

Algorithm, Flowchart and Pseudo code

Algorithm

Algorithm คือ กระบวนการและขั้นตอนในการคิดวิเคราะห์ และแก้ไขปัญหาที่ช่วยเพิ่มประสิทธิภาพในการทำงาน โดยใช้หลักการคิดแบบเป็นขั้นเป็นตอน

เราสามารถใช้อัลกอริทึมเพื่อแก้ปัญหาอะไรก็ได้ ไม่จำกัดว่าจะต้องเป็นปัญหาเกี่ยวกับการเขียนโปรแกรมคอมพิวเตอร์เพียงอย่างเดียวเท่านั้น โดยที่เราสามารถอธิบายการทำงานได้หลายวิธี เช่น

- Natural Language เป็นการอธิบายโดยใช้ภาษาที่ใช้พูดกันโดยทั่วไป
- Pseudo code เป็นการอธิบายผ่านการใช้อักษรเทียม
- Flowchart เป็นการอธิบายผ่านแผนภาพ

คุณสมบัติพื้นฐานของอัลกอริทึม

1. จะต้องไม่กำกวม เข้าใจง่าย
2. จะต้องช่วยแก้ปัญหาได้อย่างถูกต้อง
3. ขั้นตอนต่าง ๆ ควรมีความเรียงง่ายและมีประสิทธิภาพ
4. จะต้องมียุคสิ้นสุด

ตัวอย่าง

อัลกอริทึมในการเดินทางจากคณะไปเชียงใหม่

1. จองตั๋วสุวรรณภูมิ – เชียงใหม่
2. นั่งรถสองแถวจากคณะไปแอร์พอร์ตลิงก์
3. ขึ้นแอร์พอร์ตลิงก์ไปสนามบิน
4. เช็คอินที่สนามบิน
5. ขึ้นเครื่องบิน
6. ถึงเชียงใหม่

แบบฝึกหัด

1. การทำไข่เจียว

1. ตอกไข่ใส่ถ้วย
2. ตีไข่
3. ใส่น้ำปลา หรือ ซอสต่างๆ
4. ตั้งน้ำมันบนกระทะให้เดือด
5. เทไข่ลงไปในน้ำมันที่เดือด
6. รอสุกพักแล้วพลิกไข่
7. พอทอดสุกพักก็เอาขึ้นจาน
8. ตักข้าวพร้อมรับประทาน

(ไม่ได้มีเพียงวิธีนี้วิธีเดียว)

2. การเดินทางมาคณะของน้อง ๆ

1. ตื่นนอน
2. เดินออกมารอสองแถวสีม่วงไปเทคโนโลยี
3. พอรถมาก็ขึ้นรถ
4. รอรถมาถึงคณะก็กดกริ่งลง
5. เดินข้ามสะพานลอย
6. ถึงคณะแล้ว

(ไม่ได้มีเพียงวิธีนี้วิธีเดียว)




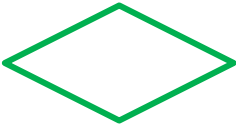

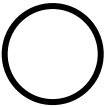
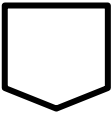
3. การทำ factorial

1. รับค่าตัวเลขมา 1 ตัวเพื่อหา factorial
2. นำตัวเลขนั้นมาคูณกับตัวเลขนั้นลบ 1
3. นำคำตอบที่ได้จากข้อสองไปคูณกับตัวเลขที่ลบ 1 ไปแล้วมาลบ 1 อีกรอบ
4. ทำซ้ำข้อสองกับสามไปเรื่อยๆจนตัวเลขนั้นเหลือ 1
5. ได้คำตอบ

(ไม่ได้มีเพียงวิธีนี้วิธีเดียว)

Flowchart

Flowchart คือแผนภาพที่ใช้แสดงลำดับขั้นตอนการทำงานของอัลกอริทึมด้วยการนำเสนอผ่านสัญลักษณ์สากลที่เป็นมาตรฐาน เพื่อให้สื่อความหมายและเกิดความเข้าใจที่ตรงกัน

สัญลักษณ์	ชื่อเรียก	ความหมาย
	Terminal	จุดเริ่มต้นและจุดสิ้นสุด
	Data	การรับข้อมูลเข้า / ส่งออกข้อมูล
	Process	การปฏิบัติงาน / ประมวลผล
	Decision	การตัดสินใจ
	Flow line	แสดงทิศทางการทำงาน
	Connector	จุดเชื่อมต่อของผังงาน
	Off-page Connector	จุดเชื่อมต่อระหว่างหน้า

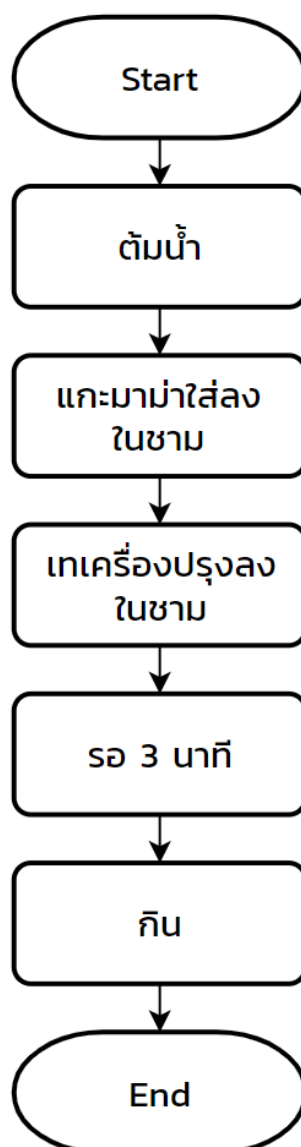
การเขียน Flowchart

1. แผนภาพต้องมีจุด Start (จุดเริ่มต้น) และ End (จุดสิ้นสุด) เสมอ
2. ในแต่ละจุดจะเชื่อมต่อกับ Flow line (ทิศทางการทำงาน)

รูปแบบของผังงาน

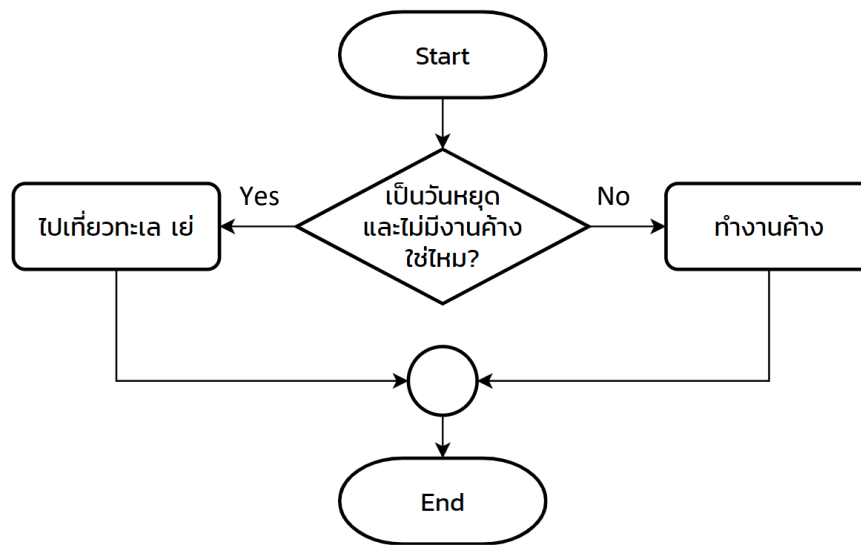
1. การทำงานตามลำดับ (Sequence)

ตัวอย่าง ฟังก์ชันซักล้างจาน



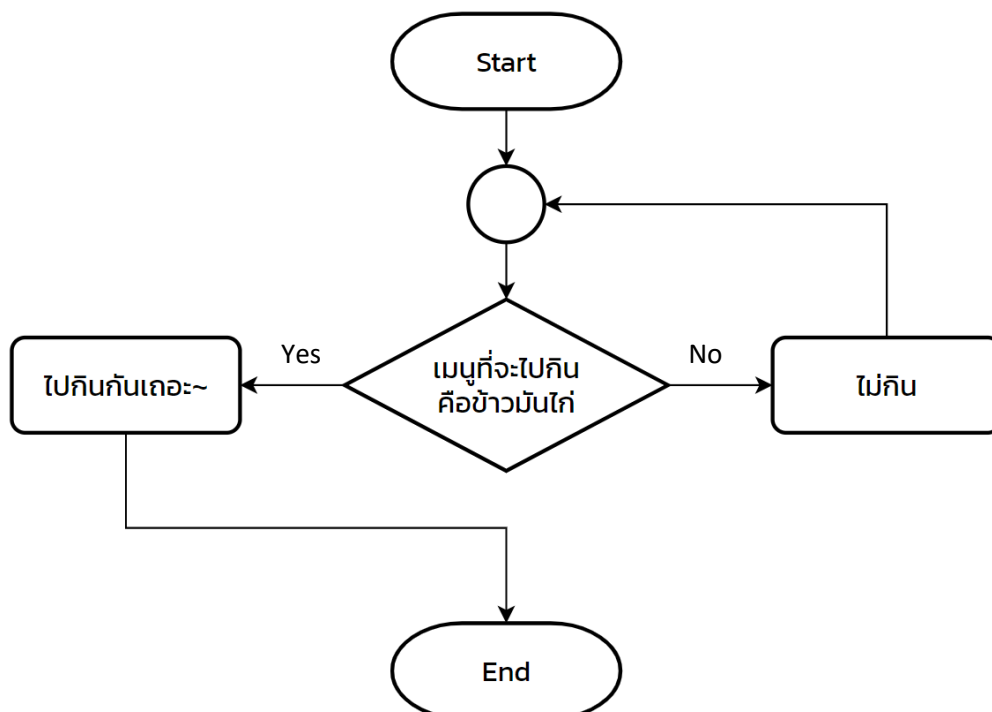
2. การเลือกทำตามเงื่อนไข (Decision)

