

Google Apps Script

เชอร์วิส Document

8 ธันวาคม 2019

เรียบเรียงโดย
วสันต์ คุณดิลกเศวต

Google Apps Script เชอร์วิส Document

เรียบเรียงโดย
วสันต์ คุณดิลกเศวต

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

สารบัญ

สารบัญ.....	4
คำนำ.....	7
บทที่ 1 เซอร์วิส Document.....	9
1.1. เซอร์วิส Document และ คลาส DocumentApp	10
1.2. โครงสร้างของเอกสาร Google Docs	11
บทที่ 2 คลาส Body.....	15
2.1. ส่วน Body ของเอกสาร	16
2.2. จับและล้าง Body	16
2.3. สร้างไฟล์ Google Docs และใส่เนื้อหาลงไป	17
2.4. Methods ตระกูล Append	18
2.4.ก.) appendParagraph() (18)	
2.4.ข.) appendPageBreak() (19)	
2.4.ค.) appendTable() (19)	
2.4.ง.) appendListItem(), getListId() , setListId() (20)	
2.4.จ.) GlyphType (22)	
2.4.ฉ.) appendImage()(23)	
2.4.ช.) appendHorizontalRule() (24)	
2.5. Methods ตระกูล Get และ set	24
2.5.ก.) getParagraphs(), getListItems(), getTables() (24)	
2.5.ข.) getText() (25)	
2.5.ค.) getMargin 4 ตัว และ getPage 2 ตัว (25)	
2.5.ง.) setMargin 4 ตัว และ setPage 2 ตัว (26)	
2.5.จ.) getParent() (26)	
2.5.ฉ.) getAttributes() คลาส Body (27)	
2.5.ช.) getAttributes() คลาส Paragraph (27)	
2.6. Enum Attribute	28
2.7. setAttributes() คลาส Paragraph	30
บทที่ 3 Heading.....	33
3.1. setHeading()	34
3.2. Enum ParagraphHeading	35
3.3. setHeadingAttributes()	35

บทที่ 4 Elements.....	37
4.1. Elements	38
4.2. Interface Element	38
4.3. getNumChildren()	39
4.4. จับและดูชนิดของ Elements (คลาส Body)	40
4.5. Enum ElementType	41
4.6. ระบุชนิด Element ด้วย Method ตระกูล As	42
4.7. findElement()	43
บทที่ 5 เคอร์เซอร์.....	45
5.1. getCursor()	46
5.2. setCursor(), newPosition()	47
บทที่ 6 Selection.....	49
6.1. getSelection(), getRangeElements(), getElement()	50
6.2. Methods ตระกูล Insert – ย้ายไป Element	54
บทที่ 7 Text.....	55
7.1. คลาส Text	56
7.2. editAtext(), insertText()	56
7.3. deleteText()	57
7.4. replaceText()	58
บทที่ 8 ภาพ.....	61
8.1. ใส่ภาพลงในตำแหน่ง Cursor	62

คำนำ

หนังสือเล่มนี้ เป็นหนึ่งในชุด **การเขียนโปรแกรม Google Apps Script** โดยในเล่มนี้อธิบายการใช้งาน **เซอร์วิส Document** ก็คือ เซอร์วิสสำหรับการใช้งาน Google Docs นั่นเอง

ผู้เขียน เขียนหนังสือเล่มนี้ จุดประสงค์ดั้งเดิม ก็คือ **เก็บไว้อ่านเอง**

เมื่อผู้เขียนศึกษาเรื่องอะไร ก็จะไปเรียนรู้จากสื่อออนไลน์ในอินเทอร์เน็ต ทั้งคอร์สออนไลน์ วิดีโอ หรือ เอกสาร ทั้งในแบบฟรีและเสียเงิน

ในยุคปัจจุบันเราต้องเรียนรู้อะไรให้เร็ว โดยเฉพาะเรื่องของ IT ผู้เขียนจึงตั้งใจจะดูวิดีโอเพียงรอบเดียว จึงไปพิมพ์สรุปไปด้วย เวลาจำอะไรไม่ได้ มาดูจากที่พิมพ์สรุปไว้ง่ายกว่าการไปย้อนดูจากวิดีโอ นอกจากนี้ก็นำมาทบทวนได้ง่าย ในอนาคตสามารถเพิ่มเติมเสริมแต่งเนื้อหาได้เรื่อยๆด้วย

หนังสือเล่มนี้ ความตั้งใจดั้งเดิมของผู้เขียน ก็คือ **ตั้งใจเก็บไว้อ่านเองคนเดียว** เหตุเพราะเนื้อหาเอามาจากของ **คนอื่นเป็นส่วนใหญ่** ผู้เขียนเขียนเพิ่มไปไม่มาก

อย่างไรก็ดี อดสำหรับพิมพ์ไว้เป็นหนังสือแล้ว จะเก็บไว้อ่านคนเดียวก็รู้สึกเสียดาย ผู้เขียนจึงนำมาแบ่งปัน

เนื่องด้วย ผู้เขียนให้ความสำคัญกับประเด็นด้านลิขสิทธิ์มาก ฉะนั้นจึงขอแจ้งไว้ ณ ที่นี้ ตั้งแต่ต้น **ก็คือ**

1. เนื้อหาในหนังสือ ผู้เขียนรวบรวมมาจากแหล่งต่างๆในอินเทอร์เน็ต ซึ่งจะพยายามให้มากที่สุด ที่จะบอกลิงค์หรือแหล่งที่มาในแต่ละหัวข้อ เพราะหนังสือเล่มนี้ถูกเขียนไว้นานแล้ว บางเรื่องลืมก็อปปีลิงค์มาแปะไว้
2. ผู้เขียนเป็นเพียงผู้รวบรวมเนื้อหา เขียนเพิ่มเติมเพียงเล็กน้อย
3. หนังสือเล่มนี้แจกฟรี ผู้เขียนไม่มีรายได้จากหนังสือเล่มนี้

หนังสือเล่มนี้ยังไม่จบเสียทีเดียว หากผู้เขียนว่าง จะมาเขียนเพิ่มเติมเรื่อยๆ ให้ดูเวอร์ชันตามวันที่ที่ปล่อย **หนังสือ**

วสันต์ คุณดิลกเศวต

wasankds@gmail.com

081-459-8343

Line ID : wasankds

บทที่ 1
เชอร์วิส
Document



1.1. เซอร์วิส Document และ คลาส DocumentApp

เซอร์วิส Document

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

เซอร์วิส Document ใช้สร้าง เข้าถึงไฟล์ และแก้ไขข้อมูลต่างๆในไฟล์ Google Docs เซอร์วิส

Document ประกอบไปด้วยหลากหลายคลาส ที่มีความสัมพันธ์กันอย่างเป็นโครงสร้าง

คลาส DocumentApp

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

การใช้งานเซอร์วิส Document เริ่มต้นจาก คลาส DocumentApp

คลาส DocumentApp ประกอบไปด้วย Properties(Enum) และ Methods ดังต่อไปนี้

Properties หรือ Enum ต่างๆในคลาส DocumentApp

Property	Type	Description
Attribute	Attribute	The Attribute enumeration.
ElementType	ElementType	The ElementType enumeration.
FontFamily	FontFamily	The FontFamily enumeration.
GlyphType	GlyphType	The GlyphType enumeration.
HorizontalAlignment	HorizontalAlignment	The HorizontalAlignment enumeration.
ParagraphHeading	ParagraphHeading	The ParagraphHeading enumeration.
PositionedLayout	PositionedLayout	The PositionedLayout enumeration.
TextAlignment	TextAlignment	The TextAlignment enumeration.
VerticalAlignment	VerticalAlignment	The VerticalAlignment enumeration.

Methods

Method	Return type	Brief description
create(name)	Document	Creates and returns a new document.
getActiveDocument()	Document	Returns the document to which the script is container-bound .
getUi()	Ui	Returns an instance of the document's user-interface environment that allows the script to add features like menus, dialogs, and sidebars.
openById(id)	Document	Returns the document with the specified ID.
openByUrl(url)	Document	Opens and returns the document with the specified URL.

หมายเหตุ : หนังสือเล่มนี้จะขอข้ามเรื่อง Methods ของคลาส DocumentApp เพราะทุกเซอร์วิส หลักๆ เช่น Spreadsheet, FormApp ใช้งานเหมือนกัน และได้อธิบายไปแล้วในหนังสือเล่มอื่น

ตัวอย่าง โค้ดต่อไปนี้ เรียกใช้เซอร์วิส Document โดยเริ่มต้นจากคลาส DocumentApp

```
// เปิดไฟล์โดยใช้ ID
var doc = DocumentApp.openById('DOCUMENT_ID_GOES_HERE');

// สร้างและเปิดเอกสารใหม่ - คืนค่ากลับมาเป็น Document
doc = DocumentApp.create('Document Name');
```

1.2. โครงสร้างของเอกสาร Google Docs

Structure of a document

https://developers.google.com/apps-script/guides/docs#structure_of_a_document

เอกสาร Google Docs ประกอบไปด้วยวัตถุหรือคลาสจำนวนหนึ่ง(เรียกรวมๆว่า Elements) มีความสัมพันธ์กันอย่างเป็นโครงสร้าง ซึ่งสามารถเขียนเป็นผัง Structure of a document ได้ในหน้าถัดไป ผังนี้ เป็นหัวใจสำคัญในการเขียนสคริปต์

ตัวที่ขีดเส้นใต้ หมายถึง แทรกลงในเอกสารได้

แต่ที่ไม่มีขีดเส้นใต้ หมายถึง สามารถจัดการในลักษณะอยู่กับที่ได้

Document

| 1.Body // Body ประกอบไปด้วย 4 ตัว ก็คือ ListItem, Paragraph, Table, TableOf Contents

| | ListItem (1)

| | | Equation

| | | | EquationFunction

| | | | EquationFunction...

| | | | EquationFunctionArgumentSeparator

| | | | EquationSymbol

| | | | Text

| | | | EquationSymbol

| | | | Text

| | | Footnote

| | | HorizontalRule

| | | InlineDrawing

| | | InlineImage

| | | PageBreak

| | | Text

| | Paragraph (2)

| | | Equation

| | | | EquationFunction

| | | | EquationFunction...

| | | | EquationFunctionArgumentSeparator

| | | | EquationSymbol

| | | | Text

| | | | EquationSymbol

| | | | Text

| | | Footnote

| | | HorizontalRule

| | | InlineDrawing

| | | InlineImage

| | | PageBreak

| | | Text

| | Table (3)

| | | TableRow

| | | | TableCell

| | | | Paragraph...

| | | | ListItem...

| | | | Table...

| | TableOfContents (4)

| | | Paragraph...

| | | ListItem...

| | | Table...

