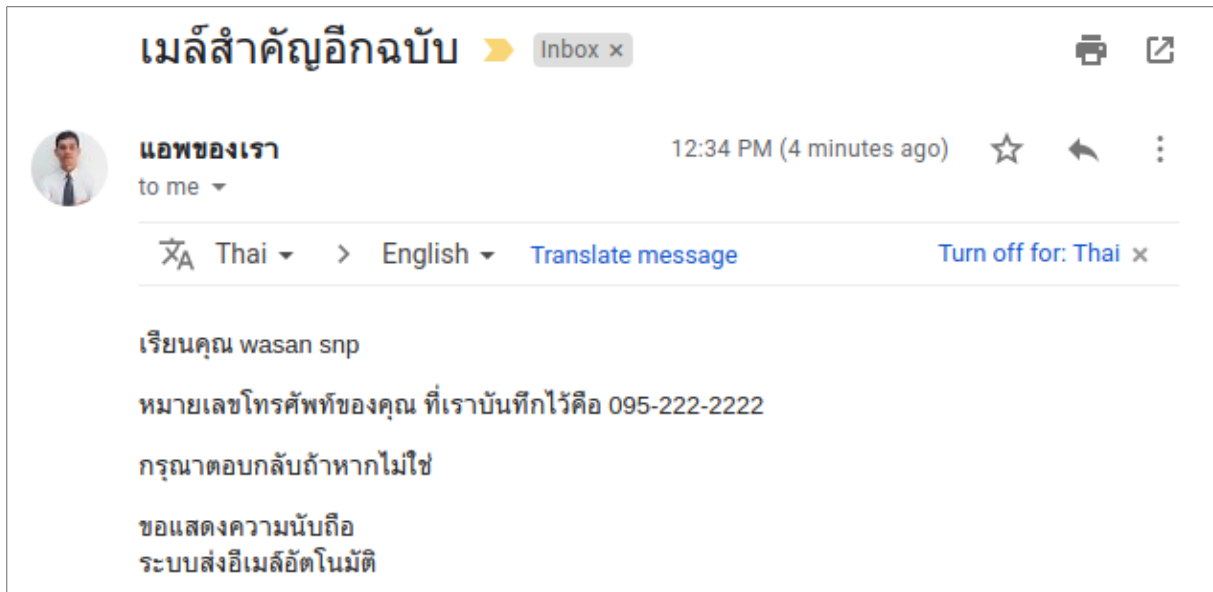
  data.forEach(function(row){

    emailTemp.fn = row[first] ;           // จับค่าของ ชื่อ ไปใส่
    emailTemp.ln = row[last] ;           // จับค่าของ นามสกุล ไปใส่
    emailTemp.phone = row[phone] ;       // จับค่าของ เบอร์โทร ไปใส่

    // จับเนื้อหาในไฟล์ HTML ที่เป็น Template
    var htmlMessage = emailTemp.evaluate().getContent();
    // Logger.log(htmlMessage); // ----- ดูต่อที่ผลการ Logs

    GmailApp.sendEmail(
      row[email] ,
      subject ,
      "อีเมลของคุณไม่สนับสนุน HTML" ,
      {name : name, htmlBody:htmlMessage}
    ); // End - sendMail
  }); // End - forEach
}
```

หลังจากส่งอีเมลไปแล้ว ผู้รับได้รับอีเมลดังต่อไปนี้



### 3.9. พัฒนาการที่ 8 : ส่งใครบ้าง

ขั้นตอนนี้ เราใช้ Checkbox เพื่อกำหนดว่าจะส่งเมลให้ใครหรือไม่ส่งให้ใครบ้าง

	A	B	C	D	E
1	Email	Name	Last Name	Phone	Send ?
2	info.poeclub@gmail.com	นภาพร	คุณดิลกเสวต	081-111-1111	
3	wasan@snpfood.com	wasan	snp	095-222-2222	
4	wasankds.poeclub@gmail.com	PoE	Club	031-333-3333	

ในคอลัมน์ Send ? ใส่ Checkbox ลงไป โดยเลือกเซลล์ แล้วไปที่ Insert → Checkbox

checkbox เป็น บูลีน มีค่าเป็น true (Check) กัล false (Uncheck)

	A	B	C	D	E
1	Email	Name	Last Name	Phone	Send ?
2	info.poeclub@gmail.com	นภาพร	คุณดิลกเสวต	081-111-1111	<input type="checkbox"/>
3	wasan@snpfood.com	wasan	snp	095-222-2222	<input checked="" type="checkbox"/>
4	wasankds.poeclub@gmail.com	PoE	Club	031-333-3333	<input type="checkbox"/>

ถัดมาเราจะเพิ่มโค้ด โดยจะส่งอีเมลเฉพาะ Checkbox ที่ถูก Check เท่านั้น

เราจะเพิ่มบรรทัดนี้เข้าไป

```
// ส่งเมลเฉพาะบรรทัด Checkbox ถูก Check
data = data.filter(function(r){ return r[4] == true});
```

```

function myFunction() {
  var email = 0;
  var first = 1 ;
  var last = 2;
  var phone = 3;

  var emailTemp = HtmlService.createTemplateFromFile("email");

  var ws = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("DataEmail");
  var wsSettings = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("Settings");

  var name = wsSettings.getRange("B2").getValue();
  var subject = wsSettings.getRange("B1").getValue();

  var data = ws.getRange("A2:D"+ws.getLastRow()).getValues();

  // ส่งเมลเฉพาะบรรทัด Checkbox ถูก Check
  data = data.filter(function(r){ return r[4] == true});
  // Logger.log(data); // ----- ดูต่อที่ผลการ Logs

  data.forEach(function(row){

    emailTemp.fn = row[first] ;           // จับค่าของ ชื่อ ไปใส่
    emailTemp.ln = row[last] ;           // จับค่าของ นามสกุล ไปใส่
    emailTemp.phone = row[phone] ;       // จับค่าของ เบอร์โทร ไปใส่

    // จับเนื้อหาในไฟล์ HTML ที่เป็น Template
    var htmlMessage = emailTemp.evaluate().getContent();

    GmailApp.sendEmail(
      row[email] ,
      subject ,
      "อีเมลของคุณไม่สนับสนุน HTML" ,
      {name : name, htmlBody:htmlMessage}
    ); // End – sendMail
  }); // End - forEach
}

