



เกิดอยากรจะเป็น

โปรแกรมเมอร์

ใช้ **Javascript** ง่ายๆ

อะไรล่ะ เริ่มกันเล้ยดีกว่า



โดย ธนศักดิ์ ตันติธารทอง

เกิดอยากจะเป็น โปรแกรมเมอร์ ด้วย Javascript

หนังสือสอน Javascript สำหรับผู้เริ่มต้นทุกคน

SaKko Sama

This book is for sale at <http://leanpub.com/soon-to-be-programmer>

This version was published on 2021-08-19



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2020 - 2021 SaKko Sama

Tweet This Book!

Please help SaKKo Sama by spreading the word about this book on [Twitter](#)!

The suggested hashtag for this book is [#soontobeprogrammer](#).

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

[#soontobeprogrammer](#)

สารบัญ

1.	Introduction	1
2.	การจัดเตรียมอุปกรณ์	3
2.1	Softwares ที่ใช้	3
	Atom	3
	Google Chrome	3
	Mac Terminal หรือ Windows Terminal	4
2.2	Environment Setup	5
	NVM (NodeJS)	5
	การลง NodeJS โดยกำหนด Version เอง	6
3.	Data types เบื้องต้น and Variables	9
3.1	Number	10
	Arithmetic Operator	13
3.2	String	14
3.3	Boolean	16
3.4	ค่าที่ false	19
	false	19
	null และ undefined	20
	NaN	20
3.5	Data types อื่นๆ	21
4.	if else elseif	22
4.1	การใช้ Terminal เบื้องต้น	22

4.2	if condition	22
4.3	else	22
4.4	else if	22
4.5	And &&	23
4.6	Or 	23
4.7	Multiple And Or	23
5.	Functions	24
5.1	ห้ดสร้าง Function	24
	Function แบบรับข้อมูล	24
5.2	console.assert	24
5.3	การส่งค่ากลับจาก Function	24
5.4	Arrow Function	25
6.	Built-in Functions	26
6.1	Number	26
	Number.isFinite(x)	26
	Number.parseInt(s)	27
6.2	Math	29
	Math.pow(base, exponent)	29
	Math.abs(x)	29
7.	var let const	31
7.1	var	31
7.2	let	31
7.3	const	31
7.4	using variable with function	31
7.5	Arrow function usage	32
7.6	Reserve Words	32
8.	Objects / Array	33
8.1	Object	33

	Object with function	33
8.2	Array	33
	Object with Array	33
8.3	Array of Objects	34
8.4	Array of anything	34
9.	For loops	35
9.1	Simple for loop	35
	แบบฝึกหัดสั้นๆ	35
9.2	Looping an Array	35
10.	While loops	36
10.1	While Loop	36
10.2	Do While	36
11.	User Input	37
11.1	readline module	37
12.	Date Time	38
12.1	Date Methods	38
	การดึงค่าจาก Date	38
	การเปลี่ยนค่าใน Date	38
13.	Node Package Manager (npm)	39
13.1	npm init	39
13.2	Moment JS	39
14.	Promise	40
14.1	resolve .then	40
14.2	reject .catch	40
15.	Async Await	41
15.1	Async Await	41

15.2	resolve values	41
15.3	reject values	41
	try catch finally	41
16.	Looking inside array	42
16.1	Find using for loop	42
16.2	Find using .filter	42
16.3	Find using .find	42
16.4	Pulling data from array using .map	42
17.	HTML with Javascript	43
17.1	HTML	43
17.2	Javascript in HTML	43
17.3	Javascript in action	43
18.	Easy Challenges	44
18.1	วิธีทำแบบฝึกหัด	44
18.2	เริ่มแบบฝึกหัด	44
19.	ไปต่อกันเถอะ	45

1. Introduction

หนังสือเล่มนี้ เหมาะสำหรับนักพัฒนา software มือใหม่ที่มีอุปกรณ์เพียงคอมพิวเตอร์ธรรมดาๆ หนึ่งเครื่อง และใช้ภาษา Javascript เนื่องจากเป็นภาษาที่ติดตั้งง่ายและทดสอบง่าย หลักการเขียนก็ง่าย ไม่เหมือนยุคที่ผมเรียนการเขียนโปรแกรมเมื่อยี่สิบปีที่แล้ว

เมื่ออ่านหนังสือเล่มนี้จบ ผม (ผู้เขียน) หวังว่าจะทำให้ทุกคนมีพื้นฐานมากพอที่จะไปเขียนโปรแกรมต่อ ไม่ว่าจะเป็น VueJS, ReactJS, AngularJS, ExpressJS, MeteorJS หรือ โปรแกรมอื่นๆที่ไม่ได้ใช้ Javascript

อุปกรณ์ขั้นต่ำที่ผู้อ่านต้องจัดเตรียมคือ

Computer

Mac (Apple Computer)

Windows 10

Ubuntu Linux

หากต้องการซื้อเครื่องใหม่ ผมแนะนำให้ใช้ Mac รุ่นใดก็ได้ โดยดูงบประมาณเป็นหลัก ซื้อเครื่องไหนไหว ก็ซื้อเครื่องนั้นพอ ไม่จำเป็นต้องแรงมาก ปัจจุบัน (2020) Macbook Air, Mac Mini สามารถหาซื้อได้ในราคา 25,000 บาทขึ้นไป และลดราคาลงอีกด้วย Education Discount