ตัวอย่าง พี่เย็นอรุณอยากไปเที่ยวทะเล พี่เย็นอรุณจะได้ไปไหม



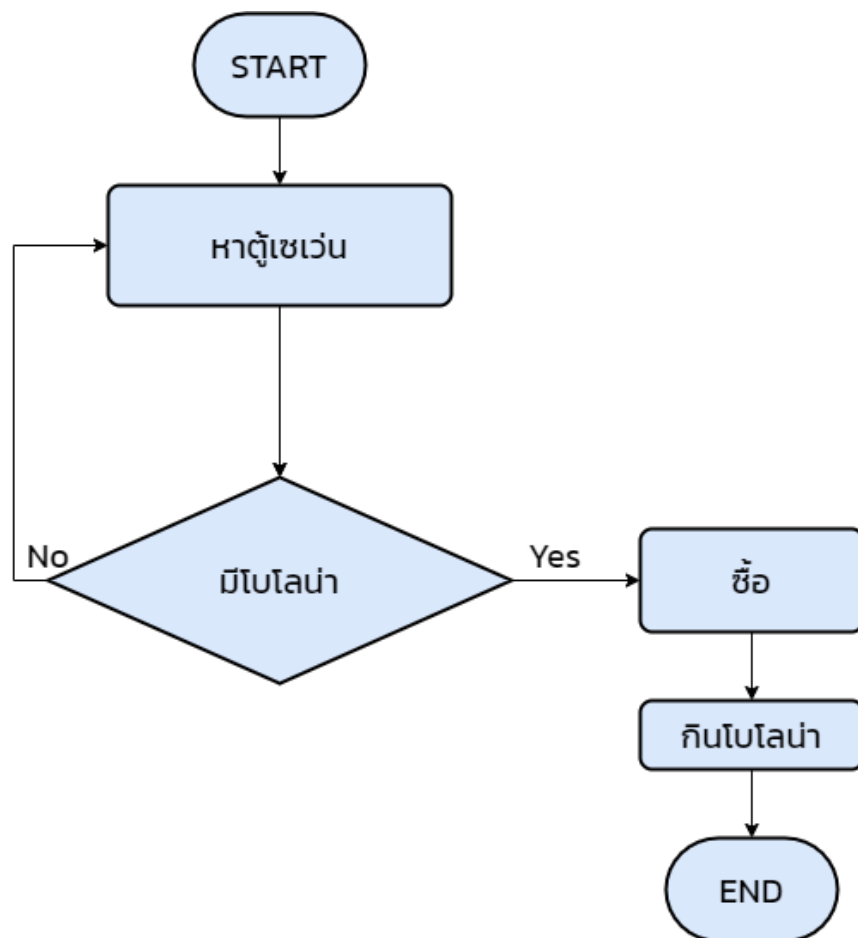
3. การทำซ้ำ (Looping or Repeating)

ตัวอย่าง เพื่อนชวนพี่นะโมไปหาอะไรกิน แต่พี่นะโมเป็นคนที่โปรดปรานข้าวมันไก่มาก ๆ และจะไม่ไปกิน ถ้าชวนไปกินเมนูอื่นที่ไม่ใช่ข้าวมันไก่



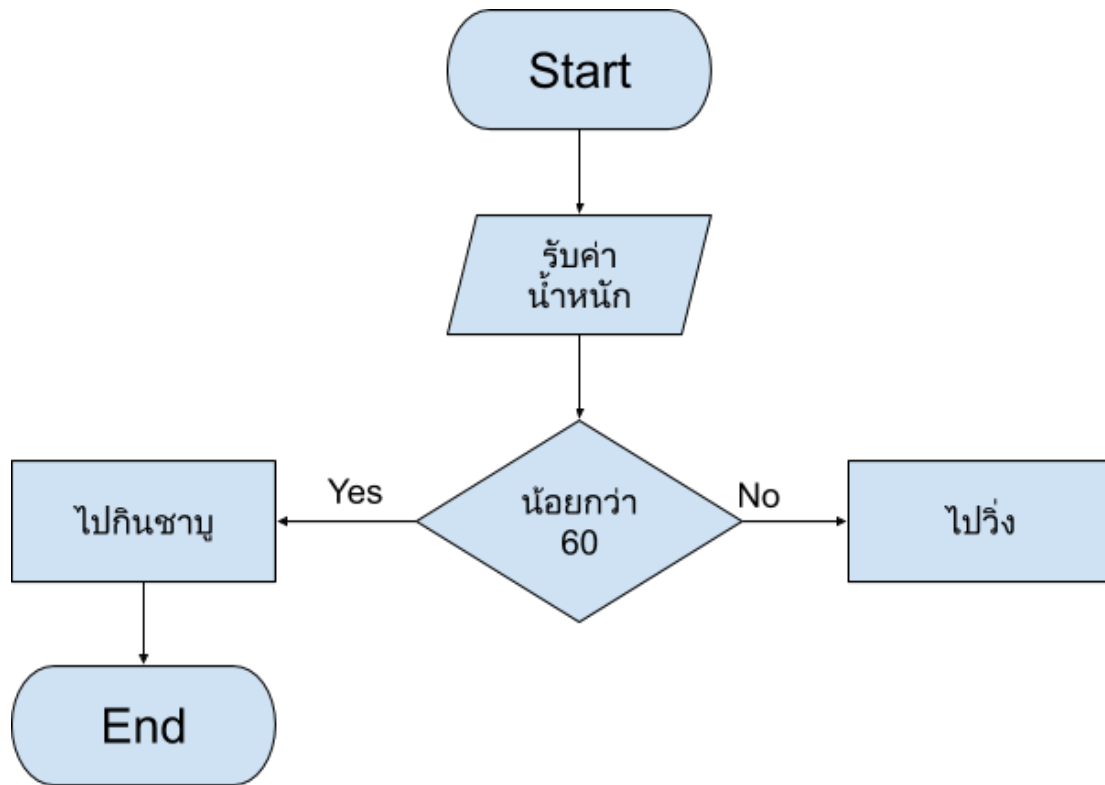
แบบฝึกหัด

1. พี่หนอยอยากกินโปโลน่าในตู้เซเว่น จงเขียน Flowchart ในการซื้อโปโลน่า



(ไม่ได้มีเพียงวิธีนี้วิธีเดียว)

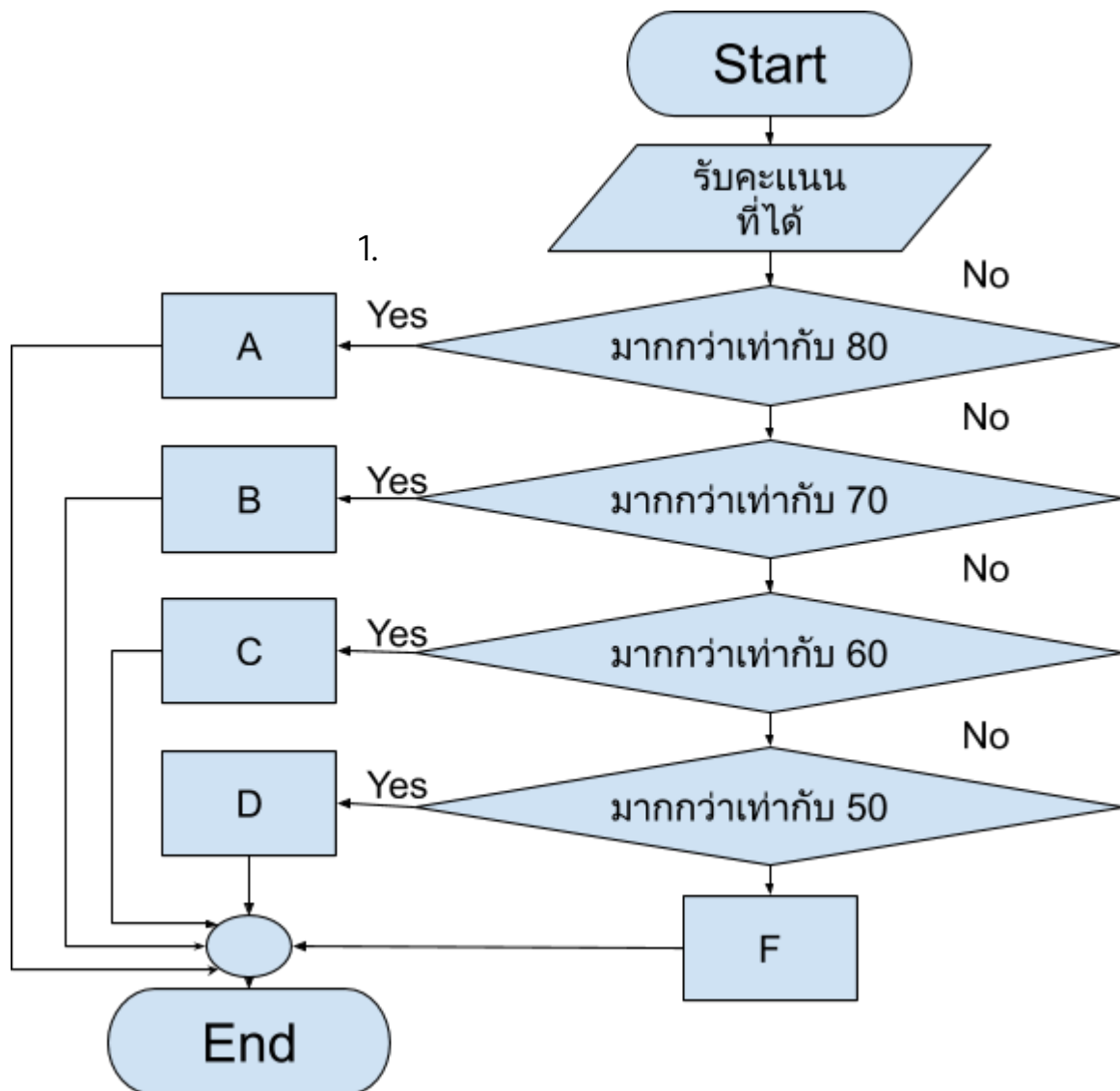
2. พี่ห.อยากไปกินชาบูกับเพื่อนแต่ก็กลัวอ้วน พี่ห.เลยตัดสินใจว่าจะไปวิ่งออกกำลังกายจนกว่าน้ำหนักจะน้อยกว่า 60 กก. แล้วถึงจะยอมไปกินชาบูกับเพื่อน



(ไม่ได้มีเพียงวิธีนี้วิธีเดียว)

3. อาจารย์ประกาศคะแนนพีมาแล้วแต่พีไม่รู้ว่พีได้เกรดอะไร น้อง ๆ ช่วยเขียน Flowchart ในการคำนวณเกรดจากคะแนนระหว่าง 0-100 คะแนนให้พวกพีสบายใจหน่อยนะ

- ถ้าคะแนนมากกว่าหรือเท่ากับ 80 ให้เกรดเท่ากับ A
- ถ้าคะแนนมากกว่าหรือเท่ากับ 70 ให้เกรดเท่ากับ B
- ถ้าคะแนนมากกว่าหรือเท่ากับ 60 ให้เกรดเท่ากับ C
- ถ้าคะแนนมากกว่าหรือเท่ากับ 50 ให้เกรดเท่ากับ D
- ถ้าคะแนนต่ำกว่า 50 ให้เกรดเท่ากับ F



(ไม่ได้มีเพียงวิธีนี้วิธีเดียว)

Pseudo code

Pseudo code คือคำสั่งเทียมที่ใช้ในการจำลองขั้นตอนและอธิบายการทำงานของอัลกอริทึม โดยเปลี่ยนจากการใช้สัญลักษณ์มาเป็นการใช้คำที่เข้าใจง่าย เช่น เขียนเป็นภาษาอังกฤษ มักเขียนให้สั้นและได้ใจความ

ประโยชน์ของ Pseudo code

1. นำมาใช้ทบทวนความถูกต้องกับสิ่งที่ได้ออกแบบไป
2. นำมาใช้เป็นแนวทางในการเขียนโปรแกรม
3. นำมาใช้เป็นเครื่องมือเพื่อการออกแบบเชิงโครงสร้างของโปรแกรม

หลักการเขียน Pseudo code

1. เขียนให้อยู่ในรูปภาษาง่าย ๆ ที่คนอ่านแล้วเข้าใจ นิยมใช้ภาษาอังกฤษ (แต่จะเขียนภาษาไทยก็ได้นะ)
2. เขียนหนึ่งคำสั่งต่อหนึ่งบรรทัด
3. ใช้ย่อหน้าให้เป็นประโยชน์ และแยกคำเฉพาะให้ชัดเจน
4. เขียนคำสั่งจากบนลงล่าง
5. ควรจัดรูปแบบโครงสร้างเป็นสัดส่วน เพื่อให้อ่านง่าย