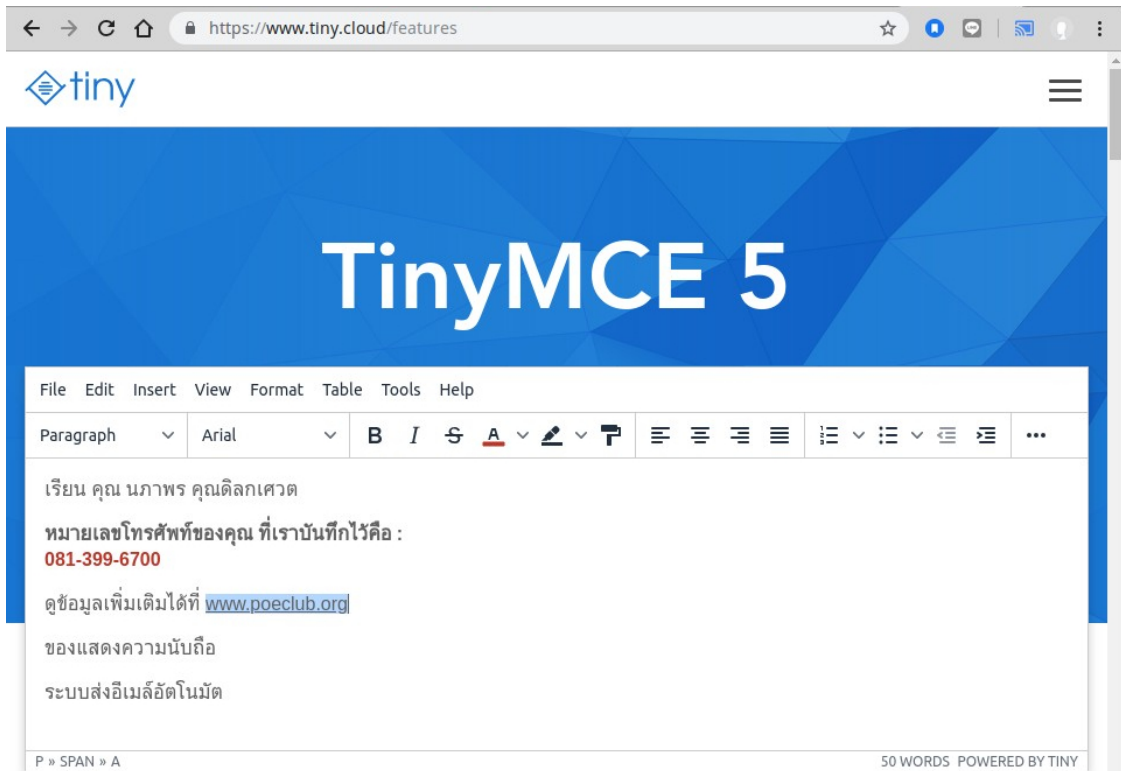
```

หาก Log ดูข้อมูลที่จับมาได้ ที่กรองแล้ว ก่อนส่งอีเมล จะเห็นว่า เหลือเฉพาะบรรทัด Checkbox ถูก Check เท่านั้น

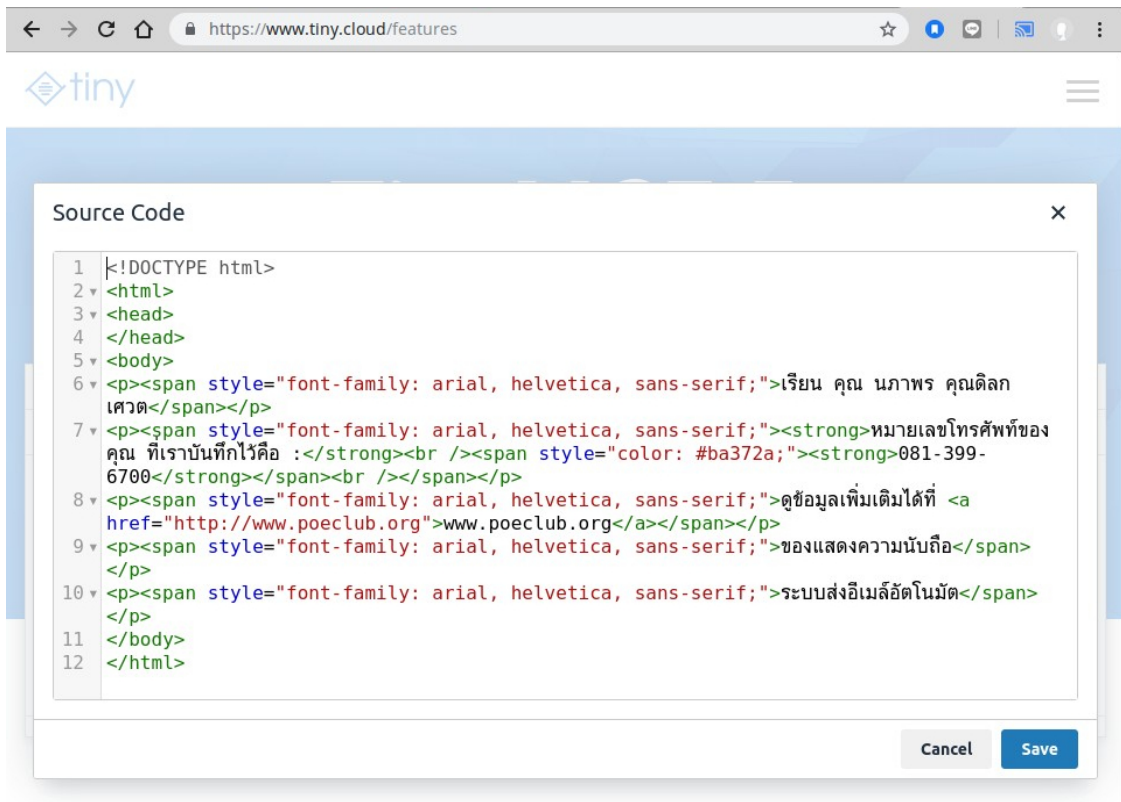
```
[ 01 ] [[wasan@snpfood.com, wasan, snp, 095-222-2222, true]]
```

### 3.10. พัฒนาการที่ 9 : เครื่องมือช่วยสร้างโค้ด HTML

ไปที่เว็บ <https://www.tiny.cloud/features> (ที่หน้าแรกของเว็บเลือกเมนู TinyMCE) มีเครื่องมือช่วยสร้างหน้าเว็บ จากการพิมพ์ข้อความ จัดรูปแบบ สร้างลิงค์ แบบ WYSIWUG จากนั้นระบบจะสร้างเป็นโค้ด HTML ให้



คลิกที่ปุ่ม 3 จุด → Source Code เพื่อดู Source Code





บทที่ 4  
โปรเจ็ค 04  
ส่งอีเมลล์ขอบคุณ  
เมื่อกรอกแบบฟอร์ม



## 4.1. ที่มา

Google Forms - Email Notification Script - Send Confirmation Emails to Users

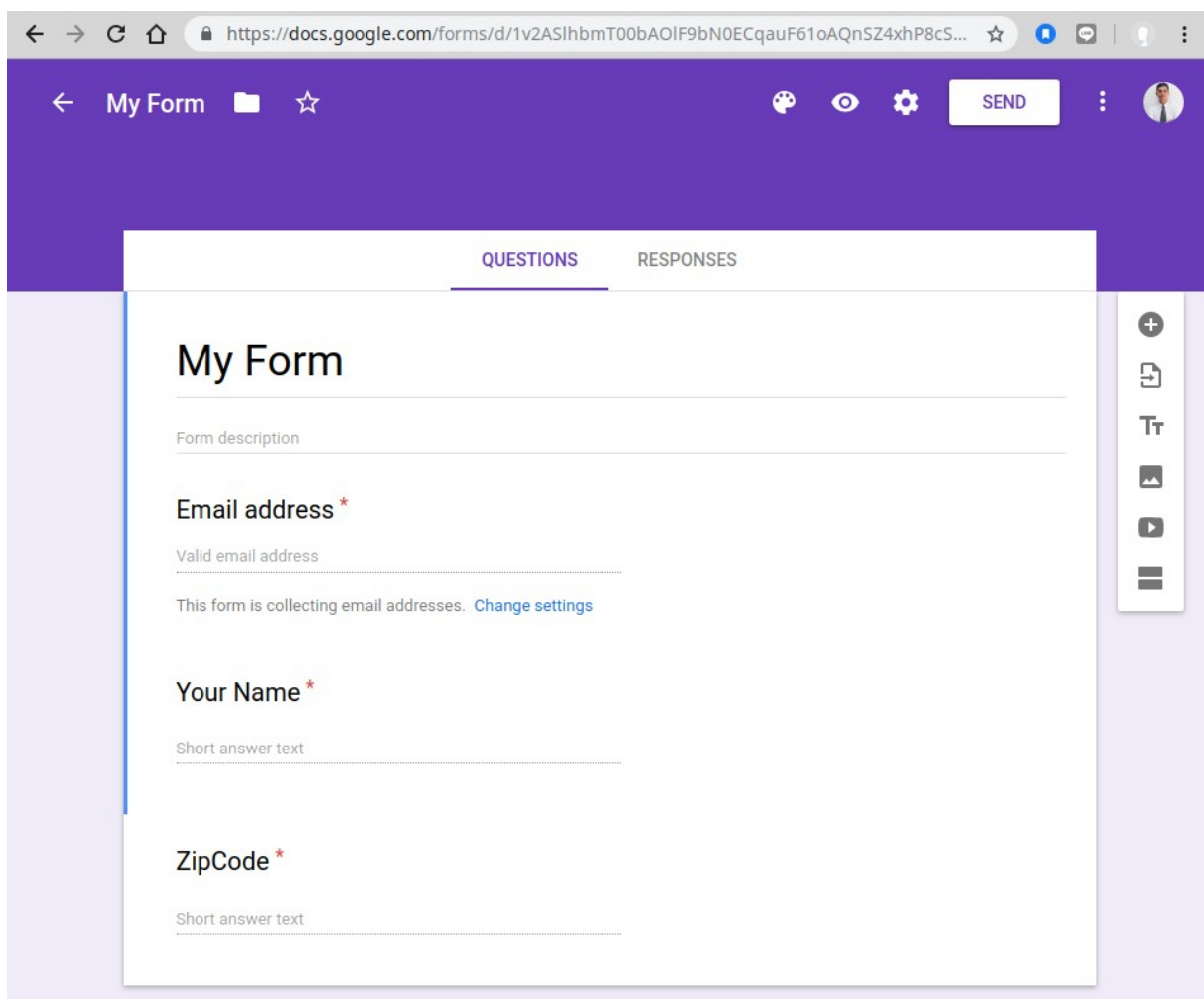
<https://www.youtube.com/watch?v=H7WFkt6J4rs&t=15s>

โปรเจกต์นี้ จะส่งอีเมลยืนยันกลับไปยังผู้กรอกแบบฟอร์ม ใน Google Form โดยอีเมลที่ส่งไปจะใช้ Template จากไฟล์ HTML ที่เราสร้างเอง

## 4.2. เตรียมพร้อม

### 4.2.ก.) สร้าง Form

สร้าง Form ตามภาพ มีการเก็บอีเมล ชื่อผู้ตอบแบบฟอร์ม และ รหัสไปรษณีย์



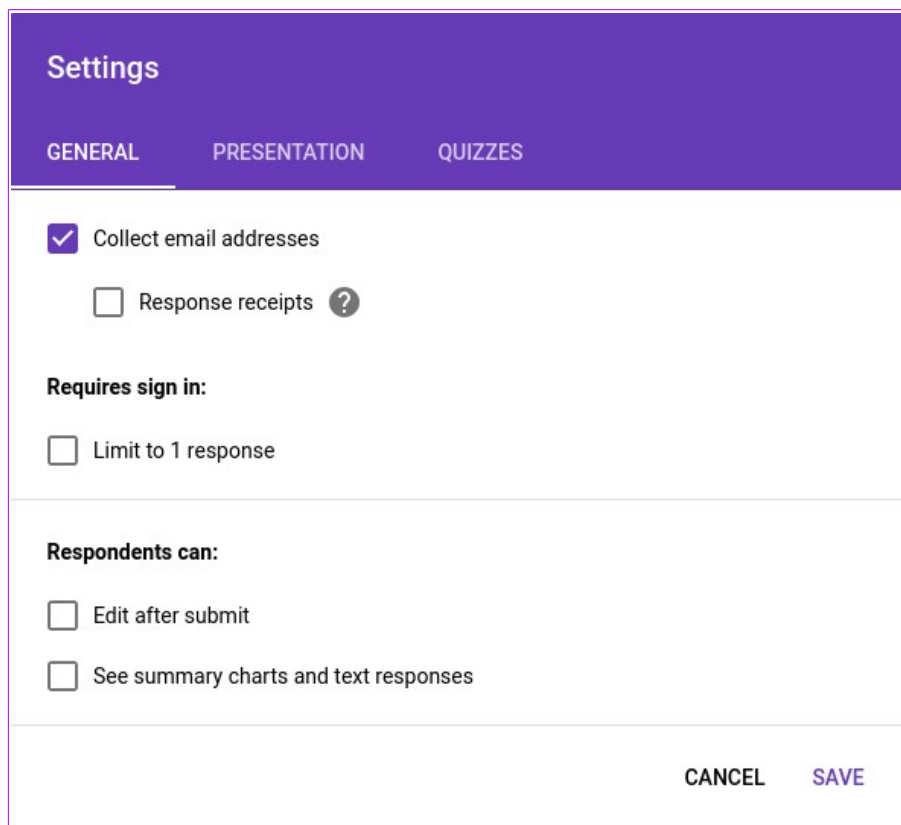
The screenshot shows a Google Form titled "My Form" in the "QUESTIONS" tab. The form has a purple header with a "SEND" button and a user profile icon. The form fields are:

- Email address \***: A text field with a placeholder "Valid email address". Below it, a message states "This form is collecting email addresses. [Change settings](#)".
- Your Name \***: A text field with a placeholder "Short answer text".
- ZipCode \***: A text field with a placeholder "Short answer text".

On the right side of the form, there is a vertical toolbar with icons for adding questions, duplicating, deleting, and other editing functions.



ฟิลด์แรกหรือช่องแรก ก็คือ Email address เป็นการเปิดใช้งานการเก็บอีเมล สามารถเปิดใช้งานฟิลด์(หรือ Item) นี้ได้โดย คลิกที่ปุ่มเฟือง จะปรากฏหน้าต่าง Settings ตามภาพ จากนั้น ดึงที่ collect email addresses

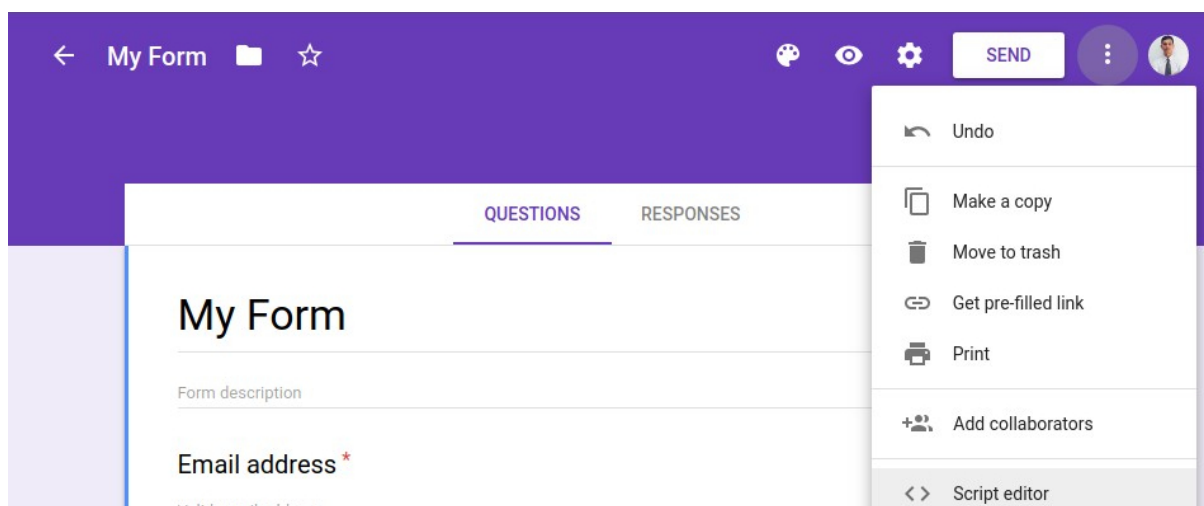


The screenshot shows the 'Settings' dialog box for a Google Form, specifically the 'GENERAL' tab. The 'Collect email addresses' option is checked. Other options include 'Response receipts' (unchecked), 'Requires sign in:' (with a sub-option 'Limit to 1 response' unchecked), and 'Respondents can:' (with sub-options 'Edit after submit' and 'See summary charts and text responses', both unchecked). At the bottom right are 'CANCEL' and 'SAVE' buttons.

#### 4.2.ข.) สร้างโปรเจ็ค Google Apps Script สำหรับ Form

สร้างโปรเจ็ค Google Apps Script แบบฝังกับไฟล์ Google Form

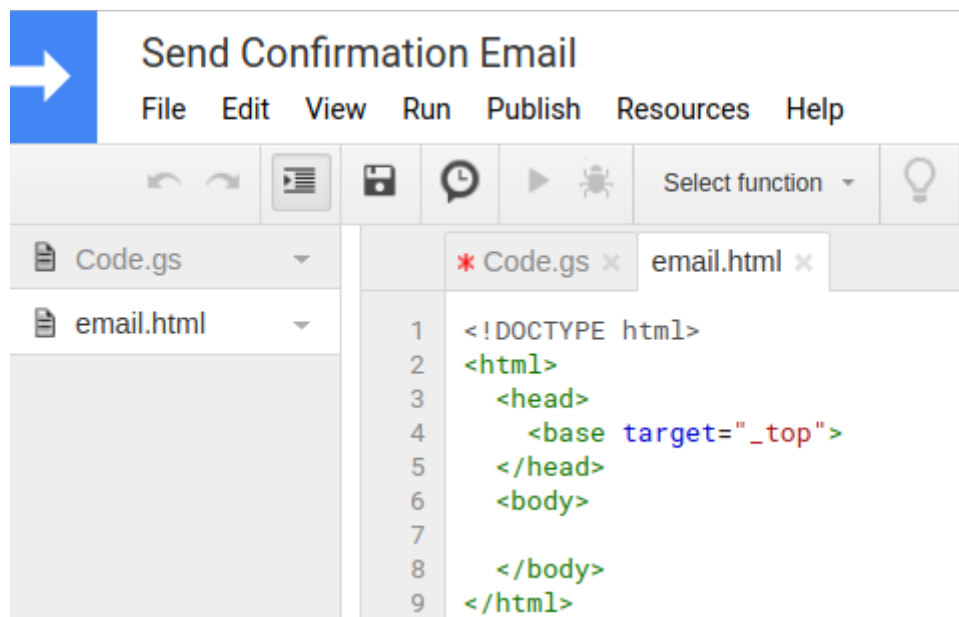
ที่หน้า Edit Form (สร้างหรือแก้ไขฟอร์ม) คลิกที่ปุ่ม 3 จุด → Script Editor จะเข้าสู่หน้าการสร้างโปรเจ็ค Google Apps Script สำหรับ Form



The screenshot shows the 'My Form' edit page in Google Forms. The 'QUESTIONS' tab is active, showing a form titled 'My Form' with a description field and an 'Email address' field with a red asterisk. A menu is open on the right, showing options: 'Undo', 'Make a copy', 'Move to trash', 'Get pre-filled link', 'Print', 'Add collaborators', and 'Script editor' (highlighted).

#### 4.2.ค.) สร้างไฟล์ email.html ในโปรเจ็ค Google Apps Script

ที่หน้าโปรเจ็ค Google Apps Script สร้างไฟล์ email.html เพิ่มเติม โดยไปที่ New → HTML โดยไฟล์นี้เราจะใช้เป็น Template ในการส่งอีเมล



เท่านี้ เราก็พร้อมเขียนโค้ดต่างๆแล้ว

#### 4.3. พัฒนาการที่ 1 : ทดสอบส่งอีเมลจาก Tempalte

ไฟล์ email.html - สร้างเนื้อหาที่จะใช้เป็น Template ในการส่งอีเมล อย่างคร่าวๆ

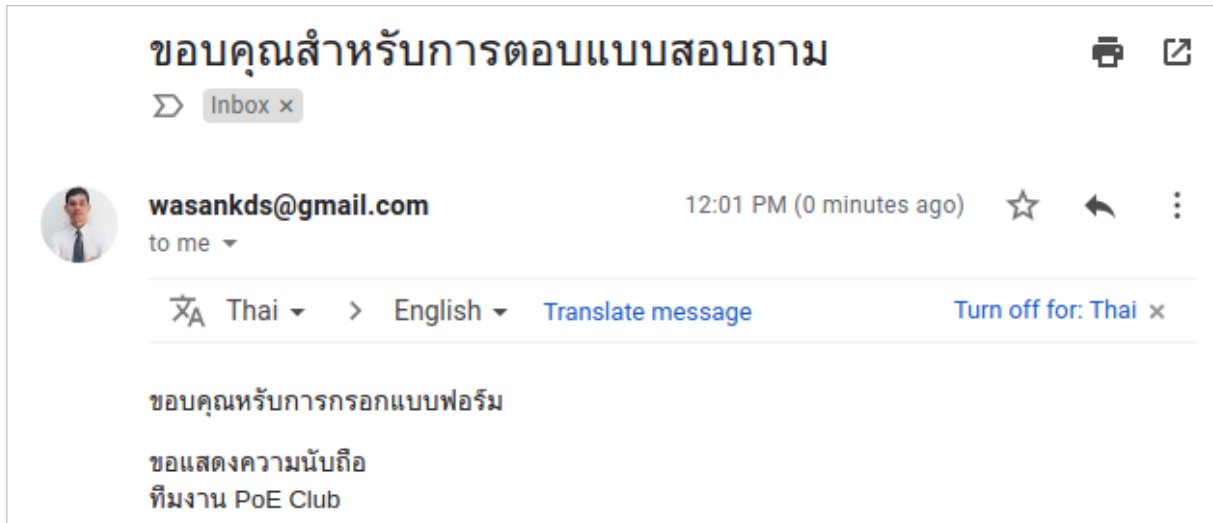
```
<!DOCTYPE html>
<html>
  <head>
    <base target="_top">
  </head>
  <body>