หากเป็นเครื่องมือสอง ซื้อเครื่องที่อยู่ในประกัน หรือ หากงบไม่ถึงให้ลองดูที่รุ่นที่มีอายุไม่เกิน 5 ปี ยิ่งใหม่ยิ่งดี อย่าซื้อเก่ามากเกินไป ท่องไว้ **เสียน้อยเสียยาก เสียมากเสียง่าย**

การเขียนโปรแกรมเป็นสิ่งที่สนุกได้ แต่ความสนุกนั้นๆ จะมาจากการที่เราได้เรียนรู้สิ่งใหม่ๆ บางทีโปรแกรมเราอาจจะพัง แต่เราก็สนุกไปกับการเรียนรู้ได้

ผมต้องการให้หนังสือเล่มนี้ ดึงดูดโปรแกรมเมอร์หน้าใหม่ ไม่ว่าจะเป็นเด็ก 10 ขวบ หรือ ผู้ใหญ่ทุกราย ได้เข้ามาลองหาความรู้ และใช้คอมพิวเตอร์ส่วนตัวให้มากกว่าแค่ทำงาน office ดูหนัง ฟังเพลง

หนังสือเล่มนี้จะสนุก เมื่อท่านรู้ว่าท่านกำลังทำสิ่งที่ที่คนหลายๆคนทำไม่เป็น และทุกๆครั้งที่ท่านฝึกฝน ท่านกำลังเก่งขึ้นเรื่อยๆ และมีอีกหลายๆคนที่ไม่รู้ด้วยซ้ำว่าท่านกำลังทำอะไรอยู่

ผมจะพยายามไม่ใช่ภาษาไทยในศัพท์ต่างๆ เพราะผมแปลไม่เป็นและเพราะว่าผมอยากให้ผู้อ่านเข้าใจว่า Programming language ส่วนใหญ่จะมีเอกสารเป็นภาษาอังกฤษ และอยากให้ผู้อ่านศึกษาภาษาอังกฤษด้วยในอนาคต เพื่อที่จะทดลองอะไรใหม่ได้ด้วยตนเอง

สุดท้ายนี้ ผมทำหนังสือเล่มนี้เพื่อความสุขของตนเอง หากพิมพ์ผิดหรือเนื้อหาผิดบางประการ รบกวนผู้อ่านที่มีความรู้มากกว่า ให้คำแนะนำด้วยครับ

ติดตามเนื้อหาอื่นๆได้ที่

<https://youtube.com/sakkosama>

ติดตามหนังสือได้ที่

<https://www.facebook.com/soon.to.be.programmer>

แนะนำหนังสือได้ที่

<https://www.facebook.com/groups/soon.to.be.programmer.book>

2. การจัดเตรียมอุปกรณ์

2.1 Softwares ทัวไป

Atom



ผมขอใช้โปรแกรม Atom เพื่อแก้ไข Code คุณสามารถ Download ได้ฟรีที่ <https://atom.io> หรือ คุณสามารถใช้ Text editor อื่นๆก็ได้ แล้วแต่นัด

สำหรับคนที่มาจาก notepad++ editplus หรือ editor อื่นๆ ก่อนใช้ atom ลองดู วิดีโอสอนใช้ text editor แบบไม่แพ้ใคร ได้ที่ <https://www.youtube.com/watch?v=pXmQLCidmpl>

ตัวอย่าง editor อื่นๆ

- VSCode
- Sublime Text

Text editor สมัยใหม่มากับ Add on เยอะแยะมากมายที่ช่วยให้เราเขียนโปรแกรมได้สนุกขึ้น แต่ว่าเราจะเริ่มเขียน Javascript แบบ Basic จึงยังไม่ต้องลงอะไรเพิ่มก็ได้ครับ

Google Chrome



ผมใช้ Chrome สำหรับค้นหา และ ทดสอบ Webpage คุณสามารถ Download ได้ฟรีที่ <https://www.google.com/chrome/>

Mac Terminal หรือ Windows Terminal



การทดสอบ code ต่างๆจะถูกรันผ่านโปรแกรมนี้ คุณจะต้องฝึกการใช้งานคอมด้วย command line ด้วย ซึ่งคำสั่งที่ใช้แรกๆนั้น มีประมาณ 5 คำสั่งเท่านั้นเอง

คุณจะต้องใช้ Program ที่ชื่อว่า Terminal บ่อยครั้ง ควรจะตั้ง shortcut ที่สามารถเปิดโปรแกรมได้อย่างรวดเร็วด้วย

2.2 Environment Setup

NVM (NodeJS)

ผมขอแนะนำให้ลง NodeJS ผ่าน NVM เนื่องจากว่าลงง่าย และ เหมาะสำหรับการสร้าง development environment ในอนาคต

ซึ่งการลง NVM นั้น จะลงตาม Operating System ของคุณ

การลง NVM อาจจะเจอปัญหาบ้าง ถ้าเป็นคอมใหม่ ไม่เคยเขียนโปรแกรมมาก่อน ส่วนใหญ่จะไม่มีปัญหา แต่ถ้าเคยลง node หรือ อะไรอื่นๆมาก่อนแล้ว อาจจะมีปัญหาบ้าง ถ้าเจอปัญหา ทางที่ง่ายที่สุดคือให้ล้างเครื่อง แล้วเริ่มทำจากศูนย์ ครับ

Mac OS / Ubuntu Linux

หากคุณใช้ Ubuntu Desktop แนะนำให้ update / upgrade เครื่องให้เป็น version ล่าสุดก่อน