คำสั่งพื้นฐานของคอมพิวเตอร์

1. การกำหนดค่า

<ชื่อตัวแปร> = <ข้อมูลที่จะเก็บ>

โดยชื่อตัวแปรทั้งหมดที่เราตั้งจะเป็นแบบ Case Sensitive เช่น ตัวแปร name กับ Name จะถือว่าเป็นคนละตัวกัน เนื่องจากตัวพิมพ์เล็กและพิมพ์ใหญ่ถือว่าเป็นคนละตัวกัน

ตัวอย่าง



```
camp_name = "ToBeIT"
year = 62
```

2. การคำนวณ

มีตัวดำเนินการดังนี้

+ : บวก เช่น $1+1$ จะได้ 2

- : ลบ เช่น $8-4$ จะได้ 4

* : คูณ เช่น $12*5$ จะได้ 60

/ :หาร เช่น $18/9$ จะได้ 2

% : modulo หรือ mod (การหารเอาเฉพาะเศษ) เช่น $21\%2$ จะได้ 1, $-23\%5$ จะได้ 2

// : floor division (การหารไม่เอาเศษ) เช่น $14//3$ จะได้ 4

** : ยกกำลัง เช่น $5**2$ จะได้ 25

ตัวอย่าง

$3 ** 3 = 27$ $26 \% 2 = 0$ $16 * 4 = 64$ $49 // 5 = 9$ $2 ** 4 = 16$

3. ตัวดำเนินการทางตรรกศาสตร์

ประกอบด้วย

== (เท่ากับ) : เป็นการเปรียบเทียบค่า ว่าค่าทั้งสองเท่ากันหรือไม่ เช่น $1 == 1$ จะคืนค่า **true**

&& (และ) : เป็นการตรวจสอบว่าค่าทางซ้ายและค่าทางขวาเป็นจริงทั้งคู่หรือไม่ ถ้าเป็นจริงทั้งคู่ จึงจะคืนค่า **true** แต่ถ้ามีค่าใดค่าหนึ่งเป็นเท็จ ก็จะไม่คืนค่า **false** กลับมา

เช่น $(1 == 1) \&\& (2 == 2)$ จะคืนค่า **true** ในขณะที่ $(1 == 2) \&\& (2 == 3)$ จะคืนค่า **false**

|| (หรือ) : เป็นการตรวจสอบค่าทั้งสองว่ามีค่าใดค่าหนึ่งเป็นจริงไหม ถ้ามีก็จะคืนค่า **true** แต่ถ้าทั้งสองค่าเป็นเท็จ ก็จะไม่คืนค่า **false**

เช่น $(1 == 1) || (2 == 3)$ จะคืนค่า **true** เนื่องจากเงื่อนไขแรกเป็นจริง

! (นิเสธ) : เป็นการกลับค่าความเป็นจริง เช่น **!true** จะคืนค่า **false** กลับมา ในขณะที่ **!false** จะคืนค่า **true** กลับมา

4. การรับข้อมูลเข้า

Read

คือการรับข้อมูลจากการอ่านไฟล์

Input / Get

คือการรับข้อมูลจากแป้นพิมพ์

ตัวอย่าง



Input <ชื่อตัวแปร>
Get text

5. การส่งข้อมูลออก

Write

คือการส่งข้อมูลออกไปบันทึกในไฟล์

Print, Put, Display, Output

คือการแสดงข้อมูลทางจอภาพ

ตัวอย่าง



```
Print <ชื่อตัวแปร/ข้อมูล>
Put "IT@KMITL"
```

6. การเปรียบเทียบเงื่อนไข

5.1 If <เงื่อนไข 1> Then

<คำสั่ง 1>

Elseif <เงื่อนไข 2> Then

<คำสั่ง 2>

Else

<คำสั่ง 3>

EndIf

If เริ่มต้นการทำงานแบบเงื่อนไข

Then ถ้า เงื่อนไข 1 เป็นจริงจะทำ คำสั่ง 1

Elseif ถ้า เงื่อนไข 1 เป็นเท็จและ เงื่อนไข 2 เป็นจริงจะทำ คำสั่ง 2

Else ถ้าเงื่อนไขทั้งหมดเป็นเท็จ จะทำคำสั่งที่อยู่ภายใต้ Else

ตัวอย่าง



```
Start
money = 120
If money > 200 Then
    Print "Shabu"
ElseIf money > 150 Then
    Print "Steak"
Else
    Print "Later :("
EndIf
End
```

5.2 Case <ตัวแปรที่ใช้เปรียบเทียบ> Of

<เงื่อนไข 1>:<คำสั่ง>

...

<เงื่อนไข n>:<คำสั่ง>

Default:<คำสั่ง>

EndCase

Case เริ่มต้นการทำงานแบบเงื่อนไข

<เงื่อนไข 1> ถ้าเงื่อนไขนี้เป็นจริงจะทำคำสั่งที่อยู่ภายใต้ <เงื่อนไข 1>

...

<เงื่อนไข n> ถ้าเงื่อนไขนี้เป็นจริงจะทำคำสั่งที่อยู่ภายใต้ <เงื่อนไข n>

<Default> ถ้าไม่มีเงื่อนไขใดเป็นจริงจะทำคำสั่งที่อยู่ภายใต้ <Default>

ตัวอย่าง

```

Start
Case money Of
    200:Print "Shabu"
    150:Print "Steak"
    50:Print "French Fries"
    Default:Print "Later :("
EndCase
End

```

7. การทำงานซ้ำ

6.1 For <ค่าเริ่มต้น> to <ค่าสิ้นสุด>

<คำสั่ง>

EndFor

วนซ้ำโดยที่มีการกำหนดจำนวนรอบที่แน่นอน โดยเริ่มจากค่าเริ่มต้นแล้วไล่ไปจนกระทั่งเลข
สุดท้าย

ตัวอย่าง

```

Start
For Count = 1 to 10
    Print Count
EndFor
End

```

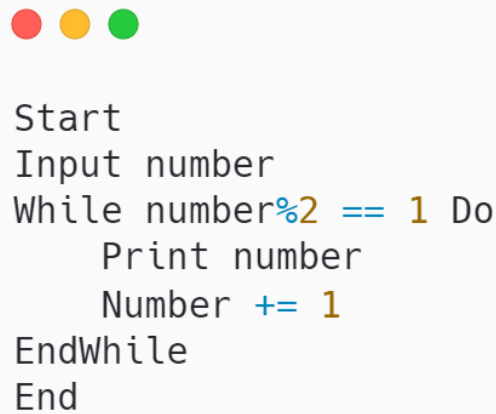
6.2 While <เงื่อนไข> Do

<คำสั่ง>

EndWhile

วนซ้ำโดยเช็คเงื่อนไข ถ้าเป็นจริงจะทำคำสั่งภายใต้ While และวนต่อไป จะกระทั่งเงื่อนไขเป็นเท็จจึงจะหยุดการวนซ้ำ

ตัวอย่าง



```
Start
Input number
While number%2 == 1 Do
    Print number
    Number += 1
EndWhile
End
```

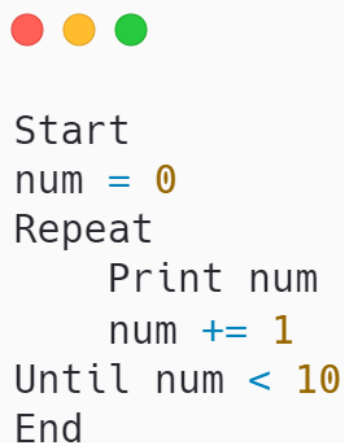
6.3 Repeat

<คำสั่ง>

Until <เงื่อนไข>

ทำงานคำสั่งภายใต้ Repeat ก่อนหนึ่งครั้ง จากนั้นจึงจะเช็คเงื่อนไข ถ้าเป็นเท็จจะทำคำสั่งและวนต่อไป จนกว่าเงื่อนไขจะเป็นจริง

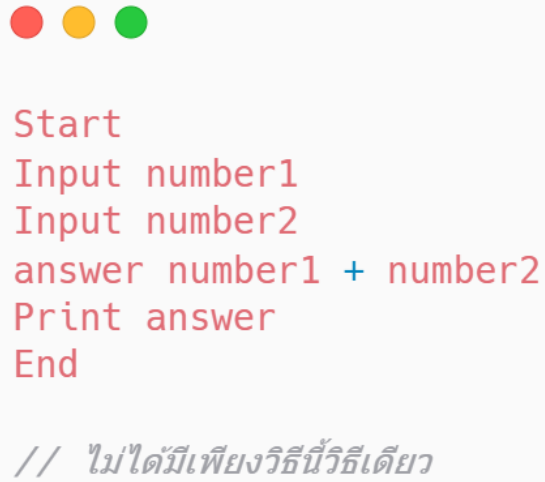
ตัวอย่าง



```
Start
num = 0
Repeat
    Print num
    num += 1
Until num < 10
End
```

แบบฝึกหัด

1. จงเขียน Pseudo code ของโปรแกรมบวกเลข



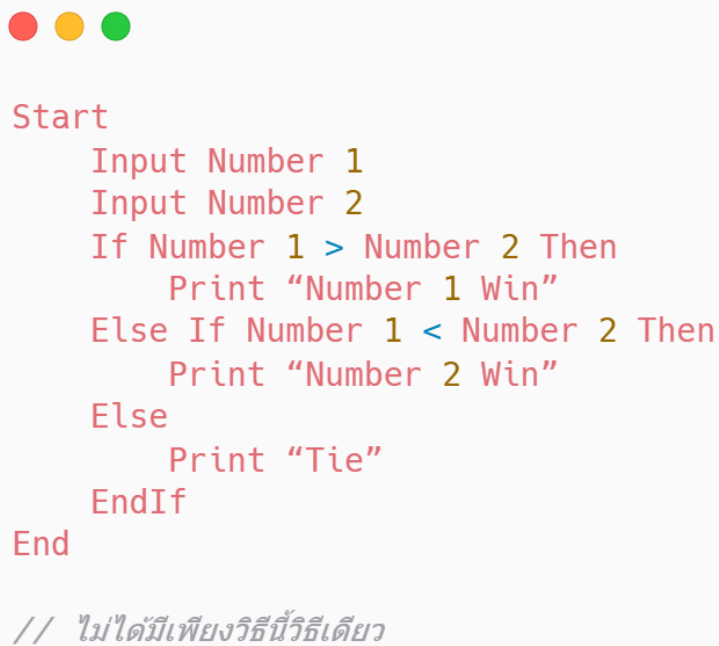
```

Start
Input number1
Input number2
answer number1 + number2
Print answer
End

// ไม่ได้มีเพียงวิธีนี้วิธีเดียว