(มีต่อ)

(ต่อ)

- |
- |
- | - 2.HeaderSection
 - | | - [ListItem](#)
 - | | | - [HorizontalRule](#)
 - | | | - [InlineDrawing](#)
 - | | | - [InlineImage](#)
 - | | | - [Text](#)
 - | | | - [UnsupportedElement](#) (page number, etc.)
 - | | - [Paragraph](#)
 - | | | - [HorizontalRule](#)
 - | | | - [InlineDrawing](#)
 - | | | - [InlineImage](#)
 - | | | - [Text](#)
 - | | | - [UnsupportedElement](#) (page number, etc.)
 - | | - [Table](#)
 - | | | - [TableRow](#)
 - | | | | - [TableCell](#)
 - | | | | | - [Paragraph...](#)
 - | | | | | - [ListItem...](#)
 - | | | | | - [Table...](#)
- | - 3.FooterSection
 - | | - [ListItem](#)
 - | | | - [HorizontalRule](#)
 - | | | - [InlineDrawing](#)
 - | | | - [InlineImage](#)
 - | | | - [Text](#)
 - | | | - [UnsupportedElement](#) (page number, etc.)
 - | | - [Paragraph](#)
 - | | | - [HorizontalRule](#)
 - | | | - [InlineDrawing](#)
 - | | | - [InlineImage](#)
 - | | | - [Text](#)
 - | | | - [UnsupportedElement](#) (page number, etc.)
 - | | - [Table](#)
 - | | | - [TableRow](#)
 - | | | | - [TableCell](#)
 - | | | | | - [Paragraph...](#)
 - | | | | | - [ListItem...](#)
 - | | | | | - [Table...](#)
- | - 4.FootnoteSection
 - | | - [ListItem](#)
 - | | | - [HorizontalRule](#)
 - | | | - [Text](#)
 - | | - [Paragraph](#)
 - | | | - [HorizontalRule](#)
 - | | | - [Text](#)

บทที่ 2

คลาส Body



2.1. ส่วน Body ของเอกสาร

คลาส Body

<https://developers.google.com/apps-script/reference/document/body.html>

Body ของเอกสาร ก็คือ พื้นที่หลักที่เราใส่เนื้อหาลงในเอกสาร Google Docs

Body ประกอบไปด้วย Elements หลักๆ 4 ตัวก็คือ **ListItem**(หัวข้ออัตโนมัติ เช่น Bullets หรือ Numbering), **Paragraph**(ย่อหน้า), **Table**(ตาราง) และ **TableOfContents**(สารบัญ) **ยกเว้น** **HeaderSection**(หัวกระดาษ), **FooterSection**(ท้ายกระดาษ) และ **FootnoteSection**(เชิงอรรถ)

2.2. จับและล้าง Body

getBody() - Method ในคลาส Document

<https://developers.google.com/apps-script/reference/document/document.html#getbody>

ใช้จับส่วน Body ของเอกสาร โดยจะคืนค่ากลับมาเป็นคลาส Body

คลาส Body

<https://developers.google.com/apps-script/reference/document/body.html>

ตัวอย่าง

โค้ดต่อไปนี้จะพิมพ์ย่อหน้าที่มีข้อความ **A paragraph** ต่อท้าย

```
// จับไปที่ไฟล์ Google Docs ที่โปรเจ็ค Google Apps Script ผังอยู่
var doc = DocumentApp.getActiveDocument();

// จับไปที่ Body ของเอกสาร
var body = doc.getBody();

// แแนบท้ายด้วยย่อหน้า
body.appendParagraph("A paragraph.");

// ขึ้นหน้าใหม่
body.appendPageBreak();
```

หากต้องการล้างข้อความใน Body ใช้ Method **clear()**

clear() - Method ในคลาส Body

<https://developers.google.com/apps-script/reference/document/body.html#clear>

ตัวอย่าง

```
body.clear() // ล้างบอดี้
```


2.3. สร้างไฟล์ Google Docs และใส่เนื้อหาลงไป

Extending Google Docs – The basics

https://developers.google.com/apps-script/guides/docs#the_basics

สร้างโปรเจกต์ Google Apps Script แบบ Stand alone จากนั้นพิมพ์โค้ดต่อไปนี้ลงไป

```
function createDoc() {  
  
  var doc = DocumentApp.create('Sample Document') ;  
  var body = doc.getBody() ;  
  
  var rowsData = [ [ 'Plants' , 'Animals' ] ,  
                    [ 'Ficus' , 'Goat' ] ,  
                    [ 'Basil' , 'Cat' ] ,  
                    [ 'Moss' , 'Frog' ] ] ;  
  
  // แทรกย่อหน้าที่ตำแหน่ง Element ที่ 0 (ย่อหน้าแรก) ข้อความในย่อหน้าคือชื่อไฟล์  
  // และ กำหนดสไตล์ย่อหน้าเป็น Heading1  
  body.insertParagraph(0, doc.getName())  
    .setHeading(DocumentApp.ParagraphHeading.HEADING1) ;  
  
  // แบนทำยาดยตาราง ใส่เนื้อหาตามค่าในตัวแปรอาร์เรย์ rowsData  
  table = body.appendTable(rowsData) ;  
  // จับแถบแรกของตาราง + แก้ไขแบบข้อความ + กำหนดเป็นตัวหนา  
  table.getRow(0).editAsText().setBold(true) ;  
}
```

ผล

Sample Document

Plants	Animals
Ficus	Goat
Basil	Cat
Moss	Frog

2.4. Methods ตระกูล Append

Methods ตระกูล Append ใช้แนบท้ายบางอย่างลงในเอกสาร

2.4.ก.) appendParagraph()

appendParagraph([text](#)) – Method ในคลาส Body

<https://developers.google.com/apps-script/reference/document/body.html#appendparagraphtext>

appendParagraph([paragraph](#)) – Method ในคลาส Body

<https://developers.google.com/apps-script/reference/document/body.html#appendparagraphparagraph>

คลาส Paragraph

<https://developers.google.com/apps-script/reference/document/paragraph.html>

appendParagraph() ใช้แนบท้ายด้วยย่อหน้า โดยจะคืนค่ากลับมาเป็นคลาส Paragraph พารามิเตอร์ของ appendParagraph() เป็น [text\(ข้อความ\)](#) หรือ [คลาส Paragraph](#) ก็ได้ การระบุเป็น [คลาส Paragraph](#) ใช้ในกรณีก๊อปปีย่อหน้า

ตัวอย่างการใช้งาน

```
function appendParas(){  
    var body = DocumentApp.getActiveDocument().getBody() ;  
  
    // สร้างย่อหน้าแล้วจับใส่ตัวแปร  
    var header = body.appendParagraph("A Document") ;  
    // กำหนดสไตล์ย่อหน้า HEADING1 กับย่อหน้า  
    header.setHeading(DocumentApp.ParagraphHeading.HEADING1) ;  
  
    // แนบท้ายเอกสารด้วยย่อหน้าที่มีข้อความ Section 1  
    var section = body.appendParagraph("Section 1") ;  
    // กำหนดสไตล์ย่อหน้า HEADING2 กับย่อหน้า  
    section.setHeading(DocumentApp.ParagraphHeading.HEADING2) ;  
  
    // แนบท้ายเอกสารด้วยย่อหน้าที่มีข้อความ ... - เป็นย่อหน้าปกติ (ใช้สไตล์ Normal)  
    var para = body.appendParagraph("This is a typical paragraph.") ;  
  
    // ก๊อปปีย่อหน้า - พารามิเตอร์เป็นวัตถุย่อหน้า  
    body.appendParagraph(para.copy()) ;  
}
```

ผล

A Document

Section 1

This is a typical paragraph.

This is a typical paragraph.

2.4.ข.) appendPageBreak()

appendPageBreak() – Method ในคลาส Body

<https://developers.google.com/apps-script/reference/document/body.html#appendpagebreak>

appendPageBreak(**pageBreak**) – Method ในคลาส Body

<https://developers.google.com/apps-script/reference/document/body.html#appendpagebreakpagebreak>

ตามเอกสารของ Google อธิบายว่า ใช้สร้างและแนบคลาส PageBreak ซึ่งอธิบายง่าย ๆ ได้อีกอย่างว่า ใช้ขึ้นใหม่นั้นเอง โดยจะคืนค่ากลับมาเป็นคลาส PageBreak

ตัวอย่าง

```
body.appendPageBreak() ; // ขึ้นหน้าใหม่
```

2.4.ค.) appendTable()

appendTable() – Method ในคลาส Body

<https://developers.google.com/apps-script/reference/document/body.html#appendtable>

appendTable(**cells**) – Method ในคลาส Body

<https://developers.google.com/apps-script/reference/document/body.html#appendtablecells>

appendTable(**table**) – Method ในคลาส Body

<https://developers.google.com/apps-script/reference/document/body.html#appendtabletable>

คลาส Table

<https://developers.google.com/apps-script/reference/document/table.html>

ใช้แนบท้ายด้วยตาราง โดยจะคืนค่ากลับมาเป็นคลาส Table พารามิเตอร์สามารถเป็น **cells(String[][])** หรือ **ข้อความในอาเรย์ 2 มิติ** , เป็น **คลาส Table** หรือ **ไม่ระบุพารามิเตอร์** ก็ได้

แบบ **ไม่ระบุพารามิเตอร์** จะแนบย่อหน้าว่างๆหลังตาราง เพราะเอกสาร Google Docs จบด้วยตารางไม่ได้

แบบ **คลาส Table** ใช้ในกรณีก๊อปปี้ตาราง การใช้งานเช่น **body.appendTable(table.copy()) ;**

ตัวอย่าง

```
var doc = DocumentApp.getActiveDocument() ;
var body = doc.getBody() ;

// อาเรย์ 2 มิติที่จะใช้เป็นข้อมูลให้กับตาราง
var data = [ [ 'Plants' , 'Animals' ] ,
              [ 'Ficus' , 'Goat' ] ,
              [ 'Basil' , 'Cat' ] ,
              [ 'Moss' , 'Frog' ] ] ;

var table = body.appendTable(data) ; // ตารางที่ 1
body.appendTable(table.copy()) ; // ตารางที่ 2 ก๊อปปี้ตารางที่ 1 มา
```

ผล - ได้ 2 ตาราง ตารางแรกสร้างจากอาร์เรย์ข้อความ อีกตารางก๊อปไปมาจากตารางแรก

ย่อหน้าที่มีอยู่ก่อนแล้ว

Plants	Animals
Ficus	Goat
Basil	Cat
Moss	Frog

Plants	Animals
Ficus	Goat
Basil	Cat
Moss	Frog

2.4.ง.) `appendListItem()`, `getListId()` , `setListId()`

`appendListItem(text)` – Method ในคลาส Body

<https://developers.google.com/apps-script/reference/document/body.html#appendlistitemtext>

`appendListItem(listItem)` – Method ในคลาส Body

<https://developers.google.com/apps-script/reference/document/body.html#appendlistitemlistitem>

คลาส ListItem

<https://developers.google.com/apps-script/reference/document/list-item.html>

ใช้สร้างและแนบ ListItem ตัวใหม่ โดยจะคืนค่ากลับมาเป็นคลาส ListItem พารามิเตอร์สามารถเป็น `text` หรือเป็น `คลาส ListItem`(ใช้ในกรณีก๊อปไป ListItem)

1 ListItem มี 1 ย่อหน้า และมี List ID ด้วย โดย List ID ใช้เป็นตัวจัดกลุ่มว่าเป็น ListItem กลุ่มเดียวกันหรือไม่

ListItem ไม่สามารถมี `\n` (new-line characters) โดยจะถูกแปลงเป็น `\r` (line-break characters) คล้ายกด `<Shift><Enter>`

```
function appendListItem(){

    var body = DocumentApp.getActiveDocument().getBody() ;
    body.clear() ;

    // แบนท้ายด้วย ListItem ลำดับที่ 1
    var item1 = body.appendListItem('Item 1') ;

    // จับ List ID ของ item1
    Logger.log(item1.getListId()) ;           // เซนพิมพ์ : kix.z8suvlutfedt

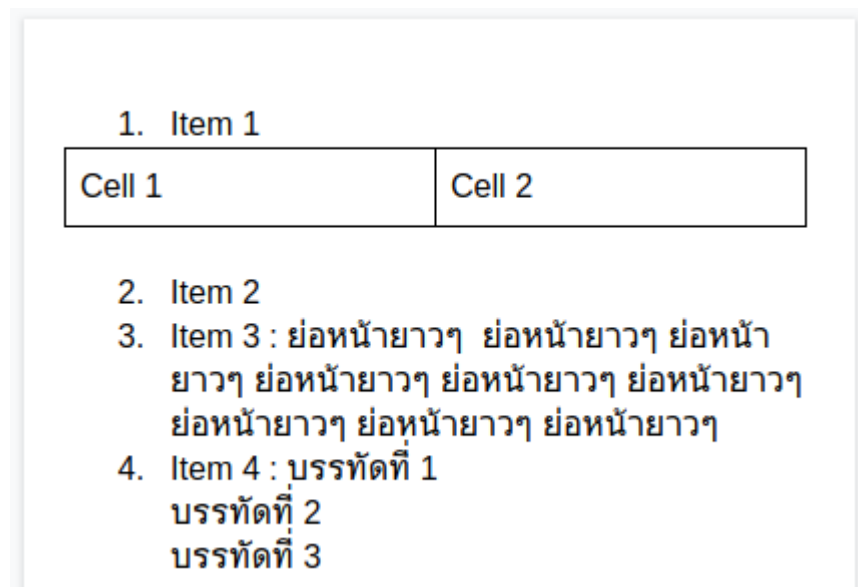
    // แบนท้ายด้วยตาราง จุดประสงค์เพื่อขึ้นกลางระหว่าง ListItem
    body.appendTable([[ 'Cell 1', 'Cell 2' ]]) ;

    // แบนท้ายด้วย ListItem ลำดับที่ 2
    // และเซต List ID ตัวเดียวกับ Item1 เพื่อให้เป็นกลุ่มเดียวกัน ตัวเลขจึงนับต่อกัน
    var item2 = body.appendListItem('Item 2') ;
    item2.setListId(item1) ;

    // แบนท้ายด้วย ListItem ลำดับที่ 3 - ข้อความยาวๆ
    var item3 = body.appendListItem('Item 3 : ย่อหน้ายาวๆ ย่อหน้ายาวๆ ย่อหน้ายาวๆ ย่อหน้ายาวๆ
                                     ย่อหน้ายาวๆ ย่อหน้ายาวๆ ย่อหน้ายาวๆ ย่อหน้ายาวๆ ย่อหน้ายาวๆ') ;

    // แบนท้ายด้วย ListItem ลำดับที่ 4 - มีการขึ้นบรรทัดใหม่ (คล้ายการกด <Shift><Enter>)
    var item4 = body.appendListItem('Item 4 : บรรทัดที่ 1\n บรรทัดที่ 2\n บรรทัดที่ 3\n') ;
}
```