    <p>ขอบคุณสำหรับการกรอกแบบฟอร์ม</p>
    <p>ขอแสดงความนับถือ<br>
    ทีมงาน PoE Club
    </p>
  </body>
</html>
```

```
function sendEmail() {  
    var html = HtmlService.createTemplateFromFile("email") ;  
    // Logger.log(typeof html);      // object  
    // Logger.log(html);             // {}  
  
    var htmlText = html.evaluate().getContent() ;  
    // Logger.log(typeof htmlText);   // string  
    // Logger.log(htmlText);         // โค้ด HTML  
  
    var emailTo = "wasankds.poeclub@gmail.com" ; // ทดสอบส่งไปก่อน  
    var subject = "ขอบคุณสำหรับการตอบแบบสอบถาม" ;  
    var textBody = "เมลล์ฉบับนี้ต้องการ HTML Support  
                    โปรดทำให้แน่ใจว่าคุณได้เปิดการสนับสนุนนี้ให้กับยูสเซอร์" ;  
    var options = { htmlBody: htmlText } ;  
                    // ผู้รับ, ชื่ออีเมล, เนื้อหา, ออฟชั่นพิเศษ  
    GmailApp.sendEmail(emailTo, subject, textBody, options) ;  
}
```

ผล หลังรันฟังก์ชัน sendEmail

ผู้รับได้รับอีเมลตามภาพ เนื้อหาเป็น Template ที่มาจากไฟล์ email.html



## 4.4. พัฒนาการที่ 2 : รับค่าจาก Form submit

ไฟล์ Code.gs - เขียนโค้ดเพิ่มเติม เพื่อส่ง Argument **e** ให้ฟังก์ชัน เพื่อนำไปใช้ในการส่งอีเมล

```
function sendEmail(e) {  
  
    var html = HtmlService.createTemplateFromFile("email") ;  
    var htmlText = html.evaluate().getContent() ;  
  
    Logger.log(typeof e);           // พิมพ์ : object  
    Logger.log(e);                 // พิมพ์ : [object Object]  
    var emailTo = e.response.getResponseEmail();  
    Logger.log(emailTo);           // พิมพ์อีเมล เช่น wasankds.poeclub@gmail.com  
  
    var subject = "ขอบคุณสำหรับการตอบแบบสอบถาม" ;  
  
    var textBody = "เมลฉบับนี้ต้องการ HTML Support  
                    โปรดทำให้แน่ใจว่าคุณได้เปิดการสนับสนุนนี้ให้กับยูสเซอร์" ;  
  
    var options = { htmlBody: htmlText } ;  
  