หากคุณใช้ Mac OS คุณควรใช้ MacOS version ล่าสุด และ จะต้องลงโปรแกรม Xcode ก่อนที่จะลง NVM นะครับ

- โหลด Xcode ได้ที่ <https://apps.apple.com/us/app/xcode/id497799835>
- อ่านและฝึกทำเองได้ที่ <https://github.com/nvm-sh/nvm>
 - วิดีโอสอนลง nvm บน Mac / Linux <https://www.youtube.com/watch?v=k20Srsf2r7k>

Windows 10

หากคุณใช้ Windows 10 หากเป็นไปได้ อยากรให้ลอง update Windows version ล่าสุดก่อน

- Windows Terminal สามารถโหลดได้ที่ <https://www.microsoft.com/en-us/p/windows-terminal/9n0dx20hk701>

- อ่านและฝึกทำเองได้ที่ <https://github.com/coreybutler/nvm-windows>
 - วิดีโอสอนลง nvm บน Windows <https://www.youtube.com/watch?v=DA4csMg7XqM>

การลง NodeJS โดยกำหนด Version เอง

เมื่อลง NVM เสร็จแล้ว เราจะทำการลง NodeJS ต่อ โดยการลง NodeJS นั้น เราสามารถกำหนด Version ได้ด้วย

NodeJS มีหลาย version และ แต่ละ version มีคุณสมบัติแตกต่างกันไป สิ่งที่เรา กำลังฝึกฝนในหนังสือเล่มนี้จริง ๆ แล้วใช้ NodeJS version ไหนก็ได้ แต่ผมขอแนะนำให้ลง Version เดียวกัน จะได้ใช้ไม่เจอปัญหาอะไรแปลกๆครับ

ผู้เขียนจะใช้ NodeJS Version ล่าสุด **14.13.0** ณ เดือน ตุลาคม 2020 เพื่อสอนการลง Version **14.13.0** นั้น สามารถทำได้เมื่อลง NVM เสร็จแล้ว ด้วยคำสั่ง

Installing node

```
1 nvm install 14.13.0
```

เมื่อต้องการใช้งาน **14.13.0** ให้พิมพ์คำสั่ง

Using nvm to switch node version

```
1 nvm use 14.13.0
```

`nvm use VERSION` เป็นการสลับ version ไปมา หากเราลง NodeJS ผ่าน `nvm install VERSION` ไว้หลาย version ก็สลับไปใช้ version นั้นๆ ได้ทันที

เราสามารถเช็ค version node ปัจจุบันด้วยคำสั่ง

Checking current node version

```
1 node --version
```

เมื่อลงเสร็จเรียบร้อยแล้ว ทดสอบว่าเครื่องใช้งาน Node Console ได้โดยการพิมพ์คำสั่ง **node**

Entering node console

```
1 node
```

คุณ将会เห็น output ด้านล่าง ซึ่งเราจะเรียกมันว่า Node Console

Successfully opened Node Console

```
1 Welcome to Node.js v14.13.0.  
2 Type ".help" for more information.  
3 >
```

ตอนเปิด Terminal ขึ้นมา เราจะอยู่ใน shell ซึ่งเป็น default ของ Mac หรือ Windows แต่ว่าเวลาเราพิมพ์คำว่า **node** เราจะเข้าไปอยู่ใน Node shell
Node Console จะรับคำสั่งแค่ภาษา Javascript เท่านั้น แรกๆอาจจะสับสนกับ Shell ของ Terminal และ Node แต่ถ้าสังเกตดีๆ Node Console จะมีเพียงแค่ > นำหน้า ไม่เหมือนกับ Shell ซึ่งอาจจะเป็น C:\> ใน Windows หรือ ~ ใน \$Mac / Linux

คุณสามารถพิมพ์คำสั่งต่างๆลงไปหลังเครื่องหมาย > ได้ ยกตัวอย่างเช่น พิมพ์คำว่า **1+1** แล้วกด enter จะมี output เป็นเลข 2 และเครื่องหมาย > จะขึ้นมาเพื่อให้คุณพิมพ์คำสั่งต่อไปที่ต้องการ

Using node console

```
1 > 1+1
2 2
3 > .exit
```

คุณสามารถออกจาก Node Shell เพื่อกลับมา Terminal Shell ง่าย ๆ ด้วยการพิมพ์คำว่า `.exit` (อย่าลืมพิมพ์จุดด้วย) หรือ กด `ctrl+c` สองที

ยินดีด้วยครับ แค่นี้คุณก็เกือบจะเป็น Programmer แล้ว

3. Data types เบื้องต้น and Variables

Computer เกิดมาเพื่อคำนวณ การที่จะคำนวณได้ ก็ต้องจัดเก็บข้อมูลได้

ในบทนี้ ผมขออธิบายการเก็บข้อมูลต่างๆของ Javascript เนื่องจากว่าเราจะต้องคำนวณค่าต่างๆ และผลลัพธ์ที่ได้มา จะออกมาเป็น Data types เหล่านี้

เราจะใช้ command **node** เพื่อเข้าไปทดสอบคำสั่งต่างๆของ javascript หากคุณเห็นเครื่องหมาย > นำหน้า จะเป็นการใช้ Node console ซึ่งผู้อ่านจะต้องเปิด terminal แล้วพิมพ์คำว่า **node** สามารถดูได้ใน บท Preparation