ผล



`getListId()` - Method ในคลาส ListItem

<https://developers.google.com/apps-script/reference/slides/list?hl=vi#getlistid>

ใช้จับ List ID

`setListId(listItem)` - Method ในคลาส ListItem

<https://developers.google.com/apps-script/reference/document/list-item#setlistidlistitem>

ใช้กำหนด List ID ให้กับ ListItem

2.4.จ.) GlyphType

Enum GlyphType

<https://developers.google.com/apps-script/reference/document/glyph-type>

ใช้กำหนดรูปแบบของ Bullet ให้กับ ListItem การใช้งานเช่น

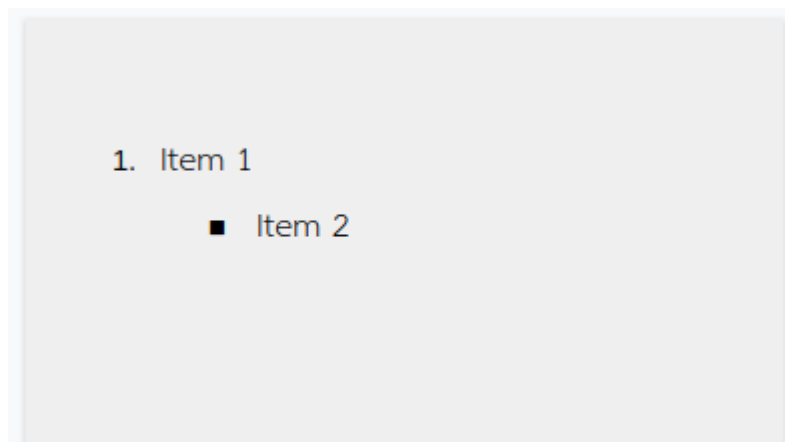
DocumentApp.GlyphType.[Property](#)

Property	Type	Description
BULLET	Enum	The default bullet, circular and filled.
HOLLOW_BULLET	Enum	A hollow bullet.
SQUARE_BULLET	Enum	A square bullet.
NUMBER	Enum	A number based bullet.
LATIN_UPPER	Enum	A latin, uppercase bullet.
LATIN_LOWER	Enum	A latin, lowercase bullet.
ROMAN_UPPER	Enum	A roman numeral, uppercase bullet.
ROMAN_LOWER	Enum	A roman numeral, lowercase bullet.

ตัวอย่าง

```
function bulletType() {  
  
    var docID = "1aqAv5bzhIvxWPheMpaQJu2vPTWZQQKd3qIsyGPlq-qM" ;  
    var docFiles = DocumentApp.openById(docID) ;  
    var docBody = DocumentApp.openById(docID).getBody() ;  
  
    // Insert at list item, with the default nesting level of zero.  
    docBody.appendListItem("Item 1") ;  
  
    // Append a second list item, with a nesting level of one, indented one inch.  
    // The two items will have different bullet glyphs.  
  
    docBody.appendListItem("Item 2")  
        .setNestingLevel(1)  
        .setIndentStart(72)  
        .setGlyphType(DocumentApp.GlyphType.SQUARE_BULLET) ;  
}
```

ผล



2.4.จ.) appendImage()

appendImage(image) – Method ในคลาส Body

<https://developers.google.com/apps-script/reference/document/body.html#appendimageimage>

appendImage(image) – Method ในคลาส Body (ใช้กรณีก๊อปปี้อภาพ)

https://developers.google.com/apps-script/reference/document/body.html#appendimageimage_1

คลาส InlineImage

<https://developers.google.com/apps-script/reference/document/inline-image.html>

ใช้สร้างและแนบท้ายด้วยภาพ InLineImage โดยจะคืนค่ากลับมาเป็นคลาส InLineImage

พารามิเตอร์ image เป็น BlobSource ที่มีข้อมูลเป็นภาพ

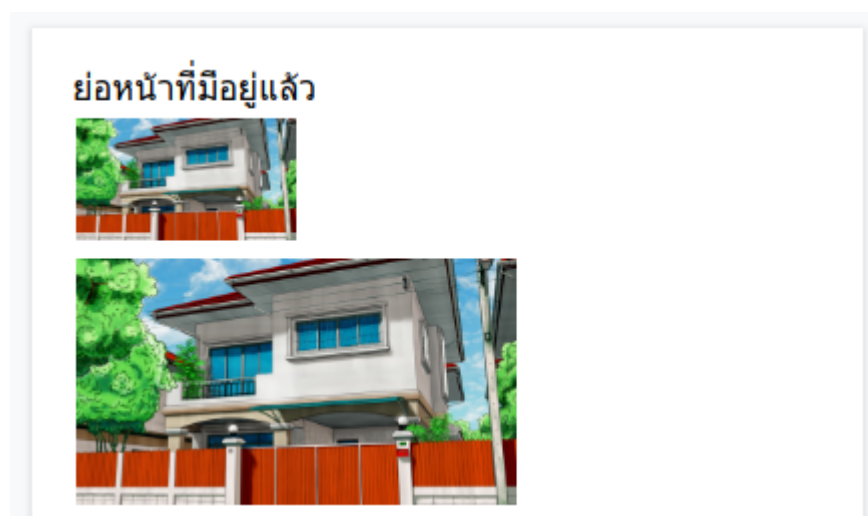
ตัวอย่าง

ดัดแปลงมาจาก - Insert Inline Image in a Google Document

<https://ctrlq.org/code/20078-insert-image-in-google-document>

```
function insertImage() {  
  // ภาพจากเว็บ  
  // var imageLink = "https://img.labnol.org/logo.png" ;  
  // var blob = UrlFetchApp.fetch(imageLink).getBlob() ;           // จับ Blob ใส่ตัวแปร  
  
  // ภาพจาก Google Drive – จับใส่ BLOB ก่อน  
  var imageLink = "https://drive.google.com/open?id=1inBoKWzWZr_ip2OvBAv3BWY7TbW9JDt" ;  
  var imageID = imageLink.match(/[a-zA-Z0-9]{25,}/).toString() ; // จับมาเฉพาะ ID  
  var blob = DriveApp.getFileById(imageID).getBlob() ;           // จับ Blob ของภาพใส่ตัวแปร  
  
  var body = DocumentApp.getActiveDocument().getBody() ;  
  
  // แนบท้ายด้วยภาพลงใน Body เอกสาร  
  var image = body.appendImage(blob).setWidth(100).setHeight(100/1.7777) ;  
  // ก๊อปปี้อภาพ  
  body.appendImage(image.copy()).setWidth(200).setHeight(200/1.7777) ;  
}
```

ผล



2.4.ช.) appendHorizontalRule()

appendHorizontalRule()

<https://developers.google.com/apps-script/reference/document/body.html#appendhorizontalrule>

ใช้แนบท้ายด้วยย่อหน้าที่มีคลาส `HorizontalRule` โดยจะคืนค่ากลับมาเป็นคลาส `HorizontalRule`

`HorizontalRule` ก็คือ เส้นที่ขีดด้วยแท็ก `<hr>` ในโค้ด HTML

2.5. Methods ตระกูล Get และ set

2.5.ก.) getParagraphs(), getListItems(), getTables()

getParagraphs() - Method ในคลาส Body

<https://developers.google.com/apps-script/reference/document/body#getparagraphs>

ใช้จับย่อหน้าที่อยู่ใน Body รวมถึงย่อหน้าที่อยู่ใน ListItem และ Table ด้วย (ถ้าอยู่ในวัตถุบรรทัดตัวอื่น ก็จับย่อหน้าในบรรทัดตัวนั้น) โดยจะคืนค่ากลับมาเป็น `Paragraph[]` (คลาส `Paragraph` ในอาร์เย)

getListItems() - Method ในคลาส Body

<https://developers.google.com/apps-script/reference/document/body#getlistitems>

ใช้จับ ListItem ที่อยู่ใน Body โดย 1 หัว(1 ย่อหน้า) จะจับมาเป็น 1 คลาส ListItem (ถ้าอยู่ในวัตถุบรรทัดตัวอื่น ก็จับ ListItem ในบรรทัดตัวนั้น) โดยจะคืนค่ากลับมาเป็น `ListItem[]` (คลาส `ListItem` ในอาร์เย)

getTables() - Method ในคลาส Body

<https://developers.google.com/apps-script/reference/document/body#gettables>

ใช้จับตารางที่อยู่ใน Body (ถ้าอยู่ในวัตถุบรรทัดตัวอื่น ก็จับตารางในบรรทัดตัวนั้น) โดยจะคืนค่ากลับมาเป็น `Table[]` (คลาส `Table` ในอาร์เย)

ตัวอย่าง

```
function Tables() {
  var body = DocumentApp.getActiveDocument().getBody() ;
  var paras = body.getParagraphs() ;
  var lists = body.getListItems() ;
  var tables = body.getTables() ;

  Logger.log(body) ;    // พิมพ์ : DocumentBodySection
  Logger.log(paras) ;   // พิมพ์ : [Paragraph, Paragraph, Paragraph, ... ]
  Logger.log(lists) ;   // พิมพ์ : [ListItem, ListItem]
  Logger.log(tables) ;  // พิมพ์ : DocumentBodySection

  Logger.log(body.getType()) ;    // พิมพ์ : BODY_SECTION      ()
  Logger.log(paras[0].getType()); // พิมพ์ : PARAGRAPH
  Logger.log(lists[0].getType()); // พิมพ์ : LIST_ITEM
  Logger.log(tables[0].getType()); // พิมพ์ : TABLE

  tables.forEach(function(table){
    body.appendTable(table.copy()) ;    // แนบท้ายด้วยตารางที่ก๊อปปี้ไว้ ลงไป
  }) ;
}
```


หมายเหตุ

getType() เป็น Method ในหลายคลาส ดู Enum ElementType

<https://developers.google.com/apps-script/reference/document/element-type>

2.5.ข.) getText()

getText() - Method ในคลาส Body

<https://developers.google.com/apps-script/reference/document/body#getText>

ใช้จับข้อความทั้งหมดที่อยู่ใน Body รวมถึงข้อความที่อยู่ใน ListItem, Paragraph และ Table (ถ้าอยู่ในวัตถุบรรทัดอื่น ก็จับข้อความในบรรทัดนั้น) โดยจะคืนค่ากลับมาเป็นข้อความ(String)

ตัวอย่าง

```
var body = DocumentApp.getActiveDocument().getBody() ;  
var text = body.getText() ;
```

2.5.ค.) getMargin 4 ตัว และ getPage 2 ตัว

getMarginTop(), getMarginBottom(), getMarginLeft(), getMarginRight() - Method ในคลาส Body

[https://developers.google.com/apps-script/reference/document/body#getMarginTop\(\)](https://developers.google.com/apps-script/reference/document/body#getMarginTop())

[https://developers.google.com/apps-script/reference/document/body#getMarginBottom\(\)](https://developers.google.com/apps-script/reference/document/body#getMarginBottom())

[https://developers.google.com/apps-script/reference/document/body#getMarginLeft\(\)](https://developers.google.com/apps-script/reference/document/body#getMarginLeft())

[https://developers.google.com/apps-script/reference/document/body#getMarginRight\(\)](https://developers.google.com/apps-script/reference/document/body#getMarginRight())