    if(emailTo !== undefined){  
        GmailApp.sendEmail(emailTo, subject, textBody, options) ;  
    }  
}
```

โค้ดข้างต้น ถ้ารันปกติโดยการ **<Ctrl><R>** จะ Error เพราะ Argument **e** ที่ส่งให้ฟังก์ชันเป็น **undefined** (ไม่มีอะไรส่งมา)

Argument **e** ที่จะส่งไปในฟังก์ชัน **sendEmail()** เตรียมไว้สำหรับ Trigger ที่จะส่งรันฟังก์ชันเมื่อมี Event เกิดขึ้น ในที่นี้ก็คือ Event(เหตุการณ์) Form Submit (คลิกปุ่ม Submit เพื่อส่งแบบฟอร์ม)

เมื่อมี Event Form Submit จะกระตุ้นให้รันฟังก์ชัน แต่ต้องไปสร้าง Trigger ก่อน ( ดูเพิ่มเติมข้อ 4.5 พัฒนาการที่ 3 : สร้าง Trigger เพื่อรันฟังก์ชันตาม Event หน้า 45 )

เมื่อมี Event Form Submit ฟอร์มจะส่งผ่านวัตถุหรือคลาส **FromResponse** มาด้วย ซึ่งก็คือ Argument **e** ที่จะถูกส่งให้กับฟังก์ชัน **sendEmail** เพื่อนำไปประมวลผลต่อ

วัตถุ **FromResponse** มี Keys อะไรบ้างดูได้จากลิงค์นี้

( [https://developers.google.com/apps-script/guides/triggers/events#google\\_forms\\_events](https://developers.google.com/apps-script/guides/triggers/events#google_forms_events) )

( <https://developers.google.com/apps-script/reference/forms/form-response> )

บรรทัด

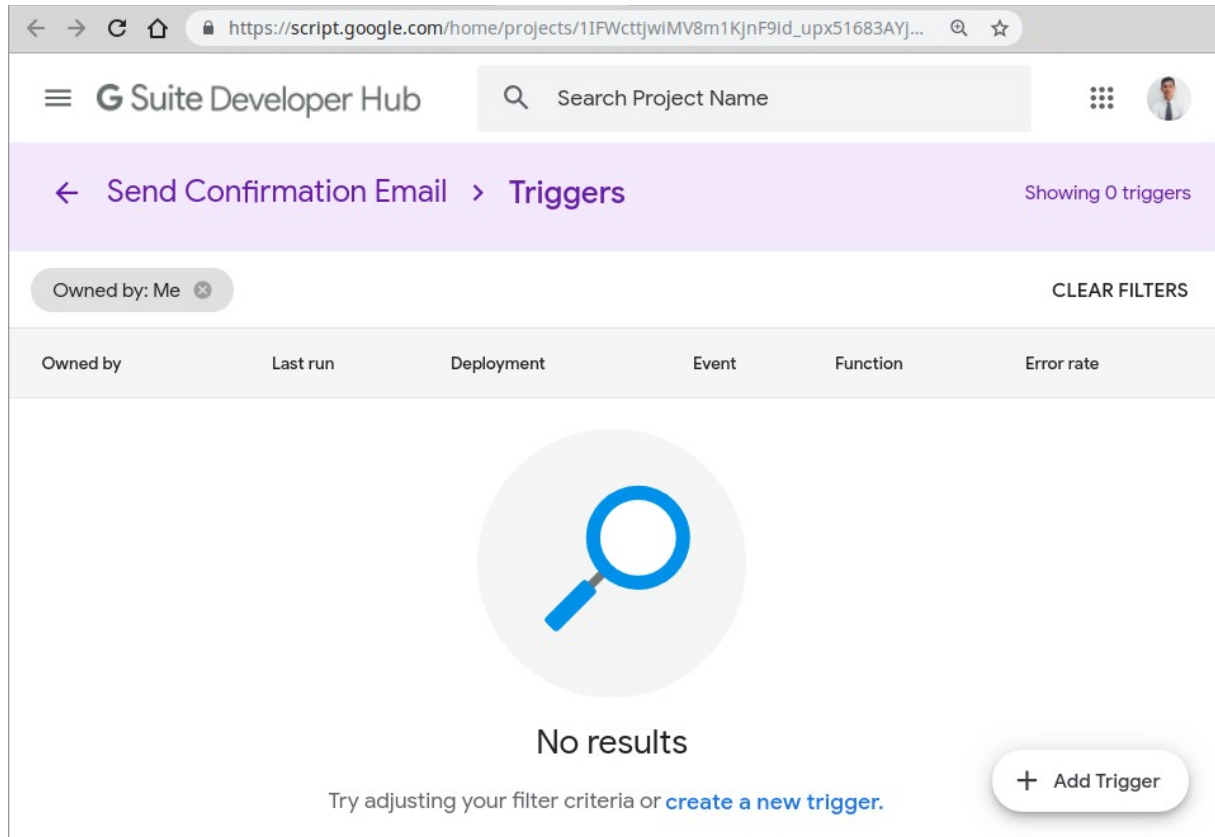
```
var emailTo = e.response.getResponseEmail();
```

ใช้จับอีเมลที่กรอกลงในแบบฟอร์ม ค่าที่คืนกลับมาเป็นอีเมล เช่น wasankds.poeclub@gmail.com

## 4.5. พัฒนาการที่ 3 : สร้าง Trigger เพื่อรันฟังก์ชันตาม Event

ในขั้นตอนนี้ จะเป็นการสร้าง Trigger (ตัวกระตุ้น) เพื่อรันฟังก์ชันตาม Event

ที่หน้าโปรเจกต์ Google Apps Script ไปที่เมนู Edit → Current Project's Triggers จะเข้าสู่หน้า Triggers ตามภาพ



สร้าง Trigger โดยคลิกที่ปุ่ม Add Trigger จะปรากฏหน้าต่างมาให้ตั้งค่า Trigger ในการรันโค้ด

ช่อง Choose which function to run

ใช้เลือกฟังก์ชันในโปรเจกต์ที่จะรัน

ช่อง Select event source

ใช้เลือกว่าแหล่งของ Event จะมาจากไหน ในที่นี้เลือก Form ก็คือ มาจาก From

ช่อง Select event type

ใช้เลือก Event ในที่นี้เลือก On form submit ก็คือ เมื่อคลิกที่ปุ่ม Submit เพื่อส่งคำตอบในการกรอกแบบฟอร์ม

## Add Trigger for Send Confirmation Email

Choose which function to run

sendEmail

Choose which deployment should run

Head

Select event source

From form

Select event type

On form submit

Failure notification settings

+

Notify me daily

Cancel

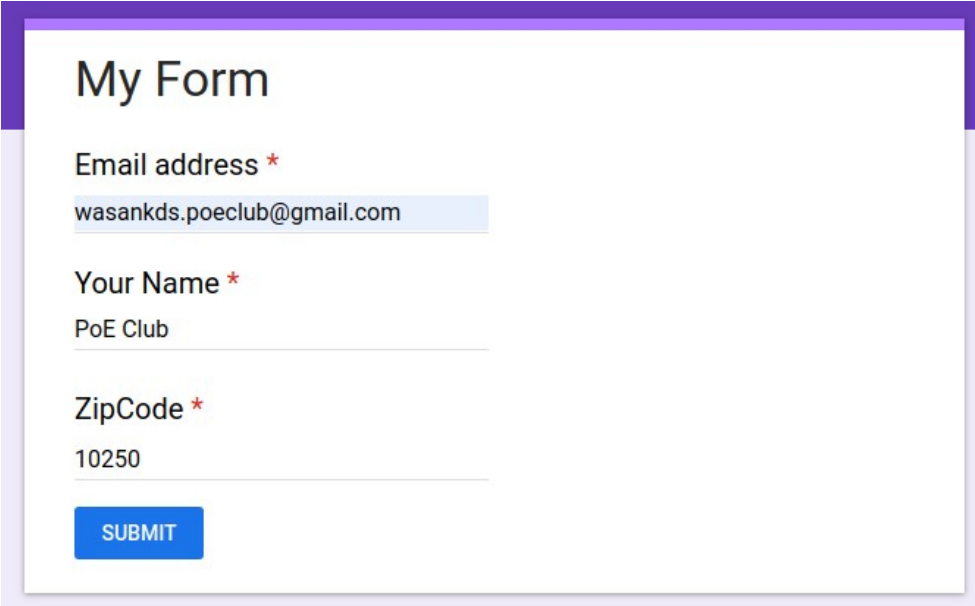
Save

เมื่อ Save แล้วจะได้ Trigger ตามภาพ

<a href="#">← Send Confirmation Email</a> <a href="#">Triggers</a> <span>Showing 1 trigger</span>					
Owned by: Me <span>×</span> <span>CLEAR FILTERS</span>					
Owned by	Last run	Deployment	Event	Function	Error rate
Me	-	Head	From form - On form submit	sendEmail	-

## 4.6. ทดสอบกรอกแบบฟอร์ม

เมื่อกรอกแบบฟอร์ม จากนั้นคลิกที่ปุ่ม **SUBMIT** จะไปรันฟังก์ชัน `sendEmail` ซึ่งจะมีการส่งอีเมลไปยังอีเมลที่กรอกลงในช่อง `Email address`

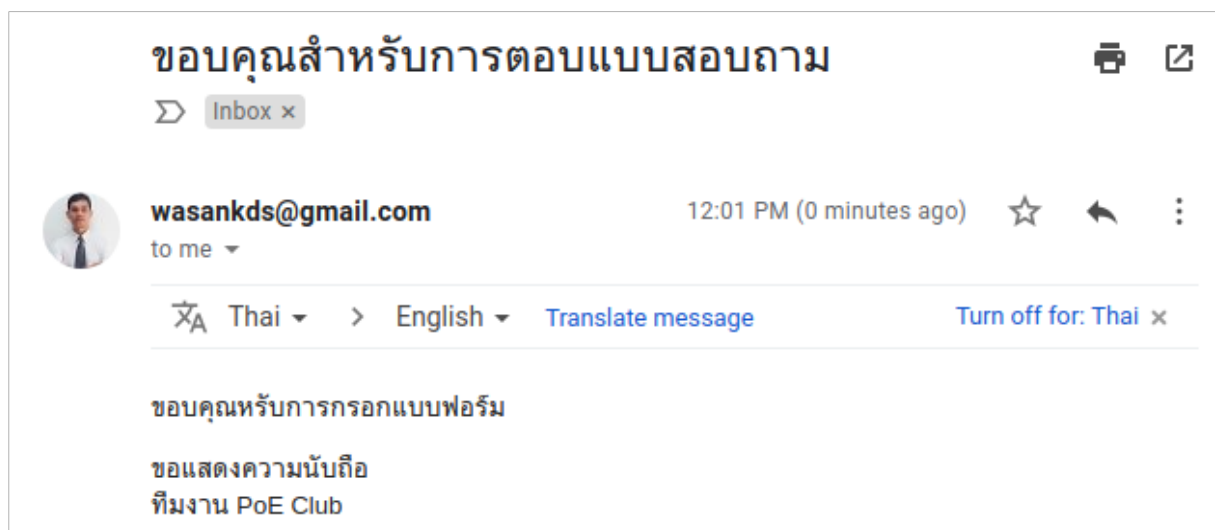


The screenshot shows a web form titled "My Form" with a purple border. It contains three input fields, each with a red asterisk indicating it is required:

- Email address \***: The input field contains the text "wasankds.poelclub@gmail.com".
- Your Name \***: The input field contains the text "PoE Club".
- ZipCode \***: The input field contains the text "10250".

Below the input fields is a blue button labeled "SUBMIT".

ผู้กรอกแบบฟอร์มจะได้รับอีเมล ตัวอย่างดังต่อไปนี้







บทที่ 5  
โปรเจ็ค 05  
สร้างฟอร์มให้ Dropdown  
ใน Form จากข้อมูล  
ใน Google Sheet



## 5.1. ที่มา

Google Forms - Drop Down List from Spreadsheet Using Apps Script \*\*\*

[https://www.youtube.com/watch?v=o3AL7ASI\\_cA](https://www.youtube.com/watch?v=o3AL7ASI_cA)

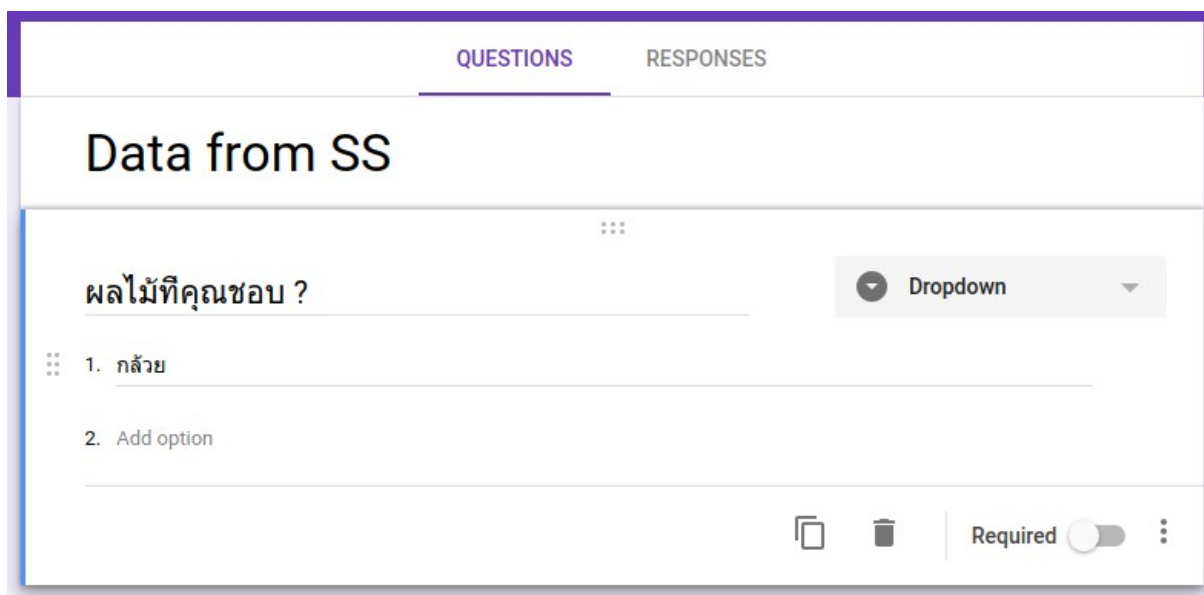
Apps Script: Dynamic Forms Multiple Choice

[https://www.youtube.com/watch?v=MPIT\\_sIWL6k](https://www.youtube.com/watch?v=MPIT_sIWL6k)

สร้างและอัปเดตช้อยส์ให้กับ Item หรือ คำถาม ที่เป็น Dropdown โดยจับข้อมูลจากใน Google Sheet มาใช้เป็นช้อยส์

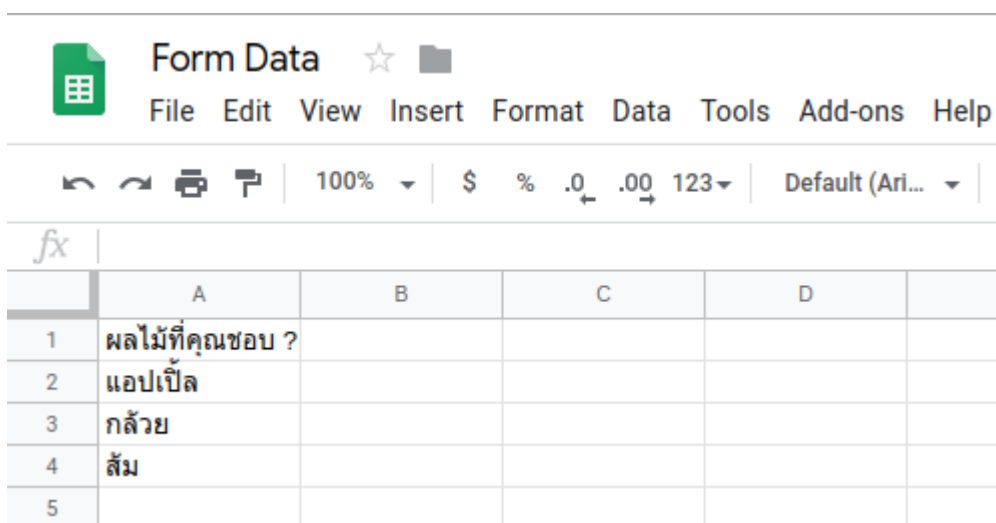
## 5.2. เตรียมพร้อม

สร้าง Form ดังต่อไปนี้ - มี Item ที่เป็น Dropdown ตามภาพ



The screenshot shows a Google Form interface with two tabs: 'QUESTIONS' and 'RESPONSES'. The 'QUESTIONS' tab is active. The form title is 'Data from SS'. Below the title, there is a question 'ผลไม้ที่คุณชอบ ?' (Fruit you like?). To the right of the question is a dropdown menu icon and the text 'Dropdown'. Below the question, there are two options: '1. กล้วย' (1. Banana) and '2. Add option'. At the bottom right of the question area, there are icons for 'Required' (a toggle switch) and a three-dot menu icon.

สร้างไฟล์ Google Sheet - พิมพ์ข้อมูลลงไป



	A	B	C	D
1	ผลไม้ที่คุณชอบ ?			
2	แอปเปิ้ล			
3	กล้วย			
4	ส้ม			
5				

และที่ไฟล์ Google Sheet ไฟล์นี้ สร้างโปรเจ็ค Google Apps Script แบบฝังในไฟล์ (ไปที่เมนู Tools → Script Editor)

### 5.3. พัฒนาการที่ 1 : หา ID ของ Item

Items(คำถาม) ที่อยู่ใน Form มี ID ประจำตัว ก่อนอื่นเราต้องหา ID ของ Items มาก่อน เพื่อจะได้จับไปที่ Item ตัวนั้นได้

เริ่มต้นเราจะหา Item ทั้งหมดก่อน