```

2. จงเขียน Pseudo code ของโปรแกรมที่รับค่าสองค่า และเช็คค่าแรกมากกว่าค่าที่สองหรือไม่
 - ถ้ามากกว่าให้แสดงคำว่า "Number 1 Win"
 - ถ้าน้อยกว่าให้แสดงคำว่า "Number 2 Win"
 - ถ้าเท่ากันให้แสดงคำว่า "Tie"



```

Start
Input Number 1
Input Number 2
If Number 1 > Number 2 Then
    Print "Number 1 Win"
Else If Number 1 < Number 2 Then
    Print "Number 2 Win"
Else
    Print "Tie"
EndIf
End

// ไม่ได้มีเพียงวิธีนี้วิธีเดียว

```


3. จงเขียน Pseudo code ของโปรแกรมที่แสดงค่าตั้งแต่ 1 ถึง 10



Start

For count = 1 to 10

Print count

EndFor

End

// ไม่ได้มีเพียงวิธีนี้วิธีเดียว

4. จงเขียน Pseudo code ของโปรแกรมคำนวณเกรดโดยรับค่าคะแนนตั้งแต่ 0 - 100

- ถ้าคะแนนมากกว่าหรือเท่ากับ 80 ให้เกรดเท่ากับ A
- ถ้าคะแนนมากกว่าหรือเท่ากับ 70 ให้เกรดเท่ากับ B
- ถ้าคะแนนมากกว่าหรือเท่ากับ 60 ให้เกรดเท่ากับ C
- ถ้าคะแนนมากกว่าหรือเท่ากับ 50 ให้เกรดเท่ากับ D
- ถ้าคะแนนต่ำกว่า 50 ให้เกรดเท่ากับ F



Start

Input score

If score \geq 80 Then

Print "A"

Else if score \geq 70 Then

Print "B"

Else if score \geq 60 Then

Print "C"

Else if score \geq 50 Then

Print "D"

Else

Print "F"

EndIf

End

// ไม่ได้มีเพียงวิธีนี้วิธีเดียว

5. จงเขียน Pseudo code ของโปรแกรมในข้อ 4 แต่เป็นการรับค่าคะแนนของนักเรียน 5 คนเข้ามา และหาเกรดของแต่ละคน และแสดงผลในรูปแบบ "นักเรียน 1 : A"



Start

Input score

For count = 1 to 5

 If score >= 80 Then

 Print "นักเรียนที่" + count + " : A"

 Else if score >= 70 Then

 Print "นักเรียนที่" + count + " : B"

 Else if score >= 60 Then

 Print "นักเรียนที่" + count + " : C"

 Else if score >= 50 Then

 Print "นักเรียนที่" + count + " : D"

 Else

 Print "นักเรียนที่" + count + " : F"

 EndIf

EndFor

End

// ไม่ได้มีเพียงวิธีนี้วิธีเดียว

Computational Thinking Skills

Computational Thinking Skills หรือ การคิดเชิงคำนวณ คือ กระบวนการคิด หรือ วิธีการที่ใช้ในการแก้ไขปัญหาอย่างเป็นระบบตามหลักการเพื่อที่จะทำให้สามารถแก้ปัญหาได้อย่างมีประสิทธิภาพและเป็นรูปแบบ

4 เสาหลักแห่ง CTS ได้แก่

1. Decomposition (การย่อยปัญหา) คือการย่อยปัญหาหรือระบบที่ซับซ้อนออกเป็นส่วนเล็ก ๆ เพื่อให้ง่ายต่อการจัดการและแก้ปัญหา
2. Pattern Recognition (การจดจำรูปแบบ) คือการหารูปแบบหรือลักษณะที่คล้าย ๆ กันของปัญหาเล็ก ๆ ที่ถูกย่อยออกมา
3. Abstraction (ความคิดด้านนามธรรม) คือการมุ่งความคิดไปที่ข้อมูลสำคัญ และคัดกรองส่วนที่ไม่เกี่ยวข้องออกไป เพื่อให้จดจำเฉพาะสิ่งที่เราต้องการจะทำ
4. Algorithm Design (การออกแบบอัลกอริทึม) คือการพัฒนาแนวทางแก้ปัญหาอย่างเป็นขั้นเป็นตอน หรือสร้างหลักเกณฑ์ขึ้นมาเพื่อดำเนินตามทีละขั้นตอนในการแก้ไขปัญหา

สรุปคำจำกัดความของการคิดเชิงคำนวณ

- ไม่ได้จำกัดอยู่เพียงการคิดให้เหมือนคอมพิวเตอร์ หรือคิดในศาสตร์ของนักวิทยาศาสตร์คอมพิวเตอร์
- แต่เป็นกระบวนการคิดแก้ปัญหาของมนุษย์ เพื่อสั่งให้คอมพิวเตอร์ทำงานและช่วยแก้ปัญหาตามที่เราต้องการได้อย่างมีประสิทธิภาพ
- วิธีคิดเชิงคำนวณ ช่วยทำให้ปัญหาที่ซับซ้อนเข้าใจได้ง่ายขึ้น เป็นทักษะที่เป็นประโยชน์อย่างยิ่งต่อทุก ๆ สาขาวิชา และทุกเรื่องในชีวิตประจำวัน

ตัวอย่าง

ณ ประเทศ KMITL มีตู้วิเศษของพี่อาร์มเอามาจำหน่ายของ แล้วเจ้าตู้เนี่ยมันฉลาดมาก มันจะทอนเงิน โดยทอนจากแบงก์ 500, 100, 50, 20 และเหรียญ 10, 5, 2, 1 ตามลำดับ พี่อยากรู้ว่ามันทำงานยังไง เลยอยากให้น้องช่วยกัน คิด วิเคราะห์ แยกแยะความฉลาดของเจ้าเครื่องนี้โดยใช้ 4 เสาหลักกันเถอะ

Decomposition:

เริ่มจากย่อยปัญหาให้เป็นปัญหาเล็ก ๆ ก่อน โดยแบ่งเป็น

- สิ้นค้าราคาเท่าไร
- ได้รับเงินมาเท่าไร
- ต้องทอนเงินเท่าไร

Pattern Recognition:

พอเราแยกปัญหามาแล้วก็มาดูรูปแบบการทำงานกัน

- รูปแบบการทอนเงินก็คือมองจากแบงก์หรือเหรียญที่มากที่สุดก่อน

Abstraction:

แยกปัญหาที่ไม่เกี่ยวข้องออก เช่น

- ไม่สนใจว่าผู้ใช้จะใส่เงินมาเป็นแบงก์หรือเหรียญอะไรเอาแค่ยอดรวม
- ไม่สนใจราคาสินค้าชิ้นอื่น ๆ ที่ผู้ใช้ไม่ได้เลือก

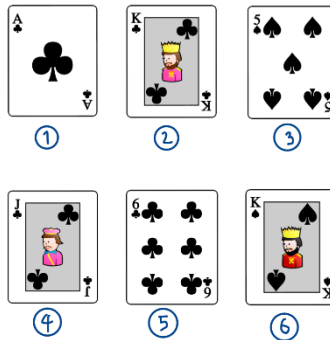
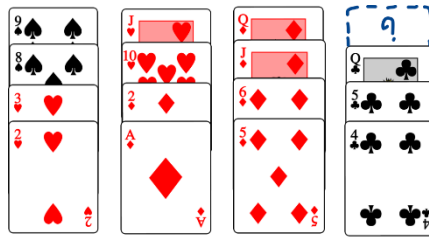
Algorithm Design:

แก้ไขปัญหทั้งหมดอย่างเป็นขั้นเป็นตอน

- รับเงิน
- ตรวจสอบว่าจำนวนเงินพอหรือไม่
- ตรวจสอบเงินทอนและทอนแบงก์หรือเหรียญที่มากที่สุดเสมอ

แบบฝึกหัด

1. พี่ปิ่นมีไพ่มาให้น้อง ๆ ลองสังเกตดูว่า ไพ่ใบที่หายไปใบนั้นคือไพ่ใบไหนได้บ้างใน 6 ใบข้างล่างนี้



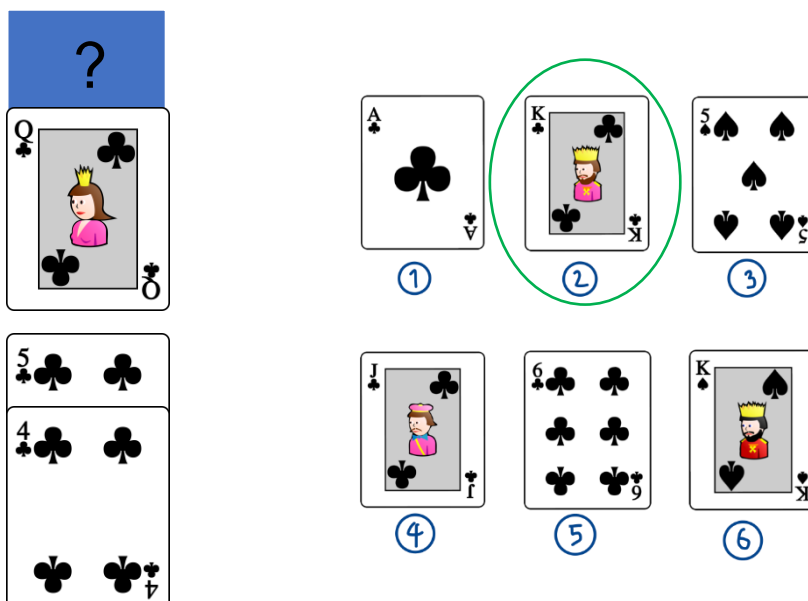
6 ใบนี้จะเป็นไพ่ใบไหนได้บ้างน่า

Pattern Recognition

- สังเกตว่าจะจับคู่ไพ่ 2 ใบ โดยไพ่คู่ที่จับจะต้องมีค่าต่างกันอยู่หนึ่งและมีสัญลักษณ์เดียวกัน

Abstraction

- ไม่สนใจว่าสัญลักษณ์จะเป็นสัญลักษณ์อะไร แค่เป็นสัญลักษณ์เดียวกันกับไพ่คู่ของมัน
- สนใจแค่ไพ่ใบที่หายไปควรมีค่าต่างกันอยู่ 1



2. กลุ่มตัวเลขกลุ่มหนึ่งมีตัวเลข 5 ตัว ได้แก่ 8, 4, 11, 9, 3

ข้อใดต่อไปนี้ ไม่ใช่ผลรวมหรือผลต่างระหว่างเลขสมาชิก 2 ตัวในกลุ่มตัวเลขนี้

A. 2

B. 7

C. 9

D. 12

E. 13

Decomposition

- แยกปัญหาโดยแบ่งเป็น ผลรวม กับ ผลต่าง

Algorithm Design

- แก้ปัญหาโดยทำงาน “วน” ไปเรื่อย ๆ จนกว่าจะได้คำตอบ (Brute Force Algorithm)

- ผลรวมทั้งหมดที่เป็นไปได้ :

$$8+4 = 12, \quad 8+11 = 19,$$

$$8+9 = 17, \quad 8+3 = 11,$$

$$4+11 = 15, \quad 4+9 = 13,$$

$$4+3 = 7, \quad 11+9 = 20,$$

$$11+3 = 14, \quad 9+3 = 12$$

- ผลต่างทั้งหมดที่เป็นไปได้ :

$$8-4 = 4, \quad 8-11 = -3,$$

$$8-9 = -1, \quad 8-3 = 5,$$

$$4-11 = -7, \quad 4-9 = -5,$$

$$4-3 = 1, \quad 11-9 = 2,$$

$$11-3 = 8, \quad 9-3 = 6$$

3. โรงเรียนพูนปี ไอที มีห้องเรียนชั้น ม.6 ทั้งหมด 4 ห้อง โดยเด็กแต่ละห้องจะมีความพิเศษคือ