การเก็บข้อมูลในบทนี้ เป็นการเก็บข้อมูลในหน่วยความจำของ Node Process ที่เราเปิดผ่าน Node Console ทำให้เวลาเราพิมพ์ **.exit** ข้อมูลที่เราเคยพิมพ์ไว้จะหายไปจาก Memory วิธีการเก็บข้อมูลให้ไม่หายไป อาจจะต้องพึ่งการเก็บข้อมูลลง File หรือ Database ซึ่งเรายังไม่ใช้ในตอนนี้

3.1 Number

Number คือตัวเลข โดยตัวเลขนั้นจะมีสองประเภทคือ

integer

จำนวนเต็ม เช่น 0, 1, 1, 2, 3, 5, 8, 13, 21, 34

floating / decimal

เลขทศนิยม เช่น 3.14159, 2.7182

การใช้งาน สามารถนำมาใช้ได้ง่ายๆโดยการสั่ง

```
1 > 1
2 1
3 > 2.5
4 2.5
5 > 1+15
6 16
7 > 3.1+3.1
8 6.2
```

ทุกครั้งที่เราพิมพ์คำสั่งหลังเครื่องหมาย > บรรทัดต่อไปจะมีบางอย่าง return กลับมา เช่น

- บรรทัดที่ 1 - ถ้าเราพิมพ์ > 1 บรรทัดต่อไปจะเป็น 1 ซึ่งเป็นผลลัพธ์ของการไม่ทำอะไรเลยกับตัวเลข > 1
- บรรทัดที่ 3 - ถ้าเราพิมพ์ > 2.5 บรรทัดต่อไปจะเป็น 2.5 ซึ่งเป็นผลลัพธ์ของการไม่ทำอะไรเลยกับตัวเลข > 2.5
- บรรทัดที่ 5 - ถ้าเราพิมพ์ > 1+15 บรรทัดต่อไปจะเป็น 16 เป็นผลลัพธ์ของการทำ > 1+15

ตัวเลข 16 ที่เราได้จากบรรทัดที่ 6 คือผลลัพธ์หรือเราจะเรียกมันว่า return หรือ response ก็ได้



ถ้าอ่านตรงนี้ไม่เข้าใจ ข้ามไปก่อนก็ได้

- การ $+$ $-$ $*$ $/$ (บวกลบคูณหาร) เลข decimal บางครั้ง จะมีปัญหาเรื่อง floating point ซึ่งเป็นสิ่งที่หลีกเลี่ยงไม่ได้ และต้องระมัดระวังเป็นอย่างมาก ยกตัวอย่างเช่น $3.1 + 3.2$ จะได้ค่าเป็น 6.300000000000001 ผู้อ่านสามารถศึกษาเพิ่มเติมได้โดยการค้นหาคำว่า floating point problem หรืออ่านคำอธิบายเบื้องต้นได้ที่ <https://floating-point-gui.de/basic/>
- การแก้ไข floating point problem ทำได้หลายวิธี แต่ส่วนใหญ่จะเป็นการแก้ไขด้วยการกำหนดจำนวนจุดทศนิยม ที่ต้องการนำมาแสดงบนหน้าจอ หรือการ round up/down แล้วแต่ความต้องการของโปรแกรม

ผลลัพธ์ของการคำนวณนั้นมีมากกว่าตัวเลขได้ ยกตัวอย่างเช่น

```
1 > 1 / 0
2 Infinity
3 > "a" / 2
4 NaN
```

การนำ 1 หารด้วย 0 จะได้ Infinity

การนำตัวหนังสือ หารด้วยตัวเลข หารไม่ได้ก็เลยกลายเป็น NaN ย่อมาจาก Not a Number แปลว่า ไม่ใช่ตัวเลข

การเขียนโปรแกรมใน Javascript นั้น เวลาเกิดปัญหาด้านการคำนวณเลข โปรแกรมจะไม่ขึ้น Error เหมือนกับภาษาอื่นๆ เช่น การนำ $1/0$ ใน ภาษา C โปรแกรมจะ crash ทันที

การที่โปรแกรมเราไม่ Crash ตอนที่ทำการหารเลขด้วย 0 ของ Javascript นั้น เรานิยามมันในภาษาอังกฤษว่า Mathematical operations are safe

เมื่อเราคำนวณเสร็จแล้ว เราสามารถนำผลลัพธ์ไปเก็บไว้ผ่าน ตัวแปร (Variable) ที่เรากำหนดขึ้นเองได้ด้วย ยกตัวอย่างเช่น

```
1 > a = 10
2 10
3 > b = 20
4 20
5 > c = a + b
6 30
7 > c
8 30
9 > a
10 10
```

บรรทัดที่ 1 - a เท่ากับ 10

บรรทัดที่ 3 - b เท่ากับ 20

บรรทัดที่ 5 - c เท่ากับ a บวก b

บรรทัดที่ 7 - พิมพ์ c เสร็จๆ โปรแกรมจะบอกว่าค่าของ c ได้ค่าเท่ากับ 30

บรรทัดที่ 9 - หากลองพิมพ์ a เสร็จๆดู ก็จะเห็นว่า a ยังมีค่าเท่ากับ 10

การกำหนด Variable นั้นเป็น Case Sensitive หมายถึง `a = 10` และ `A = 20` นั้น `a` และ `A` ถือเป็นคนละตัวแปรกัน