ใช้จับระยะ Margin(ระยะเว้นขอบกระดาษ) ทั้ง 4 ด้าน โดยจะคืนค่ากลับมาเป็นจำนวนเต็ม ในหน่วย points

getPageHeight(), getPageWidth() - Method ในคลาส Body

<https://developers.google.com/apps-script/reference/document/body#getpageheight>

<https://developers.google.com/apps-script/reference/document/body#getpagewidth>

ใช้จับความสูงและความกว้างของหน้ากระดาษ โดยจะคืนค่ากลับมาเป็นจำนวนเต็ม ในหน่วย points

ตัวอย่าง

```
function margin_page() {  
  
    var body = DocumentApp.getActiveDocument().getBody() ;  
  
    Logger.log(body.getMarginTop()) ;           // เช่นพิมพ์ : 14.1732  
    Logger.log(body.getMarginBottom()) ;        // เช่นพิมพ์ : 14.1732  
    Logger.log(body.getMarginLeft()) ;          // เช่นพิมพ์ : 14.1732  
    Logger.log(body.getMarginRight()) ;         // เช่นพิมพ์ : 14.1732  
  
    Logger.log(body.getPageHeight()) ;          // เช่นพิมพ์ : 400.899887517858  
    Logger.log(body.getPageWidth()) ;           // เช่นพิมพ์ : 283.465  
} // Close - function
```

หมายเหตุ : 1 ซม. = 28.3465 พ้อยต์
1 นิ้ว = 72 พ้อยต์
1 พ้อยต์ = 0.0352778 ซม.
1 พ้อยต์ = 0.0138889 นิ้ว

2.5.ง.) setMargin 4 ตัว และ setPage 2 ตัว

การตั้งค่าหน้ากระดาษ Google Docs ที่เมนู File → Page Setup ไม่มีให้เลือกแบบ Custom Size หรือ แบบที่เรากำหนดขนาดเองได้ มีแต่แบบสำเร็จรูปให้เลือก เช่น A4, A5 เป็นต้น

หากต้องการตั้งค่าหน้ากระดาษแบบ Custom Size เราต้องเขียน Script โดยใช้ Method ต่างๆ เช่น setMarginLeft หรือ setPageHeight ในการกำหนดขนาดและระยะต่างๆของหน้ากระดาษ

ตัวอย่าง

```
function CustomPageSize(){
  var body = DocumentApp.getActiveDocument().getBody() ;

  body.setMarginLeft(14.1732)           // ตั้งระยะเว้นขอบด้านซ้าย 0.5 ซม
  .setMarginRight(14.1732)             // ตั้งระยะเว้นขอบด้านขวา 0.5 ซม
  .setMarginTop(14.1732)               // ตั้งระยะเว้นขอบด้านบน 0.5 ซม
  .setMarginBottom(14.1732)           // ตั้งระยะเว้นขอบด้านล่าง 0.5 ซม
  .setPageWidth(283.465)               // ตั้งความกว้างของหน้ากระดาษ 10 ซม.
  .setPageHeight(400.899887517858) ;   // ตั้งความกว้างของหน้ากระดาษ 14.143 ซม.
}
```

2.5.จ.) getParent()

getParent() - Method ในคลาส Body

[https://developers.google.com/apps-script/reference/document/body#getParent\(\)](https://developers.google.com/apps-script/reference/document/body#getParent())

ใช้จับ Element ที่เป็นวัตถุบรรจุมัณฑ เช่น Body เป็นวัตถุบรรจุมัณฑของ Paragraph, ListItem และ Table เป็นต้น โดยจะคืนค่ากลับมาเป็นคลาส ContainerElement (วัตถุบรรจุมัณฑ)

คลาส ContainerElement

<https://developers.google.com/apps-script/reference/document/container-element.html>

ตัวอย่าง

```
function parent() {
  var body = DocumentApp.getActiveDocument().getBody() ;
  var para = body.getParagraphs()[0] ;
  var table = body.getTables()[0] ;
  var cell = table.getCell(0, 0) ;

  Logger.log(body.getParent()) ;           // พิมพ์ : DocumentElement
  Logger.log(para.getParent()) ;           // พิมพ์ : DocumentBodySection
  Logger.log(table.getParent()) ;          // พิมพ์ : DocumentBodySection
  Logger.log(cell.getParent()) ;           // พิมพ์ : TableRow
  Logger.log(cell.getParent().getParent()) ; // พิมพ์ : Table
} // Close – function
```

2.5.ฉ.) getAttributes() คลาส Body

getAttributes() - Method ในคลาส Body และหลายคลาส

[https://developers.google.com/apps-script/reference/document/body#getAttributes\(\)](https://developers.google.com/apps-script/reference/document/body#getAttributes())

ใช้จับคุณลักษณะ(Attributes) ของ Element (ในที่นี้ก็คือ Body) โดยจะคืนค่ากลับมาเป็น **Object[]**

หรือ วัตถุที่เก็บ Enum Attribute ได้

ดูเพิ่มเติมข้อ 2.6 Enum Attribute หน้า 28

ตัวอย่าง

```
function getAttsBody() {  
  
    var body = DocumentApp.getActiveDocument().getBody();  
    var attsBody = body.getAttributes();  
  
    for(var att in attsBody){  
        Logger.log(att + " : " + attsBody[att]) ;  
    }  
  
} // Close – function
```

ผล

Logs

```
[ 01 ] FONT_SIZE : null  
[ 02 ] ITALIC : null  
[ 03 ] PAGE_WIDTH : 283.465  
[ 04 ] LINK_URL : null  
[ 05 ] UNDERLINE : null  
[ 06 ] BACKGROUND_COLOR : null  
[ 07 ] MARGIN_BOTTOM : 14.1732  
[ 08 ] PAGE_HEIGHT : 400.899887517858  
[ 09 ] MARGIN_RIGHT : 14.1732  
[ 10 ] STRIKETHROUGH : null  
[ 11 ] MARGIN_LEFT : 14.1732  
[ 12 ] FOREGROUND_COLOR : null  
[ 13 ] BOLD : null  
[ 14 ] FONT_FAMILY : Sarabun  
[ 15 ] MARGIN_TOP : 14.1732
```

2.5.ช.) getAttributes() คลาส Paragraph

getAttributes ใช้บ่อยและเป็น Method ในหลายคลาส ฉะนั้นจึงขออธิบาย getAttributes ในคลาส Paragraph เพิ่มเติม

getAttributes() - Method ในคลาส Paragraph และหลายคลาส

[https://developers.google.com/apps-script/reference/document/paragraph#getAttributes\(\)](https://developers.google.com/apps-script/reference/document/paragraph#getAttributes())

ใช้จับคุณลักษณะ(Attributes) ของ Element (ในที่นี้ก็คือ Paragraph) โดยจะคืนค่ากลับมาเป็น Object[] หรือ วัตถุที่เก็บ Enum Attribute ไว้ (ดูเพิ่มเติมข้อ 2.6 Enum Attribute หน้า 28)

ตัวอย่าง

```
function getAttsPara() {  
    var body = DocumentApp.getActiveDocument().getBody() ;  
    // แบนทำยด้วยย่อหน้าต่อไปนี้  
    var par = body.appendParagraph('A bold, italicized paragraph.') ;  
    par.setBold(true) ;  
    par.setItalic(true) ;  
  
    // จับ Attributes ของย่อหน้า  
    var atts = par.getAttributes() ;  
  
    for (var att in atts) {  
        Logger.log(att + " : " + atts[att]) ;  
    }  
} // Close – function
```

ผล

Logs

```
[ 01 ] FONT_SIZE : 8  
[ 02 ] ITALIC : true  
[ 03 ] HORIZONTAL_ALIGNMENT : Left  
[ 04 ] INDENT_END : 0  
[ 05 ] INDENT_START : 0  
[ 06 ] LINE_SPACING : 1.15  
[ 07 ] LINK_URL : null  
[ 08 ] UNDERLINE : null  
[ 09 ] BACKGROUND_COLOR : null  
[ 10 ] INDENT_FIRST_LINE : 0  
[ 11 ] LEFT_TO_RIGHT : true  
[ 12 ] SPACING_BEFORE : 0  
[ 13 ] HEADING : Normal  
[ 14 ] SPACING_AFTER : 0  
[ 15 ] STRIKETHROUGH : null  
[ 16 ] FOREGROUND_COLOR : null  
[ 17 ] BOLD : true  
[ 18 ] FONT_FAMILY : Sarabun
```

2.6. Enum Attribute

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

Enum Attribute เป็นการแจกแจง(Enumeration) คุณลักษณะต่างๆขององค์ประกอบหรือวัตถุ หรืออธิบายอีกอย่างว่า เป็นคุณสมบัติของวัตถุที่ถูกกำหนดไว้แล้ว ว่าเราสามารถเข้าไปตั้งค่าอะไรได้บ้าง เช่น คุณสมบัติ BOLD ก็คือ คุณสมบัติตัวหนาของอักษร เป็นต้น

เราสามารถเรียกใช้ Enum Attribute ได้จาก DocumentApp.Attribute.[\[Property\]](#) โดย Properties มีดังต่อไปนี้

Property	Type	Description
BACKGROUND_COLOR	Enum	The background color of an element (Paragraph, Table, etc) or document.
BOLD	Enum	The font weight setting, for rich text.
BORDER_COLOR	Enum	The border color, for table elements.
BORDER_WIDTH	Enum	The border width in points, for table elements.
CODE	Enum	The code contents, for equation elements.
FONT_FAMILY	Enum	The font family setting, for rich text.
FONT_SIZE	Enum	The font size setting in points, for rich text.
FOREGROUND_COLOR	Enum	The foreground color setting, for rich text.
HEADING	Enum	The heading type, for paragraph elements (for example, DocumentApp.).
HEIGHT	Enum	The height setting, for image elements.
HORIZONTAL_ALIGNMENT	Enum	The horizontal alignment, for paragraph elements (for example, DocumentApp.).
INDENT_END	Enum	The end indentation setting in points, for paragraph elements.
INDENT_FIRST_LINE	Enum	The first line indentation setting in points, for paragraph elements.
INDENT_START	Enum	The start indentation setting in points, for paragraph elements.
ITALIC	Enum	The font style setting, for rich text.
GLYPH_TYPE	Enum	The glyph type, for list item elements.
LEFT_TO_RIGHT	Enum	The text direction setting, for rich text.
LINE_SPACING	Enum	The line spacing setting as a multiplier, for paragraph elements.
LINK_URL	Enum	The link URL, for rich text. The default link style (foreground color, underline) is automatically applied.
LIST_ID	Enum	The ID of the encompassing list, for list item elements.
MARGIN_BOTTOM	Enum	The bottom margin setting in points, for paragraph elements.
MARGIN_LEFT	Enum	The left margin setting in points, for paragraph elements.
MARGIN_RIGHT	Enum	The right margin setting in points, for paragraph elements.
MARGIN_TOP	Enum	The top margin setting in points, for paragraph elements.
NESTING_LEVEL	Enum	The item nesting level, for list item elements.
MINIMUM_HEIGHT	Enum	The minimum height setting in points, for table row elements.
PADDING_BOTTOM	Enum	The bottom padding setting in points, for table cell elements.
PADDING_LEFT	Enum	The left padding setting in points, for table cell elements.
PADDING_RIGHT	Enum	The right padding setting in points, for table cell elements.
PADDING_TOP	Enum	The top padding setting in points, for table cell elements.
PAGE_HEIGHT	Enum	The page height setting in points, for documents.
PAGE_WIDTH	Enum	The page width setting in points, for documents.
SPACING_AFTER	Enum	The bottom spacing setting in points, for paragraph elements.
SPACING_BEFORE	Enum	The top spacing setting in points, for paragraph elements.
STRIKETHROUGH	Enum	The strike-through setting, for rich text.
UNDERLINE	Enum	The underline setting, for rich text.
VERTICAL_ALIGNMENT	Enum	The vertical alignment setting, for table cell elements.
WIDTH	Enum	The width setting, for table cell and image elements.