เด็กห้อง 1 มีนิสัยพูดติดอ่าง เวลาที่พูดอะไร จะพูดคำนั้นออกมาสองครั้ง เช่น คำว่า "ซ้าย" เด็กห้องนี้จะพูดว่า "ซ้ายซ้าย"

เด็กห้อง 2 มีนิสัยขี้โกหก เวลาบอกอะไรกับใครจะพูดโกหกเสมอ

เด็กห้อง 3 เป็นเด็กนานาชาติ จะพูดคำต่าง ๆ ออกมาเป็นภาษาอังกฤษ

เด็กห้อง 4 เป็นเด็กขี้เกียจ จะไม่อ้าปากพูดอะไรแต่จะใช้วิธีการพิมพ์เลขและอักษรตัวหน้าแทน เช่นคำว่า "leftleftleftleftrightright" เด็กห้อง 4 จะบอกว่า "4L2R"

โดยเด็กแต่ละห้องจะคุยกันเรียงมาที่ละห้อง โดยเด็กห้อง 1 จะไปบอกเด็กห้อง 2 แล้วส่งต่อมารเรื่อย ๆ

ถ้าพี่ชินได้ส่งวิธีเดินมาห้องพัสดุให้พี่อ๊อฟโดยส่งผ่านแต่ละห้องมา แล้วพี่อ๊อฟได้ข้อมูลมาว่า "6L2R4L6R" (โดยตัว L ย่อมาจาก Left, R ย่อมาจาก Right)

พี่อ๊อฟจะต้องเดินไปอย่างไร จึงจะถึงห้องพัสดุ?

Decomposition

- แยกส่วนของปัญหาโดยแยกปัญหาของแต่ละห้องออกจากกัน

เช่น ห้อง 1 จะพูดติดอ่างทำให้ข้อมูลที่ถูกส่งมีการพูดซ้ำสองรอบ เป็นต้น

Abstraction

- เจาะจงข้อมูลที่สำคัญ โดยตัดทอนข้อมูลที่ไม่สำคัญออก เช่น ไม่สนใจว่าเด็กห้อง 1 พูดติดอ่าง จะสนใจแค่ว่า ข้อมูลที่ส่งต่อมาจากห้อง 1 มีการพูดซ้ำสองรอบ

Algorithm Design

- ข้อมูลถูกส่งมาจากห้อง 1 จนถึงห้อง 4 มาเป็นลำดับขั้นตอนจนได้ข้อมูลอย่างที่พี่อ๊อฟได้รับ

Decomposition

- ข้อมูลที่ส่งผ่านเด็กห้อง 1 จะเป็นลักษณะแบบไหน
- ข้อมูลที่ส่งผ่านเด็กห้อง 2 จะเป็นลักษณะแบบไหน
- ข้อมูลที่ส่งผ่านเด็กห้อง 3 จะเป็นลักษณะแบบไหน
- ข้อมูลที่ส่งผ่านเด็กห้อง 4 จะเป็นลักษณะแบบไหน

Abstraction

- ไม่สนใจว่าเด็กห้องไหนนิสัยเป็นแบบไหน
- สนใจที่ลักษณะข้อมูลที่ผ่านเด็กห้องต่างๆมา
- สนใจลำดับการส่งของข้อมูล

Algorithm Design

- ทำการย้อนลักษณะของข้อมูล
- ทำการแก้ปัญหที่แยกส่วนไว้แล้วเป็นลำดับไป
- ขั้นนี้จะต้องทำการแก้ปัญหโดยการมองย้อนกลับ

โดย "6L2R4L6R" เป็นข้อมูลล่าสุดที่เรามีอยู่ ก็ต้องมองย้อนกลับไปผ่านห้อง 4 ไปห้อง 3 ไปห้อง 2 และไปห้อง 1 นั้นเอง

Room	ข้อมูลที่ได้รับจากแต่ละห้อง
Room 4	6L2R4L6R
Room 3	Left*6 Right*2 Left*4 Right*6
Room 2	ขวา*6 ซ้าย*2 ขวา*4 ซ้าย*6
Room 1	ขวา*3 ซ้าย*1 ขวา*2 ซ้าย*3

คำตอบก็คือ : **ขวาขวาขวา ซ้าย ขวาขวา ซ้ายซ้ายซ้าย**

4. แยกแล้วพี่อาร์มมาเที่ยวเยอรมันแล้วกลับไปสอนน้อง ๆ ไม่ทัน พี่จะต้องเดินทางกลับไปสอนน้อง ๆ ให้ทัน โดยมีวิธีการเดินทางมีหลายวิธี คือ

1. นั่งเครื่องบินกลับ ใช้เวลา 4 ชั่วโมง ไปที่สถานี A แล้วต่อด้วยนั่งแท็กซี่อีก 3 ชั่วโมงไปถึงสถานี E แล้วเดินเท้าอีก 3.5 ชั่วโมง แต่พี่สามารถนั่งสองแถวจากสถานี A ไปยังคณะได้เลยใช้เวลา 7 ชั่วโมง หรือ พี่จะไปสถานี D โดยการนั่งวินไปใช้เวลา 3 ชั่วโมง และเดินเท้าต่ออีก 3 ชั่วโมง
2. นั่งเรือ ใช้เวลา 3.5 ชั่วโมง มาที่สถานี B แล้วนั่งเครื่องบินต่ออีก 3.5 ชั่วโมงเพื่อไปสถานี D และนั่งสองแถวมาคณะอีก 3 ชั่วโมง
3. นั่งเรือดำน้ำ ใช้เวลา 5.5 ชั่วโมงมาที่สถานี C แล้วนั่ง Speed Boat อีก 1 ชั่วโมงไปยัง สถานี D แล้วขี่ม้าไปยังคณะอีก 3 ชั่วโมง หรือพี่จะขี่ R15 จากสถานี C ไปยังคณะโดยใช้เวลา 6.5 ชั่วโมง

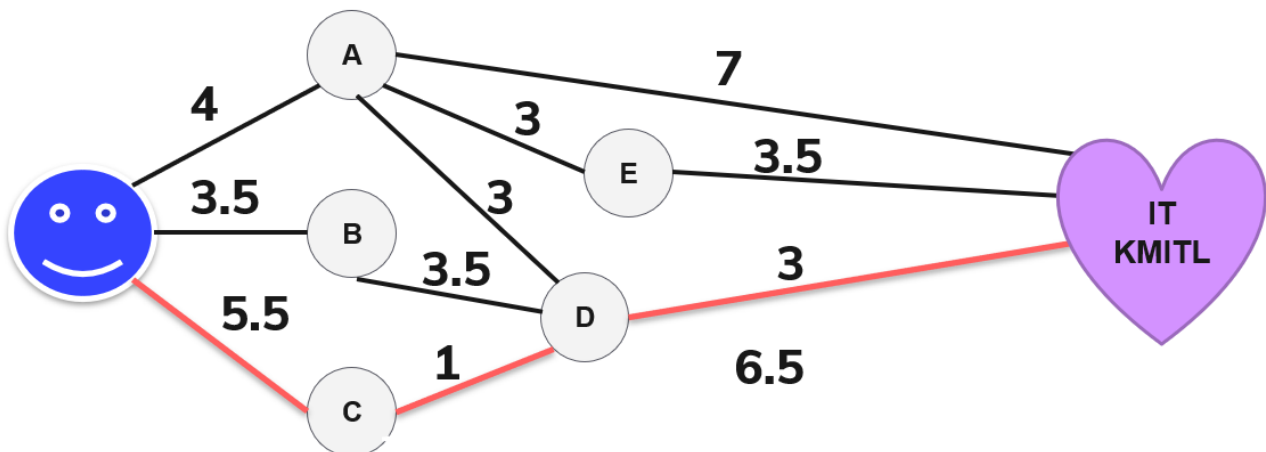
วานน้อง ๆ ช่วยพี่อาร์มคิดหน่อยนะ ว่าใช้เส้นทางไหนจึงจะทำให้มาถึงคณะเร็วที่สุด (ปล.จินตนาการสำคัญกว่าความรู้)

Decomposition

- แยกปัญหา ว่าการเดินทางแต่ละวิธีใช้เวลาเท่าไร

Abstraction

- ไม่สนใจว่าเดินทางด้วยวิธีการอะไร
- สนใจเพียงแค่ว่าเวลาที่ใช้ในการเดินทาง

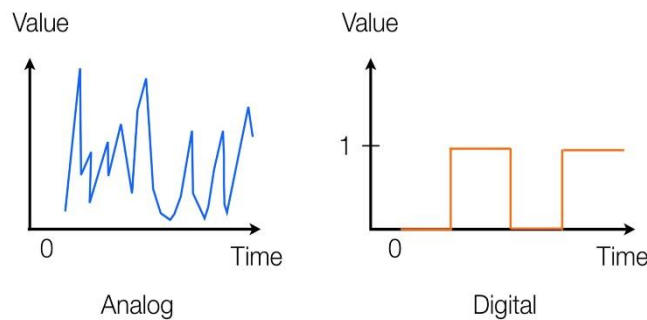


Logic

System

System หรือ ระบบ คือสิ่งที่กำหนดความสัมพันธ์ของ Input และ Output ซึ่งเมื่อป้อน Input ไป ระบบ จะทำการประมวลผลและส่งเป็น Output ออกมา

Analog vs Digital Waveforms



Analog System

ระบบ Analog มีลักษณะสำคัญคือ ค่าของ Input และ Output เป็นแบบ “ต่อเนื่อง” และมีจำนวนระดับค่าที่ “ไม่จำกัด”

Digital System

ระบบ Digital มีลักษณะสำคัญคือ ค่าของ Input และ Output เป็นแบบ “ไม่ต่อเนื่อง” และมีจำนวนระดับค่าที่ “จำกัด”

Digital Circuits (วงจรรดิจิตัล)

วงจรรดิจิตัล คือ ระบบการเชื่อมต่อกันของอุปกรณ์ดิจิตัล โดยส่วนประกอบในวงจรรดิจิตัล ทำงานโดยใช้ระบบเลขฐานสอง (Binary numbers) Input และ Output ของอุปกรณ์ดิจิตัล มีเพียง 2 สถานะ คือ 1 หรือ 0 (High หรือ Low)

Statement (ประพจน์)

ประพจน์ คือ ประโยคหรือข้อความที่มีค่าความจริงเป็น จริง หรือ เท็จ อย่างใดอย่างหนึ่ง

Truth Table (ตารางค่าความจริง)

คือ ตารางที่แสดงความสัมพันธ์ระหว่าง Output กับ Input ที่เป็นไปได้ทุกรูปแบบ

Logic Gate

Logic gate คือ ตัวดำเนินการทางตรรกศาสตร์ โดยจะรับ Input อย่างน้อย 1 ตัว มาประมวลผลและส่งออกมาเป็น Output ออกไป 1 ตัว