ฉะนั้นการตั้ง Variable ควรจะตั้งให้สื่อความหมายเช่น ตั้งเป็นตัวเล็กนำหน้า `firstName` หรือ `lastName` วิธีการตั้งตัวแปรแบบนี้เป็นมาตรฐานของ Javascript

การตั้งตัวแปรไม่ได้มีกฎตายตัว เพียงแต่ทำตามๆ กันกับคนส่วนใหญ่เพื่อกันการสับสนในการถามตอบปัญหาต่างๆหรือการแบ่งปันความรู้ ดังนั้นภาษาอื่นๆ การตั้ง Variable อาจจะเป็น `first_name` หรือ `last_name` หรือรูปแบบอื่นๆ ก็เป็นไปได้

- camelCase คือ การเริ่มต้นตัวอักษรด้วยตัวเล็ก และ ทุกๆการเว้น space จะใช้ตัวใหญ่
- PascalCase คือ การเริ่มต้นตัวอักษรด้วยตัวใหญ่ และ ทุกๆการเว้น space จะใช้ตัวใหญ่

- underscore_case หรือ snake_case คือ อักษรตัวเล็กทุกตัว แต่ space จะเปลี่ยนเป็น underscore
- UPPER_CASE คือ ทุกตัวตัวใหญ่หมด และคั่นด้วย underscore

Arithmetic Operator

ใน Javascript เรามี Operator ให้เลือกใช้นอกจาก บวกลบคูณหาร ด้วย ซึ่งในตารางด้านล่างจะเป็นสิ่งที่เราใช้งานบ่อยๆ

Operator	Description
+	บวก
-	ลบ
*	คูณ
/	หาร
**	ยกกำลัง เช่น $10^{**}2$ คือ 10 ยกกำลัง 2
%	Modulus หรือการหาเศษ เช่น $20 \% 8$ จะ เหลือ 4
++	เพิ่มค่า +1 เช่น $x++$ เปรียบเสมือน $x=x+1$
--	ลดค่า -1 เช่น $x--$ เปรียบเสมือน $x=x-1$

3.2 String

การเก็บค่า ตัวอักษร คำศัพท์ ประโยค ต่างๆ จะสามารถเก็บได้โดยการใส่เครื่องหมาย single quotation mark ' หรือ double quotation mark " ครอบสิ่งนั้นๆไว้ ยกตัวอย่างเช่น

```
1 > 'hello, world'
2 'hello, world'
3 > "hello, world"
4 'hello, world'
```

ข้อสังเกตคือ ไม่ว่าเราจะใช้ " หรือ ' ตัว Node จะ return (บรรทัดต่อไป) ให้ค่ากลับมาเป็น single quote ' ครับ

คุณสามารถทำการบวกตัวหนังสือได้เช่นกัน ซึ่งในภาษา Javascript นั้นจะเทียบกับการนำ String มา ‘ต่อ’ (concatenate) กัน

```
1 > a = "hello"
2 'hello'
3 > b = ", "
4 ', '
5 > c = "world"
6 'world'
7 > d = a + b + " " + c
8 'hello, world'
```

หากการบวก string นั้นเข้าใจยากและอ่านยาก เราสามารถทำการ format string ได้โดยการใช้เครื่องหมาย Backticks ‘ แทน quotation และเพิ่ม Variable ที่ต้องการ format เป็น String เดียวกันใน \${} ยกตัวอย่างเช่น



เครื่องหมาย Backtick อยู่บน keyboard โดยกด **shift+~**

```

1 > a = "hello"
2 'hello'
3 > b = ","
4 ','
5 > c = "world"
6 'world'
7 > `${a}${b} ${c}`
8 'hello, world'

```

- `\${a}\${b} \${c}` คือการนำ a + b + " " + c แต่ว่าทำใน operation เดียว ซึ่งเร็วกว่า และอ่านง่ายกว่า

- `\${a}\${b} \${c}` สามารถเขียนอีกวิธี `\${a+b} \${c}` แบบนี้ก็ได้อีกเช่นกัน โดยที่เราให้ a+b ก่อน แล้วค่อยออกมา format ข้างนอกกับ c อีกครั้งหนึ่ง

3.3 Boolean

ผู้อ่านอาจจะเคยได้ยินว่า Computer นั้นต้องทำอะไรหลายๆอย่างเกี่ยวกับ 0 และ 1 ซึ่งทั้งสองสิ่งนี้ ถ้าอยู่ในตัวเลขฐานสอง (Binary) นั้น 0 คือ false และ 1 คือ true

ค่า true และ false นั้นเรียกว่า Boolean โดยการใช้ Boolean วิธีการใช้งานง่ายๆคือ

```
1 > true
2 true
3 > false
4 false
```

Boolean นั้นใช้งานได้หลากหลายรูปแบบ และเป็นสิ่งหนึ่งที่สำคัญมากในการเขียนโปรแกรม เนื่องจากว่าส่วนใหญ่จะเป็นผลลัพธ์ของการเปรียบเทียบข้อมูลต่างๆ ผ่าน operator ที่ใช้เปรียบเทียบ ยกตัวอย่างเช่น

Operator	คำอธิบาย
==	เท่ากับ
===	เท่ากับ แต่ว่า strictly equal
!=	ไม่เท่ากับ
!==	ไม่เท่ากับ แต่ว่า strictly unequal
<	น้อยกว่า
<=	น้อยกว่าเท่ากับ
>	มากกว่า
>=	มากกว่าเท่ากับ