```
var ssID = "1dqs-M-llGZXE8yYvtPCALu-dz2UV3FFWGNtwYosGUUO" ; // ID ของไฟล์ Google Sheet
var formID = "1CA7J-OU6wCD6RxdZYbomggclAeDiOtbMkboYFe7RALE" ; // ID ของไฟล์ Google Form

var wsData = SpreadsheetApp.openById(ssID).getSheetByName("Data"); // จับไปที่ชีทที่มีข้อมูล
var form = FormApp.openById(formID); // จับไป Form

function myFunction() {

  var items = form.getItems();

  Logger.log(items); // พิมพ์ : [item] - มีอยู่ 1 item // ดูต่อใน Logs - >
  Logger.log(typeof items); // พิมพ์ : object // ดูต่อใน Logs - >

  for(i=0 ; i < items.length ; i++) {

    Logger.log(items[i].getId().toString()); // พิมพ์ : ตัวเลข ID // ดูต่อใน Logs - >

  } ;
}
```

ผล

#### Logs

```
[ 01 ] [Item] - มีอยู่ 1 ดูตามสมาชิกอาเรย์
[ 02 ] object
[ 03 ] 1469553275
```

### 5.4. พัฒนาการที่ 2 : จับคุณสมบัติต่างๆของ Item มา Log ดู

จับคุณสมบัติต่างๆของ Item มา Log ดู

```
var ssID = "1dqs-M-llGZXE8yYvtPCALu-dz2UV3FFWGNtwYosGUUO" ;
var formID = "1CA7J-OU6wCD6RxdZYbomggclAeDiOtbMkboYFe7RALE" ;

var wsData = SpreadsheetApp.openById(ssID).getSheetByName("Data");
var form = FormApp.openById(formID);

function myFunction() {

  // จับ Item ตาม ID
  var items = form.getItemById(1469553275) ;
  Logger.log(items.getTitle()) ; // ดูต่อใน Logs - >
  Logger.log(items.getType()) ; // ดูต่อใน Logs - >

}
```

ผล

#### Logs

[ 01 ] ผลไม้ที่คุณชอบ ?

[ 02 ] LIST

## 5.5. พัฒนาการที่ 3 : ใส่ข้อมูลให้กับคำถาม

Item ที่เราจับมาได้ มี Type เป็น List ฉะนั้นเราจะใช้ Method `asListItem()` ต่อท้าย เพื่อดึง Method ที่เกี่ยวกับ List ออกมาใช้ จากนั้น ก็ใส่ตัวเลือกต่างๆ ลงไปด้วย Method `setChoiceValues([Values])`

```
var ssID = "1dqs-M-lIGZXE8yYvtPCALu-dz2UV3FFWGNtwYosGUUO" ;
var formID = "1CA7J-OU6wCD6RxdZYbomggclAeDiOtbMkboYFe7RALE" ;

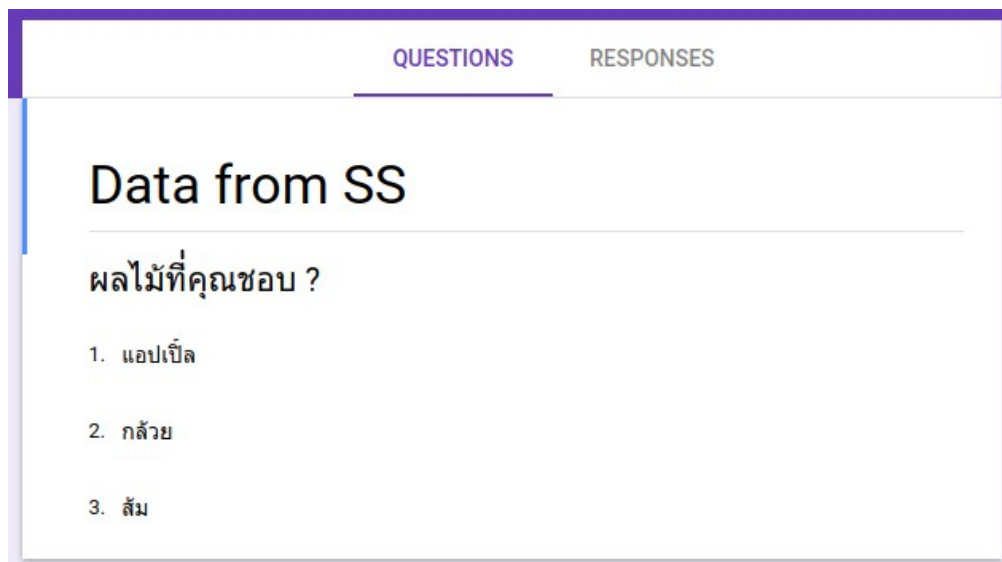
var wsData = SpreadsheetApp.openById(ssID).getSheetByName("Data");
var form = FormApp.openById(formID);

function myFunction() {

  // จับ Item ตาม ID
  var items = form.getItemById(1469553275) ;
  var values = [ "แอปเปิ้ล" , "กล้วย" , "ส้ม" ]

  item.asListItem().setChoiceValues(values);
}
```

ผล – ให้ไปดูที่ Form จะปรากฏข้อมูลของคำถามตามค่าตัวแปรในโค้ด



## 5.6. พัฒนาการที่ 4 : แยกฟังก์ชัน

### 5.6.ก.) แยกฟังก์ชันครั้งที่ 1

ขั้นตอนนี้จะเราจะแยกฟังก์ชันออกมา ฟังก์ชัน `updateDropdown` ใช้ใส่ข้อมูลให้กับคำถาม ที่มี Type เป็น List ฟังก์ชัน `myFunction` เป็นฟังก์ชันหลัก

myFunction ต้องการ ID ของ Items(หรือคำถาม) อีกครั้ง เพราะต้องใช้เป็น Argument ส่งให้ ฟังก์ชัน updateDropdown แต่ครั้งนี้จะใช้คนละวิธีในการหา

ครั้งนี้ จะหา ID ของ Item จากชื่อ Title เช่น ผลไม้ที่คุณชอบ ? เป็นต้น

```
var ssID = "1dqs-M-lIGZXE8yYvtPCALu-dz2UV3FFWGNtwYosGUUo" ;
var formID = "1CA7J-OU6wCD6RxdZYbomggcIAeDiOtMkboYFe7RALE" ;

var wsData = SpreadsheetApp.openById(ssID).getSheetByName("Data");
var form = FormApp.openById(formID);

function myFunction() {

  var title = "ผลไม้ที่คุณชอบ ?" ;

  var items = form.getItems() ;
  var titles = items.map(function(item){
    return item.getTitle();
  });

  // *****

  Logger.log(titles) ; // พิมพ์ : [ผลไม้ที่คุณชอบ ?, คำถามใหม่]

  // หาตำแหน่งลำดับตามชื่อ Title ในอาร์เรย์
  Logger.log(titles.indexOf(title)) ; // พิมพ์ : 0.0

  var pos = titles.indexOf(title) ; // จับ 0.0 ใส่ตัวแปร
  var item = items[pos] ; // จับ Item ค่าในอาร์เรย์ items ตำแหน่งลำดับที่ 0 ใส่ตัวแปร
  var itemID = item.getId() ; // จับ ID ของ Item ใส่ตัวแปร

  var values = [ "A" , "B" , "C" ]

  updateDropdown(itemID,values) ;