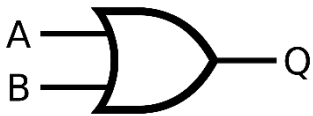
ถ้าเรานำ Logic gate มาประกอบกัน จะได้วงจรที่เรียกว่า **Logic circuit** ซึ่งเป็นวงจร Digital รูปแบบหนึ่ง

AND: Output เป็น 1 เมื่อ Input ทุกตัวเป็น 1 หรือเป็น 0 เมื่อมี Input ตัวใดตัวหนึ่งเป็น 0



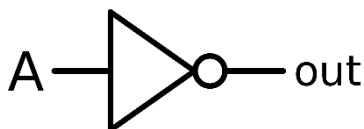
A	B	$A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

OR: Output เป็น 0 เมื่อ Input ทุกตัวเป็น 0 หรือเป็น 1 เมื่อมี Input ตัวใดตัวหนึ่งเป็น 1



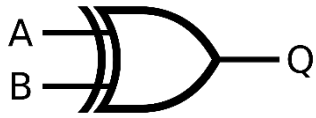
A	B	$A + B$
0	0	0
0	1	1
1	0	1
1	1	1

NOT: Output เป็น 0 เมื่อ Input เป็น 1 หรือ Output เป็น 1 เมื่อ Input เป็น 0



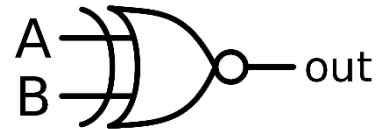
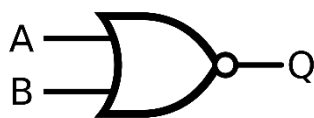
A	$\sim A$
0	1
1	0

XOR (Exclusive OR): ในกรณีที่รับ Input 2 ตัว, Output จะเป็น 1 เมื่อรับ Input ต่างกัน หรือ Output จะเป็น 0 เมื่อรับ Input เหมือนกัน



A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

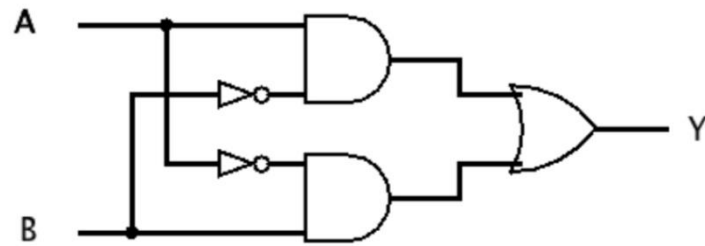
NAND / NOR / XNOR: คือ Gate AND / OR / XOR ที่ Output ต่อกับ Gate NOT



A	B	$\sim (A \cdot B)$	$\sim (A + B)$	$\sim (A \oplus B)$
0	0	1	1	1
0	1	1	0	0
1	0	1	0	0
1	1	0	0	1

แบบฝึกหัด

1. จงเขียนตารางค่าความจริงจากวงจรนี้



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

2. จงเขียนตารางค่าความจริงจากสมการพีชคณิตบูลีน $Y = (A \cdot B) + C$

A	B	C	$A \cdot B$	Y
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	0	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1

Number Bases

การแปลงเลขฐานใด ๆ เป็นเลขฐาน 10

ฐาน 2 เป็นฐาน 10

① แปลงเลขฐาน 2 เป็นเลขฐาน 10

ex. 1101_2

$\begin{array}{cccc} 1 & 1 & 0 & 1 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ \text{ตำแหน่งที่} & 3 & 2 & 1 & 0 \end{array}$

Sol

$$\begin{aligned}
 & (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) \\
 & \quad 8 + 4 + 0 + 1 \\
 & \quad = 13 \\
 & \therefore 1101_2 \rightarrow 13_{10}
 \end{aligned}$$

ฐาน 8 เป็นฐาน 10

② แปลงเลขฐาน 8 เป็นเลขฐาน 10

ex. 3743_8

$\begin{array}{cccc} 3 & 7 & 4 & 3 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ \text{ตำแหน่งที่} & 3 & 2 & 1 & 0 \end{array}$

Sol

$$\begin{aligned}
 & (3 \times 8^3) + (7 \times 8^2) + (4 \times 8^1) + (3 \times 8^0) \\
 & \quad 1536 + 448 + 32 + 3 \\
 & \quad = 2019 \\
 & \therefore 3743_8 \rightarrow 2019_{10}
 \end{aligned}$$

ฐาน 16 เป็นฐาน 10

③ แปลงเลขฐาน 16 เป็นเลขฐาน 10

ex. $3F4_{16}$

	3	F	4
	↓	↓	↓
ตำแหน่งที่	2	1	0

Sol

$$(3 \times 16^2) + (15 \times 16^1) + (4 \times 16^0)$$

$$768 + 240 + 4$$

$$= 1012_{10}$$

$$\therefore 3F4_{16} \rightarrow 1012_{10}$$

ฐาน 10 เป็นฐาน 2

① แปลงเลขฐาน 10 เป็นเลขฐาน 2

ex. 63_{10}

2	63		
2	31	เศษ	1
2	15	เศษ	1
2	7	เศษ	1
2	3	เศษ	1
	1	เศษ	1

$$63_{10} \rightarrow 111111_2$$

แบบฝึกหัด

1. จงแปลงเลข 18 ในระบบเลขฐานสิบเป็นเลขฐานสอง

ANS : key – 10010_2

2. จงแปลงเลข 1011_2 เป็นเลขฐานสิบ

ANS : key – 11_{10}

3. จงแปลงค่า $18A_{16}$ เป็นเลขฐานแปด

ANS : key - 612_8

4. จงแปลงค่า $3FA_{16}$ เป็นเลขฐานสอง

ANS : key - 11111010_2

5. $2E_{16}$ บวก 72_8 เท่ากับเท่าใดในระบบเลขฐานสิบ

ANS: key - $46_{10} + 58_{10} = 104_{10}$

Basic Programming with Python

ทำไมถึงต้อง Python?

มาดูการแสดงผลของคำว่า 'Hello, World' ของแต่ละภาษากัน

C

```
#include <stdio.h>

int main()
{
    printf("Hello, world!");
    return 0;
}
```

Java

```
public class Work_name_or_tag
{
    public static void main (String[] args)
    {
        System.out.println("Hello, world!");
    }
}
```

Python

```
print("Hello, world")
```

จะเห็นได้ว่าโครงสร้างไวยากรณ์นั้นเข้าใจง่ายเมื่อเทียบกับภาษาอื่น ๆ โดย Python ได้ตัดสัญลักษณ์และคำสั่งที่ไม่จำเป็นออกไปและใช้การกำหนดย่อหน้า (Indent) เพื่อระบุขอบเขตของบล็อกคำสั่งหรือคำสั่งย่อยแทนการใช้เครื่องหมาย { } จึงทำให้โค้ดมีความกระชับและมีบรรทัดที่น้อยลง เหมาะสำหรับผู้เริ่มต้นศึกษา

การทำงานของโปรแกรม (Running)

Compiler คือการที่โปรแกรมทำงานด้วยการ **สแกนตรวจสอบโค้ดทั้งโปรแกรม** ก่อนจึงจะทำงาน ถ้าหากพบข้อผิดพลาดระหว่างการสแกน ก็จะไม่สามารถทำงานได้เลย

Interpreter คือการที่โปรแกรม **ทำงานทีละบรรทัด** โดยโปรแกรมจะทำงานไปเรื่อย ๆ และจะหยุดทำงานก็ต่อเมื่อถึงบรรทัดที่มีข้อผิดพลาด

รู้จักกับ Datatype

1. จำนวนเต็ม (Integer)

เป็นประเภทข้อมูลที่ใช้สำหรับเก็บข้อมูลเลขจำนวนเต็ม เช่น 2, -48, 0, 16, 1024

2. จำนวนจริง (Float)

เป็นประเภทข้อมูลที่ใช้สำหรับเก็บข้อมูลเลขจำนวนจริง เช่น 3.14, 1.0

3. ตัวอักษร (String)

เป็นประเภทข้อมูลที่ใช้สำหรับเก็บตัวอักษร หรือ ประโยค เช่น "a", 'abc', "ToBeIT" โดยจะสังเกตได้ว่า ข้อมูลประเภท string จะมีเครื่องหมาย double quote (" ") หรือ single quote (' ') ครอบอยู่

4. ค่าความจริง (Boolean)

เป็นประเภทข้อมูล ที่ใช้สำหรับเก็บค่าความจริง โดยมี 2 ค่า คือ 1 และ 0

Arithmetic Operator (ตัวดำเนินการทางคณิตศาสตร์)

Name	Operator
Addition / บวก	+
Subtraction / ลบ	-
Multiplication / คูณ	*
Division / หาร	/
Floor Division / หารไม่เอาเศษ	//
Exponentiation / ยกกำลัง	**
Modulus / หารเอาเศษ	%

หลักการทางคณิตศาสตร์ PEMDAS

อธิบายลำดับการทำงานของตัวดำเนินการทางคณิตศาสตร์ โดยยึดตามหลัก **PEMDAS**

Parentheses () => Exponents ** => Multiply or Divide *, /, %, // => Add or Subtract +, -

(ถ้าทำตามหลัก PEMDAS แล้วเจอตัวดำเนินการที่มีลำดับความสำคัญเท่ากันให้ทำจากซ้ายไปขวา)

ตัวอย่าง

1. $2 - 5 + 0$

ANS : -3

2. $2 - 3 ** 2 // 2 * 5 + 1 * 0$

ANS : -18

การสร้างตัวแปร (Assign Variable name)

ตัวแปร คือ สิ่งที่เราสร้างขึ้นมานำมาเก็บค่าที่ต้องการนำไปใช้ในการประมวลผลในโปรแกรม

หลักการตั้งชื่อตัวแปร

ชื่อตัวแปร สามารถประกอบไปด้วย

- ตัวอักษรภาษาอังกฤษ (a - z, A - Z)
- ตัวเลข (0 - 9)
- เครื่องหมาย underscore (_)

*** ชื่อของตัวแปร จะต้องไม่ขึ้นต้นด้วยตัวเลข**

*** ชื่อของตัวแปร จะต้องไม่ซ้ำกับ keyword**

*** ชื่อของตัวแปร ควรมีความหมายที่ชัดเจน สื่อความหมายได้ว่าตัวแปรนี้หมายความว่าอะไร**

การรับและแสดงข้อมูลพื้นฐาน Basic Input / Output

input() : เป็นคำสั่งรับข้อมูลเข้าจากผู้ใช้

print() : เป็นคำสั่งแสดงผลข้อมูลออกทางหน้าจอ

การแปลงชนิดของข้อมูล Data type Conversion

การแปลงประเภทข้อมูลจากประเภทหนึ่งไปสู่ประเภทหนึ่ง โดยมีรูปแบบการใช้งานดังนี้

`new_datatype(data)`

ตัวอย่าง

- `int(data)` คือการแปลงข้อมูลที่อยู่ในวงเล็บ (data) ให้เป็นข้อมูลประเภท **integer**
- `float(data)` คือการแปลงข้อมูลที่อยู่ในวงเล็บ (data) ให้เป็นข้อมูลประเภท **float**
- `str(data)` คือการแปลงข้อมูลที่อยู่ในวงเล็บ (data) ให้เป็นข้อมูลประเภท **string**