ทดสอบการเปรียบเทียบค่าได้ง่ายๆดังนี้ (ลองทำที่ละบรรทัด)

```
1 > false == false
2 true
3 > true == true
4 true
5 > false != true
6 true
7 > 0 == false
8 true
9 > 1 == true
10 true
11 > 0 === false
12 false
13 > 1 === true
14 false
15 > 100 > 20
16 true
17 > 20 > 20
18 false
19 > 20 >= 20
20 true
21 > 10 < 100
22 true
23 > 10 < 10
24 false
25 > 10 <= 10
26 true
27 > 'a' == 'a'
28 true
29 > 'a' === 'a'
30 true
31 > 'a' != 'b'
32 true
33 > 'a' !== 'b'
34 true
```

- strictly equal คือการเปรียบเทียบค่าจริงๆของที่เราต้องการเปรียบเทียบ เช่น นิยามของ 0 คือ false หากเปรียบเทียบ `0 == false` จะได้ true เพราะว่า 0 และ false นิยามคือ no เหมือนกัน แต่ว่าหากใช้ `0 === false` จะได้ false เพราะค่าจริงๆของ 0 และ false ไม่เท่ากัน
- การเขียนโปรแกรมที่ดี ควรจะตรวจสอบค่าจาก type จริงๆของสิ่งที่เราอยากจะเปรียบเทียบ ฉะนั้น ควรจะใช้ `strictly equal ===` และ `!==` ในการตรวจสอบค่าเสมอ

true และ false นั้นไม่เท่ากับ String "true" และ "false" เพราะค่า true และ false นั้นเป็น datatype พิเศษ ตัวอย่างเช่น

```
1 > true == 'true'
2 false
3 > true === 'true'
4 false
5 > false == 'false'
6 false
7 > false === 'false'
8 false
```

เราสามารถนำค่าที่ได้จากการเปรียบเทียบเก็บไว้ใน Variable ได้ เหมือนกับ Data types อื่นๆ

```
1 > a = 10
2 10
3 > b = 20
4 20
5 > c = a < b
6 true
7 > c
8 true
```

กรณีนี้คือ c เท่ากับ `a < b` หรือ `10 < 20` ซึ่งเป็น true ทำให้ c มีค่าเป็น true

3.4 ค่าที่ **false**

ค่าที่ **false** คือค่าที่ Javascript มองว่าเป็น **false** เมื่อไปใช้ในบทย่อยๆ เช่น **if/else** ค่าด้านล่างจะมีความเป็น **false** อยู่

- **false** — Boolean false
- **0** — ตัวเลขศูนย์
- **""** — String ว่างๆ
- **null** - ค่าว่างๆ กำหนดไว้ว่าให้ว่าง
- **undefined** - ไม่มีค่า ไม่เคยถูกกำหนด
- **NaN** — Not a Number

false

ค่า **false** เป็นเคสพิเศษที่เมื่อเปรียบเทียบกับค่า **0** และ **""** ด้วย **==** แล้วจะได้ค่าเป็น **true** นี่เป็นสิ่งที่ต้องจำ เพราะว่าเป็นเคสที่เกิดขึ้น กับ javascript เท่านั้น

```
1 > false == 0
2 true
3 > 0 == ""
4 true
5 > "" == false
6 true
```

หากต้องการเปรียบเทียบค่าจริง ควรจะใช้ **===**

```
1 > false === 0
2 false
3 > 0 === ""
4 false
5 > "" === false
6 false
```

null และ undefined

ค่าว่างทั้งสองแบบ เปรียบเทียบด้วย == แล้วได้ true แก้ไขได้โดยการใช้ ===

```
1 > null == null
2 true
3 > undefined == undefined
4 true
5 > null == undefined
6 true
```

NaN

เป็นค่าพิเศษที่ Javascript มองว่าเป็นทั้ง false แต่ว่าเมื่อเปรียบเทียบค่าทั้ง false อื่นๆ จะได้ค่าเป็น false ซึ่งไม่เหมือนเคสด้านบน ยกตัวอย่างเช่น

```
1 > NaN == null
2 false
3 > NaN == undefined
4 false
5 > NaN == NaN
6 false
```

ข้อสังเกตคือ `NaN == NaN` และ `NaN === NaN` จะเป็น `false` เสมอ เพราะว่าเป็นการเปรียบเทียบสิ่งที่ไม่ใช่ตัวเลข กับ สิ่งที่ไม่ใช่ตัวเลข แต่ว่าสิ่งที่ไม่ใช่ตัวเลขนั้นเป็นอะไรไม่มีใครรู้ ก็เลยออกมาเป็น `false`

3.5 Data types อื่นๆ

จริงๆ แล้ว Data types ใน Programming นั้นมีอีกหลายชนิดเช่น bytes, short, long, double, float และอีกมากมาย แล้วยังมี Object เข้ามาเกี่ยวข้องอีก

แต่เริ่มจะข้อมูลเยอะเกินไป กลัวผู้อ่านไม่สนุก เดี่ยวเราไปทำอย่างอื่นกันก่อน แล้วค่อยกลับมาดู Datatype อื่นๆต่อกันครับ