  // *****
}

// ใส่ข้อมูลให้กับคำถาม ที่มี Type เป็น List
function updateDropdown(id,values) {
  var item = form.getItemById(id) ;
  item.asListItem().setChoiceValues(values) ;
}
```

ผล

ผลไม้ที่คุณชอบ ?

1. A
2. B
3. C

## 5.6.ข.) แยกฟังก์ชันครั้งที่ 2

เราจะปรับโค้ดอีกครั้ง โดยแยกฟังก์ชันอีก เพื่อให้โค้ดอ่านง่าย

```
var ssID = "1dqs-M-IlGZXE8yYvtPCALu-dz2UV3FFWGNTwYosGUUO" ;
var formID = "1CA7J-OU6wCD6RxdZYbomggclAeDiOtBmkboYFe7RALE" ;

var wsData = SpreadsheetApp.openById(ssID).getSheetByName("Data");
var form = FormApp.openById(formID);

function main(){
  var title = "ผลไม้ที่คุณชอบ ?" ;
  var values = [ "X" , "Y" , "Z" ] ;
  updateDropDownUsingTitle(title,values) ;
}

function updateDropDownUsingTitle(title,values) {

  var title = "ผลไม้ที่คุณชอบ ?" ;
  var items = form.getItems() ;
  var titles = items.map(function(item){
    return item.getTitle();
  });

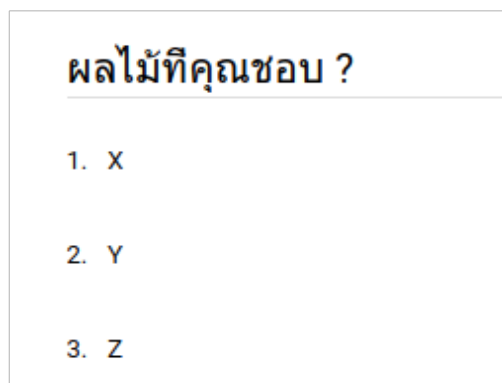
  var pos = titles.indexOf(title) ;           // จับ 0.0 ใส่ตัวแปร
  var item = items[pos] ;                     // จับ Item ค่าในอาร์เรย์ items ดรรชนีลำดับที่ 0 ใส่ตัวแปร
  var itemID = item.getId() ;                 // จับ ID ของ Item (ใน Form) ใส่ตัวแปร

  var values = [ "A" , "B" , "C" ]

  updateDropdown(itemID,values) ;
}

function updateDropdown(id,values) {
  var item = form.getItemById(id) ;
  item.asListItem().setChoiceValues(values) ;
}
```

ผล



ผลไม้ที่คุณชอบ ?

1. X
2. Y
3. Z

## 5.7. แก้ไข Form และ Google Sheet

ก่อนจะไปต่อ ให้แก้ไข Form โดยเพิ่ม Item ใน Form และ เพิ่มข้อมูลอีก 1 คอลัมน์ใน Google Sheet สังเกตว่า คำถามที่ Form กับที่ Google Sheet ข้อความเหมือนกันทุกประการ (Copy and Paste)

แก้ไข Form - ให้เป็นดังนี้

QUESTIONS

RESPONSES

### Data from SS

ผลไม้ที่คุณชอบ ?

- X
- Y
- Z

คำถามใหม่

- Option 1

แก้ไขข้อมูลใน Google Sheet – ให้เป็นดังนี้ มี 2 คำถาม ใน 2 คอลัมน์

	A	B	C
1	ผลไม้ที่คุณชอบ ?	คำถามใหม่	
2	แอปเปิ้ล	ใช่	
3	กล้วย	ไม่ใช่	
4	ส้ม		

## 5.8. พัฒนาการที่ 5 : จับข้อมูลใน Google Sheet ไปสร้าง Item ใน From

### 5.8.ก.) จับแถวคำถามมา Log ดู

ขั้นตอนนี้เราจะจับแถวคำถามใน Google Sheet มา Log ดูก่อน เฉพาะที่ฟังก์ชัน Main เขียนโค้ดดังนี้

```
var ssID = "1dqs-M-lIGZXE8yYvtPCALu-dz2UV3FFWGNtwYosGUU0" ;
var wsData = SpreadsheetApp.openById(ssID).getSheetByName("Data");

function main(){

// จับ 1 มิติ
var labels = wsData.getRange(1,1,1,wsData.getLastColumn()).getValues()[0];
  Logger.log(labels);
}
```

ผล

```
[ 01 ] [ผลไม้ที่คุณชอบ ?, คำถามใหม่]
```

### 5.8.ข.) จับแถวคำถามมา Log ดู - ดูข้อมูลเพิ่ม

ใช้ forEach วนเข้าไปดูค่าและตรรกะลำดับของ ข้อมูลที่จับมาจาก Google Sheet

```
var ssID = "1dqs-M-lIGZXE8yYvtPCALu-dz2UV3FFWGNtwYosGUUO" ;
var wsData = SpreadsheetApp.openById(ssID).getSheetByName("Data");

function main(){

// จับ 1 มิติ
var labels = wsData.getRange(1,1,1,wsData.getLastColumn()).getValues()[0];

labels.forEach(function(label,i){
  Logger.log(label);      // Value
  Logger.log(i);          // Index
});
}
```

ผล

```
[ 01 ] ผลไม้ที่คุณชอบ ?
[ 02 ] 0.0
[ 03 ] คำถามใหม่
[ 04 ] 1.0
```

### 5.8.ค.) จับข้อมูลมา Log ดู

เราจับส่วนที่เป็นคำถามได้แล้ว ถัดมาจะเป็นการจับไปที่ข้อมูลของแต่ละคำถาม

```
var ssID = "1dqs-M-lIGZXE8yYvtPCALu-dz2UV3FFWGNtwYosGUUO" ;
var wsData = SpreadsheetApp.openById(ssID).getSheetByName("Data");

function main(){

// จับ 1 มิติ
var labels = wsData.getRange(1,1,1,wsData.getLastColumn()).getValues()[0];

labels.forEach(function(label,i){
  Logger.log(label);      // Value
                          // วนจับทีละคอลัมน์
  var options = wsData.getRange(2,i+1,wsData.getLastRow()-1,1).getValues();
  Logger.log(options);    // ข้อมูล
});
}
```

ผล

```
[ 01 ] ผลไม้ที่คุณชอบ ?
[ 02 ] [[แอปเปิ้ล], [กล้วย], [ส้ม]]
[ 03 ] คำถามใหม่
[ 04 ] [[ใช่], [ไม่ใช่], []]
```



### 5.8.ง.) จับข้อมูลมา Log ดู

เนื่องจากข้อมูลที่จับมาเป็นอาร์เรย์ 2 มิติ เราจะทำให้เหลือแค่ 1 มิติ

```
var ssID = "1dqs-M-lIGZXE8yYvtPCALu-dz2UV3FFWGNtwYosGUU0" ;
var wsData = SpreadsheetApp.openById(ssID).getSheetByName("Data");

function main(){
    // จับ 1 มิติ
    var labels = wsData.getRange(1,1,1,wsData.getLastColumn()).getValues()[0];

    labels.forEach(function(label,i){
        Logger.log(label);           // Value
                                    // วนจับทีละคอลัมน์

        var options = wsData
            .getRange(2,i+1,wsData.getLastRow()-1,1)
            .getValues()
            .map(function(o){ return o[0]}); // เอาเฉพาะสมาชิกตัวแรก
        Logger.log(options);         // ข้อมูล
    });
}
```

ผล

```
[ 01 ] ผลไม้ที่คุณชอบ ?
[ 02 ] [แอปเปิ้ล, กล้วย, ส้ม]
[ 03 ] คำถามใหม่
[ 04 ] [ใช่, ไม่ใช่, ] - มีค่าว่างอยู่ เดี่ยวไปว่ากันทีหลัง
```

### 5.8.จ.) จับข้อมูลใน Google Sheet ไปสร้าง Item ใน Form

ถึงเวลาจับข้อมูลใน Google Sheet ไปสร้าง Item ใน Form ก็เปลี่ยนจาก Log เป็นเรียกใช้ฟังก์ชัน `updateDropDownUsingTitle`

```
var ssID = "1dqs-M-lIGZXE8yYvtPCALu-dz2UV3FFWGNtwYosGUU0" ;
var wsData = SpreadsheetApp.openById(ssID).getSheetByName("Data");

function main(){
    // จับ 1 มิติ
    var labels = wsData.getRange(1,1,1,wsData.getLastColumn()).getValues()[0];

    labels.forEach(function(label,i){
        Logger.log(label);           // Value
                                    // วนจับทีละคอลัมน์

        var options = wsData
            .getRange(2,i+1,wsData.getLastRow()-1,1)
            .getValues()
            .map(function(o){ return o[0]}); // เอาเฉพาะสมาชิกตัวแรก
        updateDropDownUsingTitle(label,options); // ใส่ข้อมูลใน Form
    });
}
```

ผล – สังเกตว่าที่ค่าว่างในอาเรย์ [ใช่, ไม่ใช่, ] (ตัวที่ 3 ว่าง) กลายเป็นข้อยส์ที่ไม่มีข้อความ

QUESTIONS	RESPONSES
<h2>Data from SS</h2> <p>ผลไม้ที่คุณชอบ ?</p> <ol style="list-style-type: none"><li>แอปเปิ้ล</li><li>กล้วย</li><li>ส้ม</li></ol> <p>คำถามใหม่</p> <ol style="list-style-type: none"><li>ใช่</li><li>ไม่ใช่</li><li></li></ol>	

### 5.8.จ.) กำจัดค่าว่างในอาเรย์ข้อยส์

ที่อาเรย์ของข้อยส์ ใช้ Method **filter** เพื่อกรองค่าว่างทิ้ง

```
var ssID = "1dqs-M-lIGZXE8yYvtPCALu-dz2UV3FFWGNtwYosGUUo" ;
var wsData = SpreadsheetApp.openById(ssID).getSheetByName("Data");

function main(){
    // จับ 1 มิติ
    var labels = wsData.getRange(1,1,1,wsData.getLastColumn()).getValues()[0];

    labels.forEach(function(label,i){
        Logger.log(label);           // Value
                                     // วนจับที่ละคอลัมน์

        var options = wsData
            .getRange(2,i+1,wsData.getLastRow()-1,1)
            .getValues()
            .map(function(o){ return o[0] });    // เอาเฉพาะสมาชิกตัวแรก
            .filter(function(o){ return o !== "" }); // กรองค่าว่างทิ้ง
        updateDropDownUsingTitle(label,options); // ใส่ข้อยส์ใน Form
    });
}
```

### 5.9. พัฒนาการที่ 6 : ป้องกัน Error

สุดท้ายจะเป็นการเขียนโค้ดป้องกัน Error กรณีคำถามใน Google Sheet ไม่ตรงกับใน Form หรือ ใน Form ไม่มีคำถามที่ตรงกับ Google Sheet

เช่น เพิ่มคำถามอีกคอลัมน์ใน Google Sheet ตามภาพ เมื่อรันฟังก์ชัน Main จะเกิด Error เพราะไม่มี Item ใน Form

	A	B	C
1	ผลไม้ที่คุณชอบ ?	คำถามใหม่	คำถามใหม่กว่า
2	แอปเปิ้ล	ใช่	
3	กล้วย	ไม่ใช่	
4	ส้ม	OK	
5	มะพร้าว	No K	

หากค่าที่ส่งไปหาดรชนีลำดับ ไม่มีในอาเรย์จะคืนค่า -1 กลับมา ตัวอย่างเช่น