ตัวอย่าง

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

โค้ดต่อไปนี้ เมื่อรันจะพิมพ์ข้อความ Hello ไว้ที่จุดแรกของเอกสาร Google Docs

```
// Define a style with yellow background.

var highlightStyle = {} ;
highlightStyle[DocumentApp.Attribute.BACKGROUND_COLOR] = '#FFFF00'
highlightStyle[DocumentApp.Attribute.BOLD] = true ;

// Insert "Hello", highlighted.
DocumentApp.getActiveDocument().editAsText()
    .insertText(0, 'Hello\n')
    .setAttributes(0, 4, highlightStyle);
```

ผล



2.7. setAttributes() คลาส Paragraph

setAttributes(attributes) – Method ในคลาส Paragraph และหลายคลาส

<https://developers.google.com/apps-script/reference/document/paragraph#setattributesattributes>

ใช้กำหนด Attributes ให้กับ Element ในที่นี้ก็คือ Paragraph โดยจะคืนค่ากลับมาเป็นย่อหน้าตัวที่จับมา เพียงแต่ Attributes เปลี่ยนไป พารามิเตอร์ attributes เป็น Object[] หรือ เป็นตัวแปรวัตถุที่ Properties ต่างๆตาม Enum Attribute

ตัวอย่าง

```
function setAttsPara(){

    var body = DocumentApp.getActiveDocument().getBody();

    // เตรียมตัวแปร Object วางๆรอไว้
    var style = {} ;

    // บ้อน Item ที่เป็น Property : Value ให้กับตัวแปร Object ที่เตรียมไว้
    style[DocumentApp.Attribute.HORIZONTAL_ALIGNMENT] =
        DocumentApp.HorizontalAlignment.RIGHT ;

    style[DocumentApp.Attribute.FONT_FAMILY] = 'Sarabun' ;
    style[DocumentApp.Attribute.FONT_SIZE] = 12 ;
    style[DocumentApp.Attribute.BOLD] = false ;
    style[DocumentApp.Attribute.FOREGROUND_COLOR] = '#0000FF' ;

    // แบนท้ายด้วยย่อหน้า para ลงใน Body ของเอกสาร
    var par = body.appendParagraph('A paragraph with custom style.');
```

// กำหนด Attributes ให้กับย่อหน้า para

```
body.setAttributes(style) ;
}
```

A paragraph with custom style.

บทที่ 3

Heading



3.1. setHeading()

คลาส Paragraph

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

setHeading(**heading**) – Method ในคลาส Paragraph

<https://developers.google.com/apps-script/reference/document/paragraph#setheadingheading>

กำหนดสไตล์ย่อหน้ามาตรฐานให้กับย่อหน้า โดยจะคืนค่ากลับมาเป็นย่อหน้าปัจจุบัน พารามิเตอร์

heading ดูได้จาก Enum ParagraphHeading (ดูเพิ่มเติมข้อ 3.2 Enum ParagraphHeading หน้า 35)

getHeading() – Method ในคลาส Paragraph

<https://developers.google.com/apps-script/reference/document/paragraph#getheading>

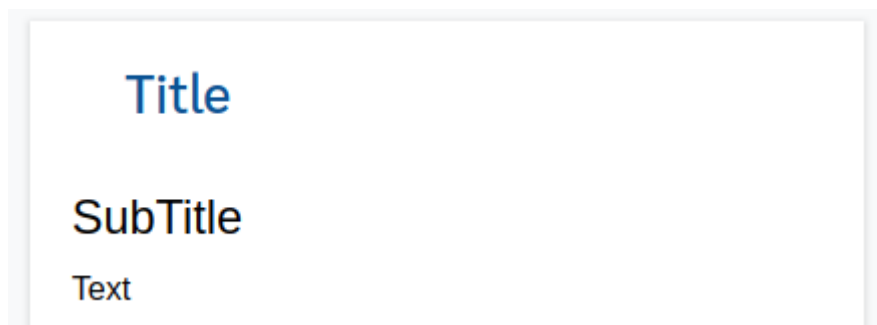
ใช้จับสไตล์ย่อหน้ามาตรฐานของย่อหน้า โดยคืนค่ากลับมาเป็น Property ใน Enum

ParagraphHeading (ดูเพิ่มเติมข้อ 3.2 Enum ParagraphHeading หน้า 35)

ตัวอย่าง

```
function example() {  
  var body = DocumentApp.getActiveDocument().getBody() ;  
  body.clear() ;  
  
  // แบนท้ายด้วยย่อหน้า + กำหนดสไตล์ย่อหน้าเป็น Heading 1  
  var par1 = body.appendParagraph("Title") ;  
  par1.setHeading(DocumentApp.ParagraphHeading.HEADING1) ;  
  
  // แบนท้ายด้วยย่อหน้า + กำหนดสไตล์ย่อหน้าเป็น Heading 2  
  var par2 = body.appendParagraph("SubTitle") ;  
  par2.setHeading(DocumentApp.ParagraphHeading.HEADING2) ;  
  
  // แบนท้ายด้วยย่อหน้า + กำหนดสไตล์ย่อหน้าเป็น Normal text  
  var par3 = body.appendParagraph("Text") ;  
  par3.setHeading(DocumentApp.ParagraphHeading.NORMAL) ;  
}
```

ผลลัพธ์รันโค้ด



3.2. Enum ParagraphHeading

Enum ParagraphHeading

<https://developers.google.com/apps-script/reference/document/paragraph-heading>

แจกแจงคุณสมบัติสไตล์ย่อหน้ามาตรฐาน

การเรียกใช้งาน เช่น `DocumentApp.ParagraphHeading.Property`

Property	Type	Description
<code>NORMAL</code>	Enum	The heading option for normal text.
<code>HEADING1</code>	Enum	The highest heading option.
<code>HEADING2</code>	Enum	The second heading option.
<code>HEADING3</code>	Enum	The third heading option
<code>HEADING4</code>	Enum	The fourth heading option.
<code>HEADING5</code>	Enum	The fifth heading option.
<code>HEADING6</code>	Enum	The lowest heading option.
<code>TITLE</code>	Enum	The title heading option.
<code>SUBTITLE</code>	Enum	The subtitle heading option.

3.3. setHeadingAttributes()

`setHeadingAttributes(paragraphHeading, attributes)` – Method ในคลาส Body

[https://developers.google.com/apps-script/reference/document/body#setHeadingAttributes\(ParagraphHeading,Object\)](https://developers.google.com/apps-script/reference/document/body#setHeadingAttributes(ParagraphHeading,Object))

ใช้กำหนด Attributes ให้กับสไตล์ย่อหน้ามาตรฐาน โดยจะคืนค่ากลับมาเป็นคลาส Body ตัวเดิม

พารามิเตอร์ – Method ในคลาส Body

Name	Type	Description
<code>paragraphHeading</code>	ParagraphHeading	the heading whose attributes will be set ดูข้อ 3.2 Enum ParagraphHeading หน้า 35
<code>attributes</code>	Object	a map of attributes and the values to set them to ดูข้อ 2.6 Enum Attribute หน้า 28

`getHeadingAttributes(paragraphHeading)`

<https://developers.google.com/apps-script/reference/document/body#getheadingattributesparagraphheading>

ใช้จับ Attributes ของสไตล์ย่อหน้ามาตรฐาน โดยคืนค่ากลับมาเป็น Object ที่มีสมาชิกเป็น Enum

Attributes และ ค่าต่างๆ พารามิเตอร์ `paragraphHeading` ก็คือ Enum `paragraphHeading`

การใช้งานเช่น

`body.getHeadingAttributes(DocumentApp.ParagraphHeading.HEADING1)`

Logs

```
[ 01 ] {FONT_SIZE=24, ITALIC=false, HORIZONTAL_ALIGNMENT=Left,
INDENT_END=0.0, INDENT_START=0.0, LINE_SPACING=1.15, UNDERLINE=false,
BACKGROUND_COLOR=#ffff00, INDENT_FIRST_LINE=0.0, SPACING_BEFORE=20.0,
SPACING_AFTER=6.0, STRIKETHROUGH=false, FOREGROUND_COLOR=#00ff00,
BOLD=false, FONT_FAMILY=Sarabun, VERTICAL_ALIGNMENT=Center}
```

ตัวอย่าง

```
function HeadingAttributes() {

  var myHeading1 = {} ;
  myHeading1[DocumentApp.Attribute.FONT_SIZE] = 24 ;
  myHeading1[DocumentApp.Attribute.FONT_FAMILY] = "Sarabun" ;

  var myHeading2 = {} ;
  myHeading2[DocumentApp.Attribute.FONT_SIZE] = 16 ;
  myHeading2[DocumentApp.Attribute.FONT_FAMILY] = "Sarabun" ;
  myHeading2[DocumentApp.Attribute.FOREGROUND_COLOR] = "#555555" ;

  var myNormal = {} ;
  myNormal[DocumentApp.Attribute.FONT_SIZE] = 12 ;
  myNormal[DocumentApp.Attribute.FONT_FAMILY] = "Sarabun" ;
  myNormal[DocumentApp.Attribute.FOREGROUND_COLOR] = "#505050" ;
  myNormal[DocumentApp.Attribute.SPACING_BEFORE] = 5 ;
  myNormal[DocumentApp.Attribute.SPACING_AFTER] = 0 ;

  var body = DocumentApp.getActiveDocument().getBody() ;

  body.setHeadingAttributes(DocumentApp.ParagraphHeading.HEADING1, myHeading1)
    .setHeadingAttributes(DocumentApp.ParagraphHeading.HEADING2, myHeading2)
    .setHeadingAttributes(DocumentApp.ParagraphHeading.NORMAL, myNormal) ;
}
```

บทที่ 4

Elements



4.1. Elements

Elements คืออะไร ?

เอกสาร Google Docs ประกอบไปด้วย **Elements(วัตถุธาตุ)** ต่างๆมากมาย เช่น **ListItem**, **Paragraph** และ **Table** เป็นต้น ซึ่งทุกตัวก็คือ **Elements** ภายในเอกสาร

Elements ลูก หรือ Child elements

เมื่อเราเข้าไปที่ Element สักตัวหนึ่งในเอกสาร Element ตัวนั้นจะมี Elements ย่อยๆข้างในอีก เช่น เมื่อจับที่ **Body** ลูกๆของ **Body** ก็คือ **Paragraph**, **ListItem**, **Table**, **TableOfContents** เป็นต้น ซึ่งเราเรียกว่า Elements ย่อยๆข้างในนี้ว่า **Elements ลูก** หรือ **Child elements**

4.2. Interface Element

Interface Element – Interface ของเซอร์วิส Document

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

Elements ต่างๆมี Methods กลุ่มหนึ่งที่ซ้ำกัน ซึ่งเราสามารถจับมามีตรวมกัน (Inherit หรือสืบทอดสิ่งที่ซ้ำกันออกมา) กลายเป็น **Interface Element** หรือ เป็นกลุ่ม **Methods**

Methods ใน **Interface Element** ยกตัวอย่างเช่น **getAttributes()**, Method ตระกูล **as** เช่น **asParagraph()** เป็นต้น

หมายเหตุ 1 : มีสิ่งที่คล้ายกันก็คือ คลาส **Element**(ตามลิงค์ด้านล่าง) แต่เป็นวัตถุหรือคลาสในเซอร์วิส XML

เซอร์วิส XML

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

เป็นบริการที่ใช้ วิเคราะห์ สืบค้น และโปรแกรม ในการสร้างเอกสาร XML

คลาส **Element** (ในเซอร์วิส XML)

<https://developers.google.com/apps-script/reference/xml-service/element>

เป็น XML ของ Element node

หมายเหตุ 2 : ภาษา XML มีโครงสร้างเหมือนกับ HTML มีแท็กเปิดแท็กปิด แต่ XML เราสามารถสร้างแท็กและกำหนดโครงสร้างของข้อมูลได้เอง ต่างจาก HTML ที่แท็กถูกกำหนดตายตัว ตัวอย่าง