ไปเขียนโปรแกรมกันดีกว่า

4. if else elseif

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

4.1 การใช้ Terminal เบื้องต้น

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

4.2 if condition

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

4.3 else

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

4.4 else if

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

4.5 And &&

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

4.6 Or ||

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

4.7 Multiple And Or

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

5. Functions

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

5.1 หั้ดสร้าง Function

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

Function แบบรับข้อมูล

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

5.2 console.assert

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

5.3 การส่งค่ากลับจาก Function

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

5.4 Arrow Function

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

6. Built-in Functions

ภาษาต่างๆ จะมี Functions ที่เตรียมมาให้เราใช้งาน โดยที่เราไม่ต้องเขียนเอง ยกตัวอย่างเช่น `Number.isInteger()` ในบทที่แล้ว

ในบทนี้ ผู้เขียนนำตัวอย่างบางส่วนมา เพื่อให้ผู้อ่านได้ทดลองอ่านและลองทำตาม การเขียนโปรแกรมนั้นจะต้องฝึกฝนการใช้ Function ของคนอื่นอยู่ตลอดเวลา ช่วงแรกๆนั้น อาจจะใช้เวลาทำความเข้าใจ และพิมพ์ผิดๆถูกๆ หากสับสนข้อไหน ผู้เขียนจะจัดเตรียม Facebook Group เอาไว้ให้



- ณ ตอนนี้อ่านบทความต่างๆ จะมีการอ้างอิงถึง Object ซึ่งผู้เขียนยังไม่ได้อธิบาย และ จะอธิบายในบทต่อไป ไม่ต้องใจร้อน เดียวได้ทำแน่นอนครับ
- ผมนำ function อ้างอิงมาจาก Mozilla คุณสามารถดูได้ว่ามี Function อะไรบ้างที่ Javascript เตรียมไว้ให้แล้ว https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects
- การใช้ Function นั้นจริงๆแล้วไม่จำเป็นต้องใช้ของที่ Javascript เตรียมไว้ให้ก็ได้ เราอาจจะไป Import Library ของคนอื่นมาใช้งานก็ได้

6.1 Number

ผู้เขียนขอยกตัวอย่างที่เข้าใจง่ายสำหรับการใช้ Built-in Functions ที่มากับ Javascript โดยการใช้ Number สำหรับคนที่ไม่ได้ศึกษาเลขมา ผมจะพยายามอธิบายคำศัพท์ให้แต่ว่า หากคำไหนไม่เข้าใจ ข้ามไปก่อนก็ได้ครับ

Number.isFinite(x)

: `isFinite` นั้นแปลตรงตัวว่า ตัวเลขตัวนี้เป็นประเภท Finite หรือไม่ ด้วยคำว่า `is` ที่นำหน้า Function ทำให้เรารู้ได้ว่าผู้ตั้งชื่อ function กำลังจะ `return` ค่าประเภท `boolean` ออกมาให้เรา

คำว่า Finite คือ จำนวน จำกัด หรือตัวเลขที่กำหนดค่าได้ ยกตัวอย่างเช่น ตัวเลขทั่วไป 1,2,3 หรือ ตัวเลขทศนิยม 3.14

คำว่า Infinite คือ จำนวนที่ไม่มีวันสิ้นสุด ใหญ่กว่าที่เราจะจินตนาการได้ ยกตัวอย่างเช่น ผลลัพธ์ของการหารด้วยศูนย์ หากไม่เข้าใจ ให้จินตนาการว่า เราจะต้องแบ่งอะไรบางอย่างให้เป็นจำนวน x ส่วน หาก x มีค่ามากกว่าศูนย์การแบ่งนั้นสามารถทำได้ แต่ถ้าแบ่งออกเป็นจำนวนศูนย์ส่วนมันเล็กมากๆจนไม่รู้จะสิ้นสุดที่เท่าไร จึงเป็น Infinity

NaN เป็นส่วนหนึ่งของ Programming ซึ่งบ่งบอกว่าสิ่งที่เราได้มานั้นไม่ใช่ตัวเลข ย่อมาจาก Not a Number

Number.isFinite()

```
1 > Number.isFinite(1 / 0);  
2 false  
3 > Number.isFinite(10 / 5);  
4 true  
5 > Number.isFinite(0 / 0);  
6 false  
7 > Number.isFinite(NaN);  
8 false
```

Number.parseInt(s)

: การ parse คือการพยายามแปลงค่าจากสิ่งหนึ่งเป็นอีกสิ่งหนึ่ง ในกรณีนี้คือ จะแปลง String เป็น Integer

ในโปรแกรมมิ่งนั้น Number กับ String นั้นทำงานต่างกัน ยกตัวอย่างเช่นถ้าเราเอา $1+1$ จะได้ 2 แต่ถ้าเราเอา `'1'+'1'` จะได้ `'11'`

บางครั้งสิ่งที่เราเห็นนั้น อาจจะไม่ใช่ data type ที่เราคิด เราควรจะระมัดระวังในการ บวก ลบ คูณ หาร ตัวเลขมากๆ เพราะว่าผลลัพธ์ที่ได้นั้น แตกต่างกันโดยสิ้นเชิง

Number.parseInt()

```
1 > Number.parseInt("1");
2 1
3 > "1" + Number.parseInt("1");
4 '11'
5 > Number.parseInt("1") + "1";
6 '11'
7 > Number.parseInt("1") + Number.parseInt("1");
8 2
9 > Number.parseInt("10,000");
10 10
11 > "10,000".replace(",", "");
12 '10000'
13 > Number.parseInt("10,000".replace(",", "")); //
14 10000
```