```
function test(){
  var arr = [ "A" , "B" ];
  Logger.log(arr.indexOf("A")); // พิมพ์ : 0
  Logger.log(arr.indexOf("B")); // พิมพ์ : 1
  Logger.log(arr.indexOf("C")); // พิมพ์ : -1
}
```

เราเอาตรงนี้มาเขียน if เพื่อตรวจสอบก่อนทำโค้ด ฟังก์ชัน updateDropDownUsingTitle ให้แก้โค้ดเป็นดังนี้

```
function updateDropDownUsingTitle(title,values) {

  var title = "ผลไม้ที่คุณชอบ ?" ;
  var items = form.getItems() ;
  var titles = items.map(function(item){
    return item.getTitle();
  });

  var pos = titles.indexOf(title) ; // จับ 0.0 ใส่ตัวแปร

  if(pos !== -1){ // ถ้าดรชนีลำดับในอาเรย์ไม่เป็น -1 ก็ทำโค้ดด้านใน
    var item = items[pos] ; // จับ Item ค่าในอาเรย์ items ดรชนีลำดับที่ 0 ใส่ตัวแปร
    var itemID = item.getId() ; // จับ ID ของ Item (ใน Form) ใส่ตัวแปร

    updateDropdown(itemID,values) ;
  } // End if
}

function updateDropdown(id,values) {
  var item = form.getItemById(id) ;
  item.asListItem().setChoiceValues(values) ;
}
```



บทที่ 6  
โปรเจ็ค 06  
สร้าง Dropdown  
แบบ 3 ระดับ



## 6.1. ที่มา

Google Sheets - Dependent Drop Down List for Entire Column, Multiple Levels

<https://www.youtube.com/watch?v=s-I8Z4nTDak>

โปรเจกต์นี้ เราจะทำ Dropdown แบบ 3 ระดับ เช่น ระดับที่ 1 เป็นจังหวัด เราเลือกจังหวัดเพชรบูรณ์ ระดับที่ 2 ก็จะปรากฏอำเภอเฉพาะในจังหวัดเพชรบูรณ์ให้เลือก พอระดับที่ 2 เลือกอำเภอ เช่น หนองไผ่ ระดับที่ 3 ก็จะปรากฏตำบลในอำเภอหนองไผ่ให้เลือก

โปรเจกต์นี้ เราจะสร้างไฟล์ Google Apps Script แบบฝังลงในไฟล์ Google Sheet เพียงอย่างเดียว ไม่ใช้แอปอื่นๆ

โดย Google Sheet มี 2 ชีท

ชีท master – เอาไว้ทำสอบทำ Validation

ชีท options – เก็บข้อมูลเอาไว้เป็น List สำหรับทำ Dropdown

	A	B	C
1	ภาค	จังหวัด	พนักงานขาย
2	กลาง	กรุงเทพฯ	สมชาย
3	กลาง	กรุงเทพฯ	นภาพร
4	กลาง	ปทุมธานี	วสันต์
5	กลาง	ปทุมธานี	สมหวั
6	ใต้	ภูเก็ต	เสถียรพงษ์
7	ใต้	ภูเก็ต	กวี
8	ใต้	นราธิวาส	พรทิพย์
9	ใต้	ยะลา	สมศักดิ์
10	เหนือ	เชียงใหม่	สง่า
11	เหนือ	เชียงใหม่	สมหมาย
12	เหนือ	ลำปาง	นงลักษณ์
13	เหนือ	ลำปาง	สัมพันธ์
14	เหนือ	พิษณุโลก	พยุงค์ดี
15	อีสาน	ขอนแก่น	เล็ก
16	อีสาน	ขอนแก่น	สมพร

## 6.2. พัฒนาการที่ 1 : ทดสอบสร้าง Validation

### 6.2.ก.) ทดสอบสร้าง Validation

ทดสอบสร้าง Validation ให้กับเซลล์ B2 โดยใช้ Google Apps Script

โค้ดต่อไปนี้จะสร้าง Validation แบบ List ก็คือ รายการใน Dropdown เกิดจากการพิมพ์เอง เป็นค่าคงที่

```
function myFunction() {  
  var ws = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("master");  
  var list = [ "a" , "b" , "g" , "z" ] ;  
  
  var rule = SpreadsheetApp  
    .newDataValidation()           // สร้าง Validation ตัวใหม่  
    .requireValueInList(list)      // สร้าง List จากอาร์เรย์ list  
    .setAllowInvalid(false)        // Reject input  
    .build();                      // สร้าง Rule  
  
  ws.getRange("B2").setDataValidation(rule);  
}
```

	A	B
1		
2		
3		a
4		b
5		g
6		z
7		
8		

หากอาร์เรย์ `list` มีสมาชิกซ้ำกัน เช่นแก้เป็น [ "a" , "b" , "g" , "z" , "b" , "g" , "z" ] จะถูกกรองตัวซ้ำให้เหลือตัวเดียว

บรรทัด `.setAllowInvalid(false)` มีผลเท่ากับการเลือก Reject input ในหัวข้อ On invalid data ก็คือให้เลือกรายการเท่านั้น หากคีย์ค่าลงไปไม่ตรงกับที่มีในรายการ จะไม่รับข้อมูล

### Data validation

Cell range:

Criteria: 

List of items

☒ Show dropdown list in cell

On invalid data: ☐ Show warning ☒ Reject input

Appearance: ☐ Show validation help text:

Cancel

Remove validation

Save

## 6.2.ข.) สร้างเป็นฟังก์ชันสำหรับสร้าง Validation ให้กับเซลล์

ขั้นตอนนี้ จะแยกโค้ดก่อนหน้าออกเป็นฟังก์ชัน เพื่อจะสร้าง Validation ให้กับเซลล์ต่างๆตามที่เราต้องการได้ง่ายๆ

แก้ไขโค้ดเป็นดังนี้

```
function myFunction() {  
  
    var ws = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("master");  
    var list = [ "a" , "b" , "g" , "z" ] ;  
    var cell = ws.getRange("C2");  
  
    applyValidationToCell(list,cell)           // เรียกใช้ฟังก์ชัน applyValidationToCell  
}  
  
function applyValidationToCell(list,cell)() {  
  
    var rule = SpreadsheetApp  
                .newDataValidation()  
                .requireValueInList(list)  
                .setAllowInvalid(false)  
                .build();  
  
    cell.setDataValidation(rule);  
}
```

## 6.3. พัฒนาการที่ 2 : สร้างฟังก์ชัน onEdit

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

ในขั้นตอนนี้ เราจะสร้างอีก 1 ฟังก์ชัน ชื่อ onEdit ซึ่งเป็นฟังก์ชันที่จะรันอัตโนมัติเมื่อมี Event **แก้ไขค่าในเซลล์** (ดูเพิ่มเติมเรื่อง Triggers)

### 6.3.ก.) แก้ไขที่เซลล์ไหนก็ได้ มีผลต่อเซลล์ B2

เมื่อมีการแก้ไขค่าในเซลล์จะเกิด Event onEdit ซึ่งจะส่งผ่าน **วัตถุ Event** (Event Object) ไปเป็น Argument ให้กับฟังก์ชัน onEdit

เราสามารถดูได้ว่า ภายในวัตถุ Event มี **Keys**(หรือ **Properties**) อะไรบ้างได้ตามลิงค์ ยกตัวอย่างเช่น Key **range** เก็บเรนจ์ที่มีการเปลี่ยนแปลง Key **value** เก็บค่าในเซลล์ที่เปลี่ยนแปลง เป็นต้น



```

// ย้ายชีทที่จับออกมาไว้ข้างนอก เพื่อให้ทุกฟังก์ชันเรียกใช้ได้
var ws = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("master");

// ย้าย var ws ออกไปข้างนอก
function myFunction() {

    var ws = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("master");
    var list = [ "a" , "b" , "g" , "z" ] ;
    var cell = ws.getRange("C2");

    applyValidationToCell(list,cell)           // เรียกใช้ฟังก์ชัน applyValidationToCell
}

// ต้องใช้ชื่อฟังก์ชัน onEdit ชื่อนี้เท่านั้น – ลองเปลี่ยนชื่อแล้ว ไม่ทำงาน
// ชื่อ Argument เปลี่ยนจาก e เป็นอะไรก็ได้ เป็น x,y หรือ z ก็ได้
function onEdit(e){
    Logger.log(typeof e);           // ดู Log บรรทัด - [01]
    Logger.log(e);                  // ดู Log บรรทัด - [02]

    var activeCell = e.range ;
    Logger.log(activeCell);         // ดู Log บรรทัด - [03]

    var val = activeCell.getValue();
    Logger.log(val);                // ดู Log บรรทัด - [04]

    ws.getRange(2,2).setValue(val); // จัปไปที่เซลล์ B2 แล้วใส่เป็นค่าที่จับมาจาก e
}

// เหมือนเดิม
function applyValidationToCell(list,cell)() {

    var rule = SpreadsheetApp
        .newDataValidation()
        .requireValueInList(list)
        .setAllowInvalid(false)
        .build();

    cell.setDataValidation(rule);
}

```

เมื่อแก้ไขเซลล์ใดๆ จะเกิดค่าใน Log เปลี่ยนไปเรื่อยๆ

ผลที่ Logs มีดังต่อไปนี้

```

Logs
[01] object
[02] { authMode=LIMITED,           // วัตถุ Event
      range=Range ,              // คลาส Range ใช้ range.getValue() เข้าไปจับค่าต่อ
      source=Spreadsheet,
      oldValue=g ,
      user=wasankds@gmail.com ,
      value=z } // กรอกค่า z ลงในเซลล์
[03] Range
[03] z

```

เมื่อเขียนฟังก์ชัน onEdit แล้วให้บันทึกโปรเจ็ค โดยไม่ต้องรัน

จากนั้น เมื่อแก้ไขค่าในเซลล์ใดๆใน Sheet เซลล์ B2 จะจับค่ามาใส่ด้วย เช่น เมื่อเลือก Dropdown ในเซลล์ A2 เป็น aaa เซลล์ B2 จะจับค่าในเซลล์ A2 มาใส่ตาม

	A	B
1		
2	aaa ▼	aaa
3		

### 6.3.ข.) แก้ไขที่เซลล์ไหนก็ได้ มีผลต่อเซลล์ที่อยู่ติดกัน

แก้ฟังก์ชัน onEdit บรรทัด

```
ws.getRange(2,2).setValue(val);
```

เปลี่ยนเป็น

```
ws.getRange(activeCell.getRow(),2).setValue(val);
```

เพื่อให้ แก้ไขที่เซลล์ไหนก็ได้ มีผลต่อเซลล์ที่อยู่ติดกัน

```
var ws = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("master");  
function onEdit(e){  
  var activeCell = e.range ;  
  var val = activeCell.getValue();  
  
  //          แถวของเร้นจ์ที่จับมา,2  
  ws.getRange(activeCell.getRow(),2).setValue(val);  
}
```

ผล - บันทึกโปรเจ็ค แล้วทดสอบแก้ไขค่าในเซลล์ต่างๆ (ไม่ต้องรันฟังก์ชัน)

	A	B
1		
2	zzz ▼	zzz
3	zzz ▼	zzz
4	aaa ▼	aaa

### 6.3.ค.) แก้ไขที่เซลล์ในคอลัมน์ A เท่านั้น จึงจับค่าไปใช้

ขั้นตอนนี้ จะแก้ไขโค้ดต่อ โดยจุดประสงค์เพื่อจับค่าที่เปลี่ยนแปลงในคอลัมน์ A เท่านั้น ถ้าเป็นคอลัมน์อื่นๆเปลี่ยนแปลง ค่าที่จับได้จะไม่มีการนำไปใช้