*** เราไม่สามารถแปลง String ที่ไม่ใช่ตัวเลข เช่น "P'Arm Narak" ให้เป็นข้อมูลประเภท int หรือ float ได้ เพราะตัวอักษรไม่สามารถแปลงเป็นตัวเลขได้**

Output Format

การจัดรูปแบบของ Output ตามชนิดข้อมูลต่าง ๆ

- รูปแบบ string ใช้ `%s` ในการแสดงผล
- รูปแบบ int ใช้ `%d` ในการแสดงผล
- รูปแบบ float ใช้ `%f` ในการแสดงผล

```
a = "Hello"
b = 11
c = 3.14
print("%s %d %f" % (a, b, c))
//Output: Hello 11 3.14
```

การกำหนดเงื่อนไข Condition

โดยปกติแล้วการทำงานของโปรแกรมนั้นมีการตรวจสอบเงื่อนไขอยู่หลายครั้ง อย่างที่น้อง ๆ เคยได้รู้จัก ถ้า...แล้ว... หรือ if...else ซึ่งหลักการทำงานของ if else condition นั้นจะมีหลักการดังนี้

โดยโปรแกรมจะตรวจสอบว่าเงื่อนไขเป็นจริงไหม หากเป็นจริงจะทำสิ่งที่อยู่ภายใต้เงื่อนไข หากไม่ก็จะทำอีกอย่างหนึ่งแทน

```
if Boolean-expression:
    if_block_statement
else:
    else_block_statement
```

Information Technology Fundamental

คอมพิวเตอร์คืออะไร

อุปกรณ์อิเล็กทรอนิกส์ที่ทำงานภายใต้ชุดคำสั่งที่ถูกเก็บเอาไว้ในหน่วยความจำ

กระบวนการทำงานของคอมพิวเตอร์



1. การนำข้อมูลเข้า (Input Data)

คือการนำเข้าข้อมูลซึ่งอาจจะอยู่ในรูปแบบของ ตัวเลข, ตัวอักษร, เสียง, รูปภาพ เข้าไปสู่คอมพิวเตอร์

2. การประมวลผล (Processing)

คือการนำข้อมูลที่ได้รับเข้ามานำมาจัดการ และ ประมวลผลด้วยกระบวนการต่าง ๆ ในคอมพิวเตอร์ไม่ว่าจะเป็น การจัดการทางคณิตศาสตร์, การจัดการทางตรรกศาสตร์, การจัดกลุ่ม

3. การนำเสนอข้อมูล (Output Data)

คือการนำข้อมูลที่ได้ผ่านการประมวลผลแล้วออกมาอยู่ในระบบสารสนเทศ เช่น สถิติหรือเอกสารต่าง ๆ

ข้อมูลกับสารสนเทศแตกต่างกันอย่างไร?

- ข้อมูล คือสิ่งที่ยังไม่ผ่านการประมวลผลเช่น ข้อมูลจากการสำรวจ แบบสอบถาม, คะแนนสอบ
- สารสนเทศ คือข้อมูลที่ได้ผ่านการประมวลผลแล้วด้วยกระบวนการต่าง ๆ เช่น ผลการประเมินจากแบบสำรวจ, ผลการเรียนของนักเรียน, ผลการเลือกตั้ง

ประเภทของคอมพิวเตอร์

1. **Super Computer (ซูเปอร์คอมพิวเตอร์)** เป็นคอมพิวเตอร์ที่มีขนาดใหญ่และมีขีดความสามารถสูงที่สุดในเครื่องประกอบไปด้วยหน่วยประมวลผลหลายหมื่นตัว มีหน้าที่ในการประมวลผลด้านในด้านหนึ่งเป็นพิเศษ เช่น การพยากรณ์อากาศ การคำนวณทางด้านดาราศาสตร์
2. **Mainframe Computer (เมนเฟรมคอมพิวเตอร์)** เป็นคอมพิวเตอร์ที่มีขีดความสามารถสูงรองลงมาเหมาะแก่การเป็นเครื่องแม่ข่ายที่ให้บริการต่าง ๆ กับเครื่องลูกข่าย ไม่ว่าจะเป็นการประมวลผล, หน่วยความจำ, หน่วยเก็บข้อมูล
3. **Server Computer (เซิร์ฟเวอร์คอมพิวเตอร์)** เป็นคอมพิวเตอร์ที่ให้บริการแก่เครื่องในเครือข่ายให้สามารถใช้บริการต่าง ๆ บนเครื่องได้เช่น การใช้บริการอีเมล, การรับฝากไฟล์
4. **Embedded Computer (คอมพิวเตอร์ฝังตัว)** เป็นคอมพิวเตอร์ขนาดเล็กที่ถูกออกแบบเป็นพิเศษให้ทำงานด้านใดด้านหนึ่งโดยเฉพาะ โดยส่วนมากจะเอาไว้ควบคุมอุปกรณ์ไฟฟ้า เช่น ไมโครเวฟ, ตู้เย็น, เครื่องซักผ้า
5. **Desktop Computer (คอมพิวเตอร์แบบตั้งโต๊ะ)** เป็นคอมพิวเตอร์ที่เอาไว้ใช้งานส่วนบุคคล สามารถพบได้บ่อยในชีวิตประจำวัน เอาไว้ใช้งานในเรื่องทั่ว ๆ ไป
6. **Portable Computer (คอมพิวเตอร์แบบพกพา)** เป็นคอมพิวเตอร์ส่วนบุคคลขนาดเล็กที่มีน้ำหนักเบา สะดวกกับการเคลื่อนย้ายไปยังที่ต่าง ๆ สามารถใช้พลังงานไฟฟ้าทั่วไปหรือใช้พลังงานจากแบตเตอรี่ก็ได้ และประสิทธิภาพไม่แพ้คอมพิวเตอร์แบบตั้งโต๊ะ
7. **Smartphone** เป็นโทรศัพท์ที่รวมเอาความสามารถแบบคอมพิวเตอร์ไว้ในเครื่องเดียวกัน สามารถเข้าถึงอินเทอร์เน็ตได้ และผู้ใช้สามารถติดตั้งโปรแกรมเสริมเพื่อเพิ่มความสามารถของโทรศัพท์ได้
8. **Cloud Computing** เป็นการใช้คอมพิวเตอร์, ระบบ และทรัพยากรของเครื่องคอมพิวเตอร์ของผู้ให้บริการ โดยสามารถเลือกกำลังการประมวลผล เลือกจำนวนทรัพยากรได้อิสระตามความต้องการในการใช้งาน

Hardware

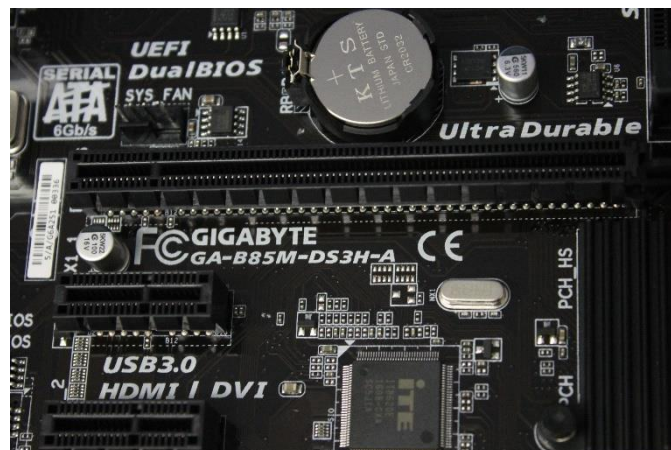
Hardware คือเครื่องมือ เครื่องจักร ชิ้นส่วน และอุปกรณ์ต่าง ๆ ที่สามารถมองเห็น และจับต้องได้ในระบบคอมพิวเตอร์ หมายถึงชิ้นส่วนหรืออุปกรณ์ต่าง ๆ ที่ประกอบกันขึ้นเป็นเครื่องคอมพิวเตอร์ รวมถึงอุปกรณ์ต่อพ่วงต่าง ๆ

ประกอบไปด้วย

- Input Device อุปกรณ์นำข้อมูลเข้าคอมพิวเตอร์
- Output Device อุปกรณ์แสดงผลข้อมูล
- System unit อุปกรณ์ต่าง ๆ ภายในเคส
- Storage Device อุปกรณ์จัดเก็บข้อมูล
- Communication Device อุปกรณ์ที่ทำให้คอมพิวเตอร์สามารถรับส่งข้อมูลกับเครื่องอื่น ๆ ได้

Hardware แบ่งออกเป็นสองประเภทใหญ่ ๆ ได้แก่

1. Peripheral คือ อุปกรณ์ต่อพ่วง เช่น เมาส์ คีย์บอร์ด ฯลฯ
2. System unit คือ อุปกรณ์ต่าง ๆ ภายในเคส เช่น CPU , Harddisk ฯลฯ



System unit

เมนบอร์ด (Mainboard)

Mainboard หรือ Motherboard คือแผงวงจรหลักที่เป็นศูนย์อุปกรณ์ต่าง ๆ ในคอมพิวเตอร์ เช่น CPU, RAM, Harddrive เป็นโครงสร้างที่อุปกรณ์ทุกอย่างต้องเชื่อมต่อผ่านที่นี่ เปรียบได้เป็นทางเดินของข้อมูลเพื่อให้แต่ละชิ้นส่วนสามารถติดต่อสื่อสารกันได้ นอกจากนี้ยังมีพอร์ต (Port) การเชื่อมต่อต่าง ๆ ไว้สำหรับอุปกรณ์ต่อพ่วง โดยเมนบอร์ด มีหลายขนาดให้เลือกใช้ เช่น ATX, mATX, Mini-ITX เป็นต้น

ซีพียู (CPU: Central Processing Unit)

คือหน่วยประมวลผลของคอมพิวเตอร์ ทำหน้าที่ควบคุม คำนวณตัดสินใจ และ ประมวลผลของชุดคำสั่ง โดยจะถูกติดตั้งไว้ในเมนบอร์ดในตำแหน่งที่เรียกว่า Socket โดยใน CPU จะมีแกนสมองหรือคอร์ (Core) ไว้หลายๆแกนบรรจุเอาไว้อยู่ เพื่อที่จะช่วยกันประมวลผล นอกจากนั้นแล้วยังมี Thread ที่ไว้เป็นการจำลองการประมวลผลในคอร์ของ CPU ให้ประมวลผลได้มากกว่าเดิม