คำอธิบาย

บรรทัดที่ 3 String + Number
บรรทัดที่ 5 Number + String
บรรทัดที่ 7 Number+Number
บรรทัดที่ 9 ,000 โดนตัดทิ้งไป
บรรทัดที่ 11 ตัวอย่างการเอา , ออกจาก string โดย replace , ด้วยค่าว่าง
บรรทัดที่ 13 เอา , ออกก่อนค่อย parseInt

สำหรับ Function ของ Number ผู้อ่านสามารถอ่านต่อได้ที่นี้

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Number

6.2 Math

สำหรับการเขียนโปรแกรมนั้น สิ่งที่น่าสนใจได้ยากน่าจะเป็นการคำนวณเลข เช่น การคำนวณเงิน ซึ่งคิดว่า programmer หลายคน สุดท้ายแล้วก็ต้องเขียนเรื่องนี้บ้าง ผู้เขียนขอยกตัวอย่างบางส่วนเกี่ยวกับ Math เพื่อใช้งานในอนาคต

Math.pow(base, exponent)

: เลขยกกำลัง ยกตัวอย่างเช่น 2 ยกกำลัง 8 คือ 2 คูณกัน 8 ครั้ง $2*2*2*2*2*2*2*2$ ได้ 256 หรือ เขียนใน Javascript ดังนี้

Math.pow()

```
1 > Math.pow(2, 8)
2 256
```

Math.pow เป็น function ที่รับค่าสองค่าคือ base ที่เราใส่เลข 2 และ exponent คือจำนวนยกกำลังที่เราใส่ 8

Math.abs(x)

: การทำ absolute ให้กับตัวเลข ในคณิตศาสตร์นั้นใช้เครื่องหมาย $|x|$ คือ ไม่ว่า x จะเป็นค่าลบหรือบวก ให้เปลี่ยนค่าเป็นบวกให้หมด

โดยตาม documentation แล้ว Function **Math.abs** จะรับค่าแค่ตัวเลขเท่านั้น หากพยายามส่ง String เข้าไป ค่าอาจจะผิดเพี้ยนได้



เช่น $|-10|$ หรือ $|10|$ จะได้ 10 เหมือนกัน

Math.abs()

```
1 > Math.abs(-10)
2 10
3 > Math.abs(10)
4 10
5 > Math.abs("10")
6 10
7 > Math.abs("10,000")
8 NaN
```

คำอธิบาย

บรรทัดที่ 5 ส่ง string “10” ได้
บรรทัดที่ 7 แต่ส่ง string “10,000” จะไม่ได้

การใช้ Function built-in นั้นเป็นสิ่งที่ง่าย แต่ว่าเราอาจจะลองหัดเขียน function เหล่านี้เองก็ได้ ยกตัวอย่างเช่นเราต้องการเขียน function absolute เอง

absolute(x)

```
1 function absolute(x) {
2   if (x > 0) {
3     return x;
4   } else {
5     return x * -1;
6   }
7 }
```



ข้อควรระวัง การเขียน function ควรจะตรวจสอบค่าที่ได้รับมาโดยละเอียดก่อน ตัวอย่างด้านบนเป็นการเขียนให้ดูเท่านั้น ถ้าจะใช้จริงควรจะเขียนให้ดีกว่านี้

7. var let const

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

7.1 var

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

7.2 let

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

7.3 const

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

7.4 using variable with function

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

7.5 Arrow function usage

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

7.6 Reserve Words

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

8. Objects / Array

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

8.1 Object

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

Object with function

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

8.2 Array

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

Object with Array

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

8.3 Array of Objects

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

8.4 Array of anything

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

9. For loops

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

9.1 Simple **for** loop

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

แบบฝึกหัดสั้นๆ

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

9.2 Looping an Array

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

10. While loops

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

10.1 While Loop

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

10.2 Do While

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

11. User Input

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

11.1 readline module

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

12. Date Time

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

12.1 Date Methods

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

การดึงค่าจาก Date

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

การเปลี่ยนค่าใน Date

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

13. Node Package Manager (npm)

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

13.1 npm init

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

13.2 Moment JS

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

14. Promise

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

14.1 resolve .then

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

14.2 reject .catch

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

15. Async Await

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

15.1 Async Await

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

15.2 resolve values

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

15.3 reject values

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

try catch finally

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

16. Looking inside array

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

16.1 Find using for loop

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

16.2 Find using **.filter**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

16.3 Find using **.find**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

16.4 Pulling data from array using **.map**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

17. HTML with Javascript

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

17.1 HTML

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

17.2 Javascript in HTML

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

17.3 Javascript in action

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

18. Easy Challenges

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

18.1 วิธีทำแบบฝึกหัด

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

18.2 เริ่มแบบฝึกหัด

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/soon-to-be-programmer>.

19. ไปต่อกันเถอะ

สำหรับคนที่อ่านถึงบทนี้แล้ว ทำแบบฝึกหัดมาบ้างแล้ว

ผมเชื่อว่าคุณสามารถลองเขียนโปรแกรมง่ายๆ เป็นเว็บได้แล้วครับ

ลองไปดูที่ youtube channel ของผมเอง เพื่อทำอะไรสนุกๆกันนะครับ

สำหรับทุกๆคน ผมขอแนะนำให้เริ่มจาก VueJS และ NuxtJS นะครับ

สามารถดูได้ที่ Youtube Channel ของผมได้เลยครับ

<https://youtube.com/sakkosama>

หวังว่าทุกคนจะสนุกกับการเป็นโปรแกรมเมอร์ครับ

อย่าเพิ่งเบื่อกันนะครับ อาชีพนี้ non-stop learning ครับ