```
var ws = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("master");

function onEdit(e){
  var activeCell = e.range ;
  var val = activeCell.getValue();
  var r = activeCell.getRow();
  var c = activeCell.getColumn();

  // จับค่าไปใช้ เมื่อเกิดการเปลี่ยนแปลงที่คอลัมน์ A ตั้งแต่แถวที่ 2 ขึ้นไป เท่านั้น
  // แถวที่ 1 เป็นป้ายชื่อ
  if( c === 1 && r > 1) {
    ws.getRange(r,c).setValue(val);
  }
}
```

### 6.3.ง.) มีผลเฉพาะชีทที่ระบุเท่านั้น

ขั้นตอนนี้ จะทำให้มีผลในชีทที่กำหนดเท่านั้น ก็คือ ชีทชื่อ master

```
var ws = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("master");

function onEdit(e){
  var activeCell = e.range ;
  var val = activeCell.getValue();
  var r = activeCell.getRow();
  var c = activeCell.getColumn();
  var wsName = activeCell.getSheet().getName(); // จับชื่อชีท ของเร้นที่ส่งมา

  // จับค่าไปใช้ เมื่อเกิดการเปลี่ยนแปลงที่
  // คอลัมน์ A + ตั้งแต่แถวที่ 2 ขึ้นไป + ที่ชีท master เท่านั้น
  if( wsName == "master" && c === 1 && r > 1) {
    ws.getRange(r,c).setValue(val);
  }
}
```

## 6.4. พัฒนาการที่ 3 : จับข้อมูลในอีกชีทมาทำเป็น List ให้กับ Dropdown

### 6.4.ก.) จับข้อมูลในอีกชีทมาทำเป็น List ให้กับ Dropdown

ขั้นตอนนี้จะเป็นการจับค่าในชีท options มาทำเป็น List ให้กับ Dropdown

```

var ws = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("master");
// จับชื่อ options
var wsOptions = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("options");
// จับ Data ในชื่อ options ไม่รวมหัวตาราง
var options = wsOptions.getRange(2, 1,wsOptions.getLastRow()-1,3).getValues();
// ไม่ได้ใช้ - เพราะเราเล่นที่ Event ไม่ได้รันฟังก์ชัน
function myFunction() {

    var ws = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("master");
    var list = [ "a" , "b" , "g" , "z" ] ;
    var cell = ws.getRange("C2");

    applyValidationToCell(list,cell)
}

function onEdit(e){
    var activeCell = e.range ;
    var val = activeCell.getValue();
    var r = activeCell.getRow();
    var c = activeCell.getColumn();

    var wsName = activeCell.getSheet().getName();

    if( wsName == "master" && c === 1 && r > 1){
        if(val === "") { // เช็คว่าค่าในคอลัมน์ A ว่าง - ให้ล้างคอลัมน์ B ด้วย
            ws.getRange(r,2).clearContent() ;
            ws.getRange(r,2).clearDataValidations() ;

        } else { // คอลัมน์ A ไม่ว่าง - กรองคอลัมน์ที่ค่าเป็น Val + เอาคอลัมน์ถัดไปมาทำ List
            ws.getRange(r,2).clearContent() ;
            var filterOptions = options.filter(function(o){ return o[0] === val }) ;
            var listToApply = filterOptions.map(function(o){ return o[1] }) ;
            var cell = ws.getRange(r,2) ;
            applyValidationToCell(listToApply,cell) ;
        } End - if else
    } End - if
} End - function

// เหมือนเดิม
function applyValidationToCell(list,cell)() {
    var rule = SpreadsheetApp
        .newDataValidation()
        .requireValueInList(list)
        .setAllowInvalid(false)
        .build();
    cell.setDataValidation(rule);
}

```

ผล – บันทึกโปรเจ็ค แล้วทดสอบแก้ไขค่าในเซลล์ต่างๆ (ไม่ต้องรันฟังก์ชัน)

	A	B	C
1	ภาค	จังหวัด	พนักงานขาย
2	ใต้		
3		ภูเก็ต	
4		นราธิวาส	
5		ยะลา	
6			

#### 6.4.ข.) เพิ่มค่าคงที่ firstLevelColumn และ secondLevelColumn

เพิ่มตัวแปรค่าคงที่ firstLevelColumn และ secondLevelColumn จากนั้นจับไปแทนที่ในฟังก์ชัน

```
var ws = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("master");
var wsOptions = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("options");
var options = wsOptions.getRange(2, 1,wsOptions.getLastRow()-1,3).getValues();

// เพิ่มค่าคงที่
var firstLevelColumn = 1;
var secondLevelColumn = 2;

// ไม่ได้ใช้ - เพราะเราเล่นที่ Event
function myFunction() {
    // เหมือนเดิม
}

function onEdit(e){
    var activeCell = e.range ;
    var val = activeCell.getValue();
    var r = activeCell.getRow();
    var c = activeCell.getColumn();

    var wsName = activeCell.getSheet().getName();

    if( wsName == "master" && c === firstLevelColumn && r > 1){
        if(val === "") { // เช็คว่าค่าในคอลัมน์ A ว่าง - ให้ล้างคอลัมน์ B ด้วย
            ws.getRange(r,secondLevelColumn).clearContent() ;
            ws.getRange(r,secondLevelColumn).clearDataValidations() ;

        } else { // คอลัมน์ A ไม่ว่าง - กรองคอลัมน์ที่ค่าเป็น Val + เอาคอลัมน์ถัดไปมาทำ List
            ws.getRange(r,secondLevelColumn).clearContent() ;
            var filterOptions = options.filter(function(o){ return o[0] === val }) ;
            var listToApply = filterOptions.map(function(o){ return o[1] }) ;
            var cell = ws.getRange(r,secondLevelColumn) ;
            applyValidationToCell(listToApply,cell) ;
        } // End - if else
    } // End - if
} // End - function

function applyValidationToCell(list,cell)() {
    // เหมือนเดิม
}
```

#### 6.4.ค.) แยกฟังก์ชัน

ขั้นตอนนี้ เราจะแยกฟังก์ชันที่ใช้ทำ Validation ออกมา ซึ่งทำได้ไม่ยาก แคโยกย้ายโค้ดนิดหน่อยๆ เท่านั้นเอง สุดท้ายแล้วจะได้ผลลัพธ์เหมือนกับข้อก่อนหน้า

```
var ws = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("master");
var wsOptions = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("options");
var options = wsOptions.getRange(2, 1,wsOptions.getLastRow()-1,3).getValues();
var firstLevelColumn = 1;
var secondLevelColumn = 2;

// ไม่ได้ใช้ - เพราะเราเล่นที่ Event
function myFunction() {
    // เหมือนเดิม
}

function onEdit(e){
    var activeCell = e.range ;
    var val = activeCell.getValue();
    var r = activeCell.getRow();
    var c = activeCell.getColumn();

    var wsName = activeCell.getSheet().getName();

    if( wsName == "master" && c === firstLevelColumn && r > 1){
        applyFirstLevelValidation(val,r);
    } // End - if
} // End - function

function applyFirstLevelValidation(val,r){

    if(val === "") {
        ws.getRange(r,secondLevelColumn).clearContent() ;
        ws.getRange(r,secondLevelColumn).clearDataValidations() ;

    } else {
        ws.getRange(r,secondLevelColumn).clearContent() ;
        var filterOptions = options.filter(function(o){ return o[0] === val }) ;
        var listToApply = filterOptions.map(function(o){ return o[1] }) ;
        var cell = ws.getRange(r,secondLevelColumn) ;
        applyValidationToCell(listToApply,cell) ;
    } // End - if else
} // End - function

function applyValidationToCell(list,cell)() {
    // เหมือนเดิม
}
```

## 6.5. พัฒนาการที่ 4 : สร้าง Validation ให้กับระดับที่ 2

สร้างฟังก์ชัน applySecondLevelValidation ทำงานคล้ายกับ applyFirstLevelValidation

```
var mainWsName = "master" ;
var optionsWsName = "options" ;
var ws = SpreadsheetApp.getActiveSpreadsheet().getSheetByName(mainWsName) ;
var wsOptions = SpreadsheetApp.getActiveSpreadsheet().getSheetByName(optionsWsName) ;
var options = wsOptions.getRange(2, 1,wsOptions.getLastRow()-1,3).getValues() ;
var firstLevelColumn = 1 ;
var secondLevelColumn = 2 ;
var thirdLevelColumn = 3 ;

function myFunction() {
    // เหมือนเดิม - ไม่ได้ใช้
}

function onEdit(e){
    var activeCell = e.range ;
    var val = activeCell.getValue();
    var r = activeCell.getRow();
    var c = activeCell.getColumn();

    var wsName = activeCell.getSheet().getName();

    if( wsName == "master" && c === firstLevelColumn && r > 1){
        applyFirstLevelValidation(val,r);
    } else if ( wsName === mainWsName && c === secondLevelColumn && r > 1) {
        applySecondLevelValidation(val,r);
    } // End - if
} // End - function

function applyFirstLevelValidation(val,r){

    if(val === "") {
        ws.getRange(r,secondLevelColumn).clearContent() ;
        ws.getRange(r,secondLevelColumn).clearDataValidations() ;
        ws.getRange(r,thirdLevelColumn).clearContent() ;
        ws.getRange(r,thirdLevelColumn).clearDataValidations() ;
    } else {
        ws.getRange(r,secondLevelColumn).clearContent() ;
        ws.getRange(r,secondLevelColumn).clearDataValidations() ;
        ws.getRange(r,thirdLevelColumn).clearContent() ;
        ws.getRange(r,thirdLevelColumn).clearDataValidations() ;
        var filterOptions = options.filter(function(o){ return o[0] === val }) ;
        var listToApply = filterOptions.map(function(o){ return o[1] }) ;
        var cell = ws.getRange(r,secondLevelColumn) ;
        applyValidationToCell(listToApply,cell) ;
    } // End - if else
} // End - function
```

```

function applySecondLevelValidation(val,r){
  if(val === ""){
    ws.getRange(r,thirdLevelColumn).clearContent();
    ws.getRange(r,thirdLevelColumn).clearDataValidations();
  } else {
    ws.getRange(r,thirdLevelColumn).clearContent();
    var firstLevelColValue = ws.getRange(r,firstLevelColumn).getValue();
    var filterOptions = options.filter(function(o){
      return o[0] === firstLevelColValue && o[1] === val
    });
    var listToApply = filterOptions.map(function(o){ return o[2] });
    var cell = ws.getRange(r,thirdLevelColumn);
    applyValidationToCell(listToApply,cell);
  } // End – if else
}

// เหมือนเดิม
function applyValidationToCell(list,cell)() {
  var rule = SpreadsheetApp
    .newDataValidation()
    .requireValueInList(list)
    .setAllowInvalid(false)
    .build();
  cell.setDataValidation(rule);
}

```

ผล

	A	B	C
1	ภาค	จังหวัด	พนักงานขาย
2	เหนือ	ลำปาง	สัมพันธ์
3	กลาง	กรุงเทพฯ	นภาพร
4			นภาพร
5			สมชาย
6			
7			