XML Tutorial

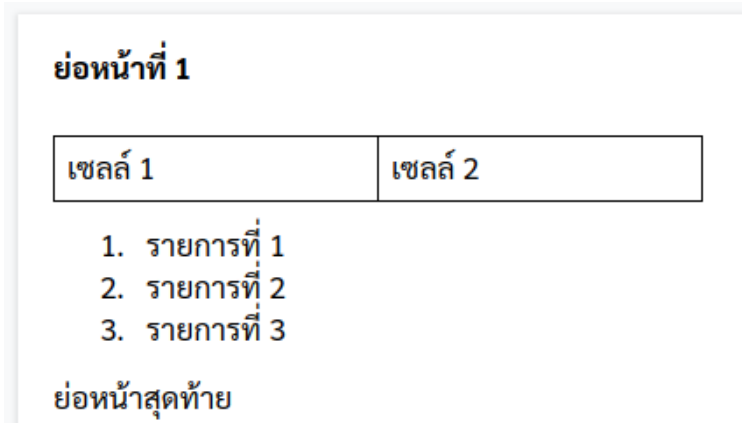
<https://www.w3schools.com/xml/>

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

แต่ละแท็ก ที่เราสร้างเรียกว่า โหนด(node) เช่น โหนด Note เป็นต้น

ยกตัวอย่าง การดูชนิดของ Elements ใน Body

องค์ประกอบในส่วน Body ในเอกสาร Google Docs มีตามภาพ



โค้ดต่อไปนี้ จับ Elements ใน Body ของเอกสาร Log ดูว่ามีชนิดเป็นอะไรบ้าง

```
function testElem() {  
  
    var body = DocumentApp.getActiveDocument().getBody();  
    var elemInBody = body.getNumChildren();  
  
    for(var x = 0 ; x < elemInBody ; x++){  
        Logger.log(body.getChild(x).getType()) ;    // วนหาลูป Log ชนิด Elements ที่จับมา  
    };  
}
```

ผล

```
[ 01 ] PARAGRAPH  
[ 02 ] PARAGRAPH    // ย่อหน้าว่าง  
[ 03 ] TABLE  
[ 04 ] LIST_ITEM  
[ 05 ] LIST_ITEM  
[ 06 ] LIST_ITEM  
[ 07 ] PARAGRAPH
```

4.3. getNumChildren()

getNumChildren() - Method ในคลาส ContainerElement

<https://developers.google.com/apps-script/reference/document/container-element#getnumchildren>

ใช้จับจำนวนของ Elements ลุก โดยจะคืนค่ากลับมาเป็นเลขจำนวนเต็ม

คลาส ContainerElement

<https://developers.google.com/apps-script/reference/document/container-element>

Element ที่เป็นตัวเก็บ Elements อื่นๆ (เรียกว่า Element บรรจุภัณฑ์ก็ได้)

ตัวอย่าง

ให้ใส่อะไรลงไปเอกสาร Google Docs บ้างจากนั้นจึงรันโค้ดต่อไปนี้

```
var body = DocumentApp.getActiveDocument().getBody();
var elemInBody = body.getNumChildren();
// ถ้า Log ดูจะได้ผลลัพธ์เช่น 7 (เป็นจำนวนของ Elements) เป็นต้น
```

4.4. จับและดูชนิดของ Elements (คลาส Body)

getChild() เป็น Method ในหลายคลาส หลักๆก็คือใช้จับ Child elements เช่น

getChild(name) – Method ในคลาส Element

<https://developers.google.com/apps-script/reference/xml-service/element#getchildname>

getChild(name, namespace) – Method ในคลาส Element

<https://developers.google.com/apps-script/reference/xml-service/element#getchildname,-namespace>

getChild(childIndex) – Method ในคลาส Paragraph

<https://developers.google.com/apps-script/reference/document/paragraph#getchildchildindex>

getChild(childIndex) – Method ในคลาส Body ***

<https://developers.google.com/apps-script/reference/document/body#getchildchildindex>

ใช้จับ Child elements ใน Body โดยคืนค่ากลับมาเป็น คลาส Element พารามิเตอร์ childIndex

เป็นเลขตรรกศาสตร์ (เริ่มจาก 0

getType() – Method ในคลาส Body

<https://developers.google.com/apps-script/reference/document/body#gettype>

ใช้จับชนิดของ Elements โดยจะคืนค่ากลับมาเป็น Enum ElementType

ตัวอย่าง

```
function getChildElems() {
    var body = DocumentApp.getActiveDocument().getBody();
    // จับไปที่ Child element ตัวแรกใน Body
    var firstChild = body.getChild(0);
    // ถ้า Element เป็นย่อหน้า(PARAGRAPH) ให้ใส่ข้อความ ... ลงไปแทน
    if (firstChild.getType() == DocumentApp.ElementType.PARAGRAPH) {
        firstChild.asParagraph().setText("This is the first paragraph.");
    }
}
```

ผล

ก่อนรัน

ย่อหน้าที่ 1	
เซลล์ 1	เซลล์ 2

หลังรัน

This is the first paragraph.	
เซลล์ 1	เซลล์ 2

4.5. Enum ElementType

Enum ElementType

<https://developers.google.com/apps-script/reference/document/element-type>

เป็นตัวแจกแจงชนิดของ Elements การใช้งานก็คือ `DocumentApp.ElementType.(Property)`

Property	Type	Description
<code>BODY_SECTION</code>	Enum	The type corresponding to the <code>Body</code> element.
<code>COMMENT_SECTION</code>	Enum	The type corresponding to the <code>CommentSection</code> element.
<code>DOCUMENT</code>	Enum	The type that corresponds to the root of the document.
<code>EQUATION</code>	Enum	The type corresponding to the <code>Equation</code> element.
<code>EQUATION_FUNCTION</code>	Enum	The type corresponding to the <code>EquationFunction</code> element.
<code>EQUATION_FUNCTION_ARGUMENT_SEPARATOR</code>	Enum	The type corresponding to the <code>EquationFunctionArgumentSeparator</code> element.
<code>EQUATION_SYMBOL</code>	Enum	The type corresponding to the <code>EquationSymbol</code> element.
<code>FOOTER_SECTION</code>	Enum	The type corresponding to the <code>FooterSection</code> element.
<code>FOOTNOTE</code>	Enum	The type corresponding to the <code>Footnote</code> element.
<code>FOOTNOTE_SECTION</code>	Enum	The type corresponding to the <code>FootnoteSection</code> element.
<code>HEADER_SECTION</code>	Enum	The type corresponding to the <code>HeaderSection</code> element.
<code>HORIZONTAL_RULE</code>	Enum	The type corresponding to the <code>HorizontalRule</code> element.
<code>INLINE_DRAWING</code>	Enum	The type corresponding to the <code>InlineDrawing</code> element.
<code>INLINE_IMAGE</code>	Enum	The type corresponding to the <code>InlineImage</code> element.
<code>LIST_ITEM</code>	Enum	The type corresponding to the <code>ListItem</code> element.
<code>PAGE_BREAK</code>	Enum	The type corresponding to the <code>PageBreak</code> element.
<code>PARAGRAPH</code>	Enum	The type corresponding to the <code>Paragraph</code> element.
<code>TABLE</code>	Enum	The type corresponding to the <code>Table</code> element.
<code>TABLE_CELL</code>	Enum	The type corresponding to the <code>TableCell</code> element.
<code>TABLE_OF_CONTENTS</code>	Enum	The type corresponding to the <code>TableOfContents</code> element.
<code>TABLE_ROW</code>	Enum	The type corresponding to the <code>TableRow</code> element.
<code>TEXT</code>	Enum	The type corresponding to the <code>Text</code> element.
<code>UNSUPPORTED</code>	Enum	The type corresponding to <code>UnsupportedElement</code> . Unsupported elements represent document portions that do not support scripting.

4.6. ระบุชนิด Element ด้วย Method ตระกูล As

เมื่อจับ Child elements แล้ว หากจะเข้าไปทำอะไรต่อ เราต้องระบุ Element ตัวนั้นตามชนิดด้วย Method ตระกูล as เช่น `asParagraph()`, `asTable()` หรือ `asListItem()` เป็นต้น จากนั้นเราจะสามารถใช้ Method ที่ตรงกับชนิดของ Element เพื่อทำอะไรต่อก็ว่าไป

`asParagraph()` - Method ใน Interface Element

<https://developers.google.com/apps-script/reference/document/element#asparagraph>

คืนค่า Element ที่จับมาได้เป็น Paragraph

เช่น

```
function getChildElems() {  
  
    var body = DocumentApp.getActiveDocument().getBody() ;  
    var childsCount = body.getNumChildren() ;  
  
    for(var x = 0 ; x < childsCount ; x++) {           // วนลูปจับทีละ Element ไปใส่ if/else  
        var child = body.getChild(x).copy() ;  
  
        // ถ้าเป็นย่อหน้า + ย่อหน้าไม่ว่าง  
        if ( child.getType() == DocumentApp.ElementType.PARAGRAPH &&  
            child.asParagraph().getText() != "" ) {  
            var para = child.asParagraph() ;           // ระบุ Element เป็น Paragraph  
            body.appendParagraph(para).appendText('ตัวก๊อปปี') ;  
  
            // ถ้าเป็นตาราง  
        } else if ( child.getType() == DocumentApp.ElementType.TABLE ) {  
            var table = child.asTable() ;               // ระบุ Element เป็น Table  
            var rowAdd = table.getRow(0).copy() ;       // ก๊อปปีแถวแรก  
            body.appendTable(table).appendTableRow(rowAdd) ; // แแนบแถวแรกที่ก๊อปปี  
        } // Close - else if  
    } // Close - for  
} // Close - function
```

ผล

ก่อน

ย่อหน้าดั้งเดิม

เซลล์ 1	เซลล์ 2
เซลล์ 3	เซลล์ 4

1. รายการที่ 1
2. รายการที่ 2

หลัง

ย่อหน้าดั้งเดิม

เซลล์ 1	เซลล์ 2
เซลล์ 3	เซลล์ 4

1. รายการที่ 1
2. รายการที่ 2

ย่อหน้าดั้งเดิม(ตัวก๊อปปี)

เซลล์ 1	เซลล์ 2
เซลล์ 3	เซลล์ 4
เซลล์ 1	เซลล์ 2

Logs

```
[ 01 ] PARAGRAPH
[ 02 ] TABLE
[ 03 ] LIST_ITEM
[ 04 ] LIST_ITEM
[ 05 ] LIST_ITEM
[ 06 ] PARAGRAPH
```

หมายเหตุ

ดูรายละเอียดเพิ่มเติมได้จากลิงค์ต่อไปนี้

getChild – Method ในคลาส ContainerElement/Element

<https://developers.google.com/apps-script/reference/document/container-element#getchildchildindex>

<https://developers.google.com/apps-script/reference/xml-service/element#getchildname>

ใช้จับ Element ตามตรรกษาลำดับ เช่น **getChild(0)** เป็นต้น คืนค่ากลับมาเป็น Element

getNumChildren() – Method ในคลาส ContainerElement

<https://developers.google.com/apps-script/reference/document/container-element#getnumchildren>

ใช้จับจำนวน Elements คืนค่ากลับมาเป็น เลขจำนวนเต็ม

4.7. findElement()

findElement(elementType, from) – Method ในหลายคลาส Body และอีกหลายคลาส

<https://developers.google.com/apps-script/reference/document/list-item.html#findelementelementtype,-from>

ใช้ค้นหาเนื้อหาใน Element แบบเจาะจงชนิด โดยจะคืนค่ากลับมาเป็นคลาส RangeElement

พารามิเตอร์

Name	Type	Description
elementType	ElementType	the type of element to search for
from	RangeElement	the search result to search from

ตัวอย่าง

```
function example() {  
  
  var body = DocumentApp.getActiveDocument().getBody() ;  
  
  // สร้างตัวแปร  
  var searchType = DocumentApp.ElementType.PARAGRAPH ;  
  var searchHeading = DocumentApp.ParagraphHeading.HEADING1 ;  
  var searchResult = null ;  
  
  // Search until the paragraph is found.  
  while (searchResult = body.findElement(searchType, searchResult)) {  
    var par = searchResult.getElement().asParagraph() ;  
    if (par.getHeading() == searchHeading) {  
  
      // Found one, update and stop.  
      par.setText('This is the first header.');      return ;  
    }  
  }  
}
```

ผล - หลังรันโค้ดย่อหน้าแรกที่ใช้สไตล์ย่อหน้า Heading1 ข้อความจะถูกเปลี่ยนเป็น **This is the first header**

บทที่ 5

เคอร์เซอร์



5.1. `getCursor()`

`getCursor()` – Method ในคลาส Document

<https://developers.google.com/apps-script/reference/document/document#getcursor>

ใช้จับเคอร์เซอร์ในเอกสารที่กำลังทำงานด้วย โดยจับได้เฉพาะของยูสเซอร์ที่รันสคริปต์ และเป็นสคริปต์ฝังในไฟล์ (ถูกห่อหุ้ม หรือ **bound** ← คลิกเพื่อดูรายละเอียด) โดยจะคืนค่ากลับมาเป็นคลาส **Position** แต่ถ้าจับไม่ได้จะคืนค่ากลับมาเป็น **null**

คลาส Position

<https://developers.google.com/apps-script/reference/document/position.html>

ตำแหน่งอ้างอิงในเอกสาร Google Docs เป็นแบบอ้างอิงจาก Element

ตัวอย่าง : แทรกข้อความ ณ ตำแหน่งเคอร์เซอร์

โค้ดต่อไปนี้จะรันจากโปรเจ็ค Google Apps Script จะค้าง วิธีรันต้องสร้างเมนู แล้วรันจากเมนู ทั้งนี้เพื่อที่จะจับตำแหน่งของเคอร์เซอร์ได้ ถ้ารันจาก Google Apps Script เคอร์เซอร์ไม่แอ็คทีฟที่เอกสาร Google Docs

ก่อนรัน ให้คลิกแทรกในย่อหน้าหรือข้อความไว้ จากนั้นจึงรันสคริปต์จากเมนู แต่ถ้าคลิกเลือกไว้ที่ภาพ จะปรากฏหน้าต่างมาแจ้งว่า **Cannot insert text here** ก็คือ ไม่สามารถแทรกข้อความได้ (เพราะไม่มีเคอร์เซอร์)

```
function onOpen() {  
    DocumentApp.getUi().createMenu('My Menu')  
        .addItem('Insert someting at cursor' , 'insertAtCursor')  
        .addToUi() ;  
}  
  
function insertAtCursor(){  
    // จับตำแหน่งของเคอร์เซอร์ ได้กลับมาเป็นคลาส Position  
    var cursor = DocumentApp.getActiveDocument().getCursor() ;  
  
    // ถ้าจับเคอร์เซอร์ได้ - แทรกข้อความ ณ ตำแหน่งของเคอร์เซอร์  
    if (cursor) { // if/else #1  
        var element = cursor.insertText('๖_๖') ;  
        if (element) { // if/else #2  
            element.setBold(true); // ทำตัวหนา  
        } else {  
            DocumentApp.getUi().alert('Cannot insert text here.');        } // Close - if/else #2  
  
        // ถ้าจับเคอร์เซอร์ไม่ได้ - แสดงหน้าต่างแจ้ง Error  
    } else {  
        DocumentApp.getUi().alert('Cannot find a cursor.');    } // Close - if/else #1  
}
```

5.2. setCursor(), newPosition()

setCursor(position) – Method ในคลาส Document

<https://developers.google.com/apps-script/reference/document/document#setcursorposition>

ใช้กำหนดตำแหน่งของเคอร์เซอร์ ณ ตำแหน่งที่ระบุใน **position** (เป็น **New position**)

newPosition(element, offset) – Method ในคลาส Document

<https://developers.google.com/apps-script/reference/document/document#newpositionelement,-offset>

สร้างวัตถุ **Position ตัวใหม่** โดยอ้างอิงจากจาก Element ในเอกสาร

Name	Type	Description
element	Element	Element ที่จะใช้บรรจุวัตถุ Position ตัวใหม่ ซึ่งสามารถเป็นได้ทั้ง Text หรือ Paragraph
offset	Integer	สำหรับ Element ที่เป็น Text เป็นตัวเลขจำนวนตัวอักษรก่อนตำแหน่ง สำหรับ Elements อื่นๆ เป็นตัวเลขจำนวน Child elements ก่อนตำแหน่ง ภายในวัตถุบรรจุภัณฑ์ตัวเดียวกัน

ตัวอย่าง

```
function setCurAtNewPos() {  
  
    var doc = DocumentApp.getActiveDocument();  
    var paragraph = doc.getBody().appendParagraph('My new paragraph.');
```

// สร้างวัตถุ Position ณ ตำแหน่ง หลังตัวอักษรที่ 6 ของย่อหน้า

```
    var position = doc.newPosition(paragraph.getChild(0), 6);  
  
    // กำหนดตำแหน่งให้กับเคอร์เซอร์  
    doc.setCursor(position);  
}
```

ผล – มีย่อหน้าต่อท้ายเอกสาร โดยเคอร์เซอร์จะปรากฏอยู่หลังตัวอักษรที่ 6 (หลัง w)

ย่อหน้าที่มีอยู่ก่อนแล้ว
My new|paragraph.

บทที่ 6

Selection



6.1. `getSelection()`, `getRangeElements()`, `getElement()`

`getSelection()` - Method ในคลาส Document

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

จับสิ่งที่เลือกโดยยูสเซอร์ ในเอกสารที่แอ็คทีฟอยู่ คืนค่ากลับมาเป็นคลาส `Range` แต่ถ้าไม่เลือกอะไรไว้

จะคืนค่า `null`

คลาส `Range`

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

เรนจ์ของ Element ในเอกสาร โดยสิ่งที่ยูสเซอร์เลือกก็คือ `Range`

`getRangeElements()` - Method ในคลาส `Range`

<https://developers.google.com/apps-script/reference/document/range.html#getrangeelements>

ใช้จับ Elements ทั้งหมดใน `Range` (Element ที่ถูกเลือก) แม้จะเป็นบางส่วนของ Element ที่ถูก

เลือก ยกตัวอย่างการเลือกบางส่วนของย่อหน้า โดย Method นี้ จะคืนค่ากลับมาเป็น `RangeElement[]` หรือ

อาร์เรย์ที่บรรจุไปด้วย คลาส `RangeElement`

คลาส `RangeElement`

<https://developers.google.com/apps-script/reference/document/range-element.html>

ตัวห่อ Element ที่ถูกเลือก ตั้งแต่จุดเริ่มและจบครอบ Element เช่น ถ้าเลือก 1 ตัวอักษรในย่อหน้า

`RangeElement` ก็คือ ย่อหน้าทั้งก้อน

`getElement()` - Method ในคลาส `RangeElement`

<https://developers.google.com/apps-script/reference/document/range-element#getelement>

ใช้จับ Element ในวัตถุ `RangeElement`

`isPartial()` - Method ในคลาส `RangeElement`

<https://developers.google.com/apps-script/reference/document/range-element.html#ispartial>

ใช้เช็คค่า `RangeElement` ที่เลือก ถูกเลือกทั้งก้อน หรือบางส่วนของข้อความ(Text element) โดยจะ

คืนค่ากลับมาเป็น Boolean จุดเริ่มต้นและจุดจบของ `RangeElement` สามารถจับได้โดยใช้ Method

`getStartOffset()` และ `getEndOffsetInclusive()` ตามลำดับ

`getStartOffset()` - Method ในคลาส `RangeElement`

<https://developers.google.com/apps-script/reference/document/range-element.html#getstartoffset>

ตำแหน่งจุดเริ่มต้นของ `RangeElement` (Text element) อธิบายไว้ข้างบน Method นี้คืนค่ากลับมา

เป็นเลขจำนวนเต็มของตำแหน่ง และคืนค่ากลับมาเป็น -1 หากเป็นอย่างอื่น

`getEndOffsetInclusive()` - Method ในคลาส `RangeElement`

<https://developers.google.com/apps-script/reference/document/range-element.html#getendoffsetinclusive>

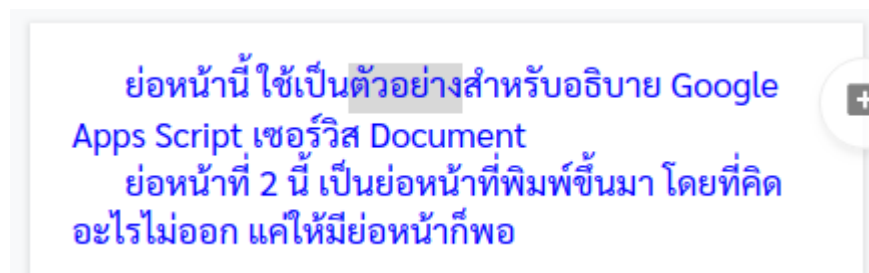
ตำแหน่งจุดจบของ `RangeElement` (Text element) อธิบายไว้ข้างบน Method นี้คืนค่ากลับมาเป็น

เลขจำนวนเต็มของตำแหน่ง และคืนค่ากลับมาเป็น -1 หากเป็นอย่างอื่น

ตัวอย่างที่ 1

```
function catchElem() {  
    // ก่อนรันโค้ดให้เลือก ไว้สักตัวอักษรหนึ่งในย่อหน้า เป็นอย่างน้อย หรือ เลือกตารางก็ได้  
    var selection = DocumentApp.getActiveDocument().getSelection();  
    // Logger.log(selection);           // พิมพ์ : Range  
  
    var rangeElems = selection.getRangeElements();  
    // Logger.log(rangeElems);           // พิมพ์ : [RangeElement, RangeElement]  
                                         // ( กรณีเลือกบางส่วนของ 2 ย่อหน้าไว้ )  
  
    var elemInRangeElems = rangeElems[0].getElement().getType();  
    // Logger.log(elemInRangeElems);     // พิมพ์ชนิดของ Element  
                                         // เช่น TEXT / TABLE_CELL  
  
    var text = selection.getRangeElements()[0].getElement().getText();  
    Logger.log(text);                   // พิมพ์ : ข้อความทั้งย่อหน้า ที่ถูกเลือกอยู่  
}
```

ก่อนรันโค้ด เลือกข้อความบางส่วนในย่อหน้าไว้



ผล - หลังรันโค้ด

```
Logs  
[ 01 ] Range  
[ 02 ] [RangeElement]  
[ 03 ] TEXT           // ถ้าเลือกตารางไว้ ขึ้นเป็น TABLE_CELL  
[ 04 ] ย่อหน้านี้ ใช้เป็นตัวอย่างสำหรับอธิบาย Google Apps Script เซอร์วิส Document
```

ตัวอย่างที่ 2

```
function Example2() {  
    var selection = DocumentApp.getActiveDocument().getSelection();  
    var Elements = selection.getRangeElements();  
    Logger.log(Elements);  
  
    Elements.forEach(function(element,i){  
        Logger.log(element+' '+i+' type : ' +element.getElement().getType());  
    });  
}
```

ผล – ก่อนรันโค้ดเลือกไว้หลายอย่าง ทั้งตารางและย่อหน้า

Logs

[01] [RangeElement, RangeElement, RangeElement, RangeElement, RangeElement, RangeElement, RangeElement, RangeElement, RangeElement]

[02] RangeElement 0 type : PARAGRAPH

[03] RangeElement 1 type : TABLE

[04] RangeElement 2 type : PARAGRAPH

[05] RangeElement 3 type : TABLE

[06] RangeElement 4 type : PARAGRAPH

[07] RangeElement 5 type : TABLE

[08] RangeElement 6 type : PARAGRAPH

[09] RangeElement 7 type : LIST_ITEM

[10] RangeElement 8 type : LIST_ITEM

getType() - Method ใน Interface Element

<https://developers.google.com/apps-script/reference/document/element#getType>

ใช้จับชนิดของ Element คืนค่ากลับมาเป็น Enum ElementType

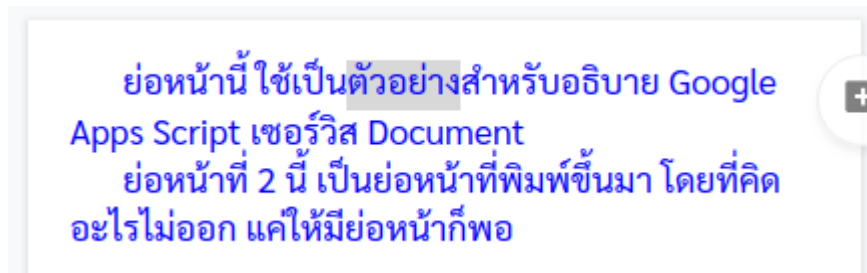
Enum ElementType – แจกแจงชนิดของ Elements

<https://developers.google.com/apps-script/reference/document/element-type.html>

ตัวอย่างที่ 3

```
function Example3() {  
    var selection = DocumentApp.getActiveDocument().getSelection() ;  
    // จับ Element ตัวแรกที่ถูกเลือก  
    var rangeElement = selection.getRangeElements()[0] ;  
    Logger.log('Element type: ' + rangeElement.getElement().getType());  
    // ถ้าถูกเลือกบางส่วน  
    if (rangeElement.isPartial()) {  
        Logger.log('The character range begins at ' + rangeElement.getStartOffset());  
        Logger.log('The character range ends at ' + rangeElement.getEndOffsetInclusive());  
    }  
    // ถ้าถูกเลือกทั้ง Element  
    } else {  
        Logger.log('The entire range element is included.');
```

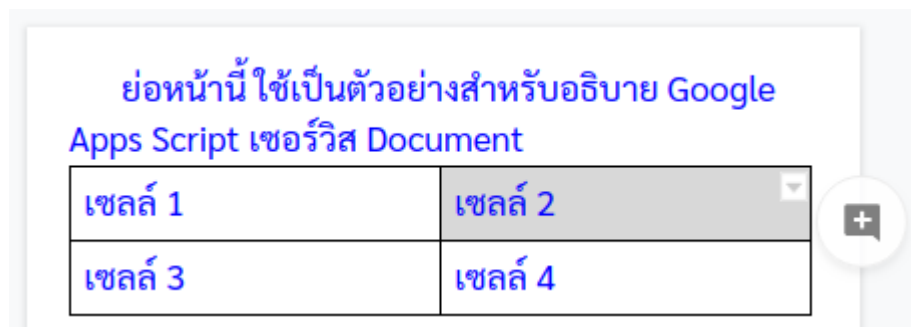
รันครั้งที่ 1 : ก่อนรันโค้ด เลือกข้อความบางส่วนในย่อหน้าไว้



ผลการรันครั้งที่ 1

[01] Element type: **TEXT**
[02] The character range begins at 18
[03] The character range ends at 25

รันครั้งที่ 2 : ก่อนรันโค้ด เลือกบางส่วนของเซลล์ในตารางไว้ก่อน



ผลการรันครั้งที่ 2

[01] Element type: **TABLE_CELL**
[02] The entire range element is included.

6.2. Methods ตระกูล Insert – ย้ายไป Element

`insertParagraph(childIndex, paragraph)`

<https://developers.google.com/apps-script/reference/document/body#insertParagraphchildIndex,-paragraph>

แทรกย่อหน้า ณ ตำแหน่งที่กำหนด

พารามิเตอร์

Name	Type	Description
<code>childIndex</code>	Integer	the index at which to insert
<code>paragraph</code>	Paragraph	the paragraph to insert

Class Paragraph

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

`deleteText(startOffset, endOffsetInclusive)`

<https://developers.google.com/apps-script/reference/document/text#deleteTextstartOffset,-endOffsetInclusive>

พารามิเตอร์

Name	Type	Description
<code>startOffset</code>	Integer	the character offset of the first character to delete
<code>endOffsetInclusive</code>	Integer	the character offset of the last character to delete

ตัวอย่าง

```
function example1() {  
  var body = DocumentApp.getActiveDocument().getBody() ;  
  // แทรก 2 ย่อหน้า โดยมีเส้น horizontal rule เป็นตัวขึ้น  
  body.insertParagraph(0, "An editAsText sample.") ;  
  body.insertHorizontalRule(0) ;  
  body.insertParagraph(0, "An example.") ;  
  
  // Delete " sample.\n\n An" removing the horizontal rule in the process.  
  // ลบข้อความ 'n editAsText'  
  body.editAsText().deleteText(14, 25) ;  
}
```

ผล

An example.

A sample.

บทที่ 7

Text



7.1. คลาส Text

คลาส Text

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

Element ที่บรรจุด้วย ข้อความ หรือ Text โดย Text ทุกตัวที่อยู่ในเอกสารถูกบรรจุใน Element ชนิด Text

Text เป็นส่วนประกอบเล็กๆ ภายในองค์ประกอบอื่นๆมากมาย เช่น เป็นส่วนประกอบของ Document, Equation, ListItem หรือ Paragraph เป็นต้น แต่ตัวมันเอง(Text) ไม่สามารถบรรจุอะไรได้เลย

เราใช้ Method ของคลาส Text ได้เยอะมาก อะไรที่มี Text ใช้ได้หมด

7.2. editAsText(), insertText()

editAsText() – Method ในคลาส Body

<https://developers.google.com/apps-script/reference/document/body.html#editastext>

แก้ไข Element ตัวนั้นในแบบ Text โดยจะคืนค่ากลับมาเป็น Text แต่ใช้ไม่ได้กับ Element ประเภท Non-text อย่าง InlineImage และ HorizontalRule

insertText(text) – Method ใน คลาส Position

<https://developers.google.com/apps-script/reference/document/position#inserttexttext>

insertText(offset, text) – Method ใน คลาส Text

<https://developers.google.com/apps-script/reference/document/text#inserttextoffset,-text>

ใช้แทรกข้อความ ณ ตำแหน่งที่ระบุ ซึ่งเป็นระยะออฟเซตตัวอักษร

Name	Type	Description
offset	Integer	the character offset at which to insert
text	String	the text to insert

ตัวอย่าง

```
function example() {
  var body = DocumentApp.getActiveDocument().getBody() ;

  // แก้ไข Body แบบ Text
  var text = body.editAsText() ;

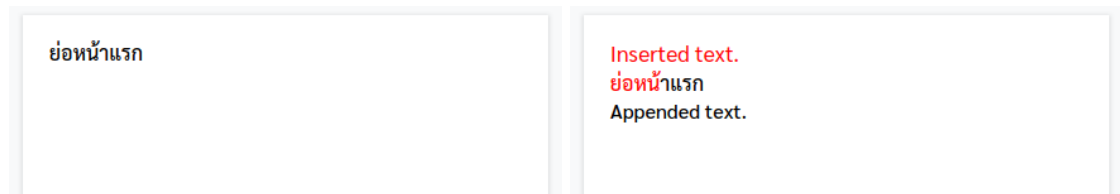
  // แทรก Text ตรงๆ ลงใน Body จะ Error
  // body.insertText(0, 'Inserted text.\n') ;

  // แทรกข้อความ Inserted text. ที่จุดแรก (0) + กด Enter
  text.insertText(0, 'Inserted text.\n') ;

  // แนบท้ายด้วย Text
  text.appendText('\nAppended text.') ;

  // ระบายสี เริ่มจาก ตัวแรก ถึง แบ่งครึ่ง Text ใน Body สีแดง
  text.setForegroundColor(0, text.getText().length / 2, '#FF0000') ;
}
```


ผล



7.3. deleteText()

`deleteText(startOffset, endOffsetInclusive)` - Method ใน คลาส Text

[https://developers.google.com/apps-script/reference/document/text#deleteText\(startOffset, -endOffsetInclusive\)](https://developers.google.com/apps-script/reference/document/text#deleteText(startOffset, -endOffsetInclusive))

ใช้ลบข้อความแบบเป็นเร้นจ์

พารามิเตอร์

Name	Type	Description
<code>startOffset</code>	Integer	the character offset of the first character to delete
<code>endOffsetInclusive</code>	Integer	the character offset of the last character to delete

ตัวอย่าง

```
function example() {  
  var body = DocumentApp.getActiveDocument().getBody() ;  
  
  // แทรก 2 ย่อหน้า ขึ้นโดยเส้น Horizontal rule  
  body.insertParagraph(0, "An editAsText sample.") ;           // ย่อหน้าที่ 2  
  body.insertHorizontalRule(0) ;                                   // เส้น hr  
  body.insertParagraph(0, "An example.") ;                       // ย่อหน้าแรก  
  
  // ลบอักขร "n editAsText" ในย่อหน้าที่ 2  
  body.editAsText().deleteText(14, 25) ;  
}
```

ผล



7.4. replaceText()

Replacing text

https://developers.google.com/apps-script/guides/docs#replacing_text

มาตรฐาน RE2

<https://github.com/google/re2>

replaceText([searchPattern](#), [replacement](#)) - Method ในคลาส Text

<https://developers.google.com/apps-script/reference/document/text#replacetextsearchpattern,-replacement>

ใช้ค้นหาและแทนที่ข้อความ โดยจะคืนค่ากลับมาเป็น Element เดิม

พารามิเตอร์

Name	Type	Description
searchPattern	String	the regex pattern to search for (ข้อความธรรมดา หรือ Regular Expression มาตรฐาน RE2)
replacement	String	ข้อความที่จะมาแทนที่

ตัวอย่าง

```
var body = DocumentApp.getActiveDocument().getBody();  
  
// ล้างย่อหน้าที่มีข้อความ Apps Script อยู่ด้วย  
// ให้เหลือเพียงข้อความ Apps Script เท่านั้น  
body.replaceText("^. * Apps ?Script. *$", "Apps Script");
```

ตัวอย่างที่ 2

โค้ดต่อไปนี้จะค้นหาข้อความ เช่น **{name}** จากนั้นแทนที่ด้วยค่าในตัวแปรแบบวัตถุ เช่น

`instructor.name` ค่าก็คือ **Wasan Khunnadiloksawet** เป็นต้น

ตัวอย่างโค้ดนี้ มักเอาไปใช้ทำ MailMerge

```
// พิมพ์ย่อหน้า สำหรับทดสอบแทนที่ข้อความ  
function createParas() {  
  
    var body = DocumentApp.getActiveDocument().getBody();  
  
    // พิมพ์ย่อหน้าที่มีข้อความต่างๆลงไป รวมแล้วมี 4 ย่อหน้า  
    body.appendParagraph('{name}') ;  
    body.appendParagraph('{address}') ;  
    body.appendParagraph('{city} {province}') ;  
    body.appendParagraph('{zip}') ;  
}  
  
// มีต่อ
```

```
// ต่อ
function Search_n_Replace() {
    var body = DocumentApp.getActiveDocument().getBody();
    var instructor = { name: 'Wasan Khunnadiloksawet' ,
                        address: '1 Moo 2 Nongpha' ,
                        city: 'Nongphai' ,
                        province: 'Petchaboon' ,
                        zip: 67140                } ;

    body.replaceText('{name}', instructor.name)
    .replaceText('{address}', instructor.address)
    .replaceText('{city}', instructor.city)
    .replaceText('{province}', instructor.state)
    .replaceText('{zip}', instructor.zip) ;
}
```


บทที่ 8

ภาพ



8.1. ใส่ภาพลงในตำแหน่ง Cursor

Insert Inline Image in a Google Document

<https://ctrlq.org/code/20078-insert-image-in-google-document>

insertInlineImage(**image**) – Method ในคลาส Position

<https://developers.google.com/apps-script/reference/document/list-item.html#insertinlineimagechildindex,-image>

สร้างและแทรกภาพ InlineImage ณ ตำแหน่งวัตถุ Position พารามิเตอร์ **image** ก็คือ Blob ของภาพ

insertImage(**childIndex**, **image**) – Method ในคลาส Body

สร้างและแทรกภาพ InlineImage ณ ตำแหน่ง **childIndex**(ตัวเลขตำแหน่งของ Child element ใน Body) พารามิเตอร์ **image** ก็คือ Blob ของภาพ insertImage() คืนค่ากลับมาเป็นคลาส InlineImage

ตัวอย่าง

```
function insertImage() {  
    // ภาพจากเว็บ  
    var image = "https://img.labnol.org/logo.png" ;  
    var blob = UrlFetchApp.fetch(image).getBlob() ;  
  
    // ภาพจาก Google Drive  
    var image = "https://drive.google.com/open?id=1inBoKWzWZr_ip2OvBAv3BWY7TbW9JDt" ;  
    // จับมาเฉพาะ ID ของภาพ โดยใช้ match ตัดตาม Regular Expression ที่ระบุ  
    var fileID = image.match(/[w\_-]{25,}/).toString() ;  
    // [w\_-] เฉพาะ ตัวเลขและตัวอักษร และ _ และ -  
    // {25,} ตั้งแต่ตัวที่ 25 เป็นต้นไป (นับตัวที่ Match [w\_-] )  
    var blob = DriveApp.getFileById(fileID).getBlob() ;  
  
    var doc = DocumentApp.getActiveDocument() ;  
    var cursor = doc.getCursor() ; // จับตำแหน่งของ Cursor  
  
    if (cursor) { // ถ้ามี Cursor ในเอกสาร – ใส่ภาพ ณ Cursor  
        cursor.insertInlineImage(blob).setWidth(300).setHeight(168.75) ;  
    } else { // ถ้าไม่มี Cursor  
        doc.getBody().insertImage(0, image).setWidth(300).setHeight(168.75) ;  
    }  
}
```

ผล

Cursor อยู่ก่อนย่อหน้านี้
ย่อหน้าที่มีอยู่แล้ว



Cursor อยู่ก่อนย่อหน้านี้
ย่อหน้าที่มีอยู่แล้ว