CPU จะประกอบไปด้วย 3 ส่วน คือ

1. Control Unit

ควบคุมการทำงาน ประสานการทำงาน และการประสานงานต่าง ๆ ใน CPU

2. Arithmetic Logic Unit (ALU)

ทำหน้าที่คิดคำนวณทางคณิตศาสตร์ และตรรกศาสตร์

3. Memory Unit

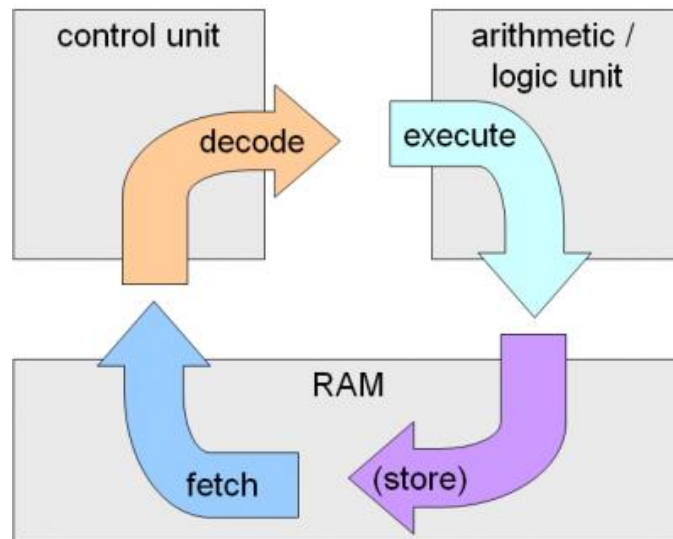
เป็น Memory ที่อยู่ใน CPU มีหน้าที่เก็บชุดคำสั่งที่ใช้งานบ่อย ๆ

โดยใน CPU จะมีหน่วยความจำของตัวเองโดยเฉพาะ คือ

1. รีจิสเตอร์ (Register) เป็นแหล่งเก็บข้อมูลแบบพิเศษที่ทำให้ CPU สามารถดึงข้อมูลไปใช้งานได้เร็วกว่าหน่วยความจำธรรมดา
2. แคช (Cache) เป็นแหล่งเก็บข้อมูลที่จะเก็บชุดคำสั่งที่เราใช้งานบ่อยๆ เพื่อใช้ในการใช้งานครั้งต่อไปโดยไม่ต้องเรียกข้อมูลจากแหล่งต้น

ขั้นตอนการประมวลผลของ CPU มี 4 ขั้นตอน คือ

1. Fetch เป็นกระบวนการที่หน่วยควบคุมนำชุดคำสั่งจาก RAM มาเก็บไว้ในรีจิสเตอร์
2. Decode เป็นกระบวนการถอดรหัสชุดคำสั่งเพื่อเตรียมส่งไปยังหน่วย ALU
3. Execute เป็นกระบวนการประมวลผลชุดคำสั่งโดยหน่วย ALU เพื่อที่จะทำงานตามที่ต้องการ โดยจะประมวลผลทีละคำสั่ง
4. Store เป็นการจัดการเก็บผลลัพธ์ที่ได้จากการประมวลผล โดยจะส่งกลับไปเก็บใน RAM



โดยขั้นตอนการประมวลผลของ CPU ทั้ง 4 ขั้นนี้เรียกว่า 1 Machine cycle

ความเร็วในการประมวลผล (Processing Speed)

ความเร็วในการประมวลผลของ CPU สามารถวัดได้จากสัญญาณนาฬิกา (Clock Speed) ที่ถูกสร้างโดยเมนบอร์ด โดยในสัญญาณนาฬิกาจะวัดว่าสามารถทำ Machine cycle ได้กี่ครั้งใน 1 วินาที โดยสัญญาณนาฬิกาของ CPU จะทำงานหลายล้านรอบใน 1 วินาที โดยมีหน่วยวัดเป็น Hertz, MHz, GHz เช่น CPU Intel Core i5 มีความเร็ว 3.5 GHz หมายถึงว่าสามารถทำ Machine cycle 3.5 พันล้านรอบต่อวินาที

หน่วยความจำ (Memory)

หน่วยความจำเป็นพื้นที่สำหรับเก็บชุดคำสั่งและข้อมูล สามารถแบ่งได้เป็น 2 ชนิด คือ

1. หน่วยความจำแบบลบเลือน (Volatile Memory) เป็นหน่วยความจำที่ต้องมีไฟเลี้ยงตลอดเวลา ถ้าไม่มีไฟ หรือ ปิดเครื่องข้อมูลในนั้นจะหายไป เช่น RAM
2. หน่วยจำแบบไม่ลบเลือน (Non-Volatile Memory) เป็นหน่วยความจำที่ข้อมูลยังคงอยู่แม้ไม่มีไฟเลี้ยงก็ตาม เช่น ROM, CMOS, หน่วยความจำแบบ Flash

RAM (Random Access Memory)

เป็นหน่วยความจำหลักของคอมพิวเตอร์และเป็นหน่วยความจำแบบลบเลือน เมื่อคอมพิวเตอร์ถูกเปิดขึ้น ข้อมูลของระบบปฏิบัติการจะถูกโหลดไปเก็บไว้ที่ RAM และเมื่อมีการเรียกใช้งานโปรแกรมต่าง ๆ ข้อมูลของโปรแกรมนั้นก็จะถูกโหลดไปที่ RAM เพื่อรอการประมวลผลในซีพียู

ซึ่งแรมแบ่งได้ 2 ชนิด คือ

1. Static RAM (SRAM)

ข้อดี: ใช้ไฟน้อย, ความเร็วสูงมาก

ข้อเสีย: ราคาสูง, ความจุน้อย

2. Dynamic RAM (DRAM)

ข้อดี: ราคาถูกกว่า

ข้อเสีย: ความเร็วต่ำ

ROM (Read Only Memory)

รอมเป็นหน่วยความจำใช้เก็บข้อมูลแบบถาวร โดยแบ่งได้เป็น 3 แบบคือ

1. PROM (Programmable ROM) คือหน่วยความจำที่ไม่สามารถแก้ไขข้อมูลที่บันทึกไว้ได้ โดยส่วนมากแล้วจะเป็นโปรแกรมที่ถูกบันทึกมาจากโรงงาน

2. EPROM (Erasable Programmable ROM) เป็นหน่วยความจำที่สามารถลบข้อมูลหรือโปรแกรมใหม่ได้ ซึ่งการลบข้อมูลจะใช้วิธีการฉายแสง Ultra Violet เราจะสังเกตอุปกรณ์ที่เป็น EPROM ได้จากลักษณะของอุปกรณ์ที่มีแผ่นกระจกใส ๆ อยู่ตรงกลางอุปกรณ์

3. EEPROM (Electrical EPROM) เป็นหน่วยความจำที่สามารถลบข้อมูลและโปรแกรมใหม่ได้ด้วยการใช้ไฟฟ้าในการลบ ซึ่งแตกต่างจากแบบ EPROM ที่ต้องใช้การฉายแสง Ultra Violet ในการลบข้อมูล

หน่วยความจำแฟลช (Flash Memory)

เป็นหน่วยความจำใช้เก็บข้อมูลแบบถาวรคล้ายรอม แต่ข้อมูลที่บรรจุในหน่วยความจำแบบแฟลชสามารถลบและบันทึกซ้ำได้เรื่อย ๆ

CMOS (Complementary Metal-Oxide Semiconductor)

เป็นชิปชนิดพิเศษ ซึ่งทำจากวัสดุกึ่งตัวนำ มีจุดเด่นที่ใช้พลังงานน้อย โดยข้อมูลส่วนใหญ่ที่บันทึกใน CMOS ได้แก่ BIOS, วันที่และเวลา ทำให้เวลาสามารถเดินต่อไปได้แม้ปิดเครื่องเอาไว้ เนื่องจาก CMOS ใช้พลังงานจากถ่านที่อยู่บนเมนบอร์ด

สล็อตเพิ่มขยาย (Expansion Slots)

คือสล็อตสำหรับเสียบการ์ดเพื่อเชื่อมต่ออุปกรณ์เพิ่มเติม ทำให้เพิ่มขีดความสามารถของคอมพิวเตอร์ โดยสล็อตจะมีได้หลายชนิด เช่น PCI, PCIe และการ์ดที่นำมาเสียบกับสล็อตนั้นมีหลายชนิด เช่น

การ์ดจอ (Graphic Card)

เป็นอุปกรณ์ที่รับข้อมูลเกี่ยวกับการแสดงผล (ภาพ, วิดีโอ, กราฟฟิก) จากหน่วยความจำ มาคำนวณ และประมวลผล จากนั้นจึงส่งข้อมูลในรูปแบบสัญญาณเพื่อนำไปแสดงผลยังอุปกรณ์แสดงผล เช่น จอภาพ

การ์ดเสียง (Sound Card)

คืออุปกรณ์คอมพิวเตอร์ที่ทำหน้าที่แปลงข้อมูลดิจิทัลที่เก็บรายละเอียดเสียงต่าง ๆ กลับมาเป็น สัญญาณเสียงในรูปแบบสัญญาณอนาล็อก

การ์ดไวเลส (Wireless Card)

เป็นอุปกรณ์ที่ใช้สำหรับรับส่งข้อมูลในระบบเครือข่ายแบบไร้สาย เช่น ไวไฟ

การ์ดแลน (LAN Card)

เป็นอุปกรณ์ที่ใช้สำหรับรับส่งข้อมูลจากเครื่องคอมพิวเตอร์เครื่องหนึ่งไปยังอีกเครื่องหนึ่ง หรือไปยัง อุปกรณ์อื่น ๆ ในระบบเครือข่าย

การ์ดบันทึกภาพหน้าจอ (Video Capture Card)

เป็นอุปกรณ์สำหรับบันทึกภาพต่าง ๆ ที่ถูกส่งไปที่อุปกรณ์ปลายทางโดยจะทำให้ได้คุณภาพที่สูง มากกว่าโปรแกรมที่เอาไว้อิงบันทึกภาพทั่ว ๆ ไปและไปกินแรงเครื่อง

หน่วยเก็บข้อมูล

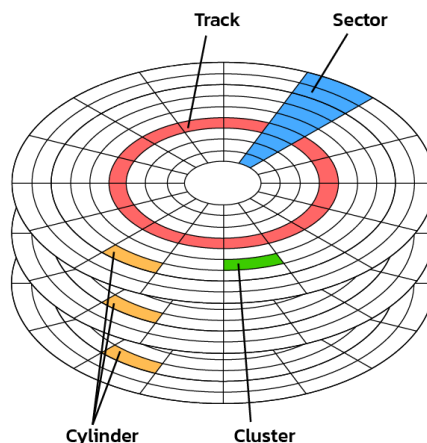
ทำหน้าที่เก็บข้อมูลหรือโปรแกรมไว้เพื่อใช้งานในอนาคต เนื่องจากข้อมูลที่คอมพิวเตอร์ทำงานจะอยู่ในแรมหรือหน่วยงานความจำที่ลบเลือนได้ ซึ่งเมื่อปิดเครื่องคอมพิวเตอร์ข้อมูลจะสูญหายไปหมด เมื่อต้องการใช้ข้อมูลหรือโปรแกรมเดิมอีกครั้งจึงต้องมีหน่วยเก็บข้อมูลขึ้นมา

Harddisk Drive (HDD)

คืออุปกรณ์คอมพิวเตอร์ที่บรรจุข้อมูลแบบไม่ลบเลือน มีลักษณะเป็นแผ่นอลูมิเนียมแข็งเรียกว่า Platters ที่เคลือบด้วยสารแม่เหล็ก ซึ่งสามารถบรรจุข้อมูลได้ Platter ถูกนำมาเรียงทับกันหลาย ๆ ชั้น บนแกนขับ โดยในช่องว่างของแต่ละชั้นจะมีหัวอ่าน/เขียนที่สามารถเคลื่อนที่เข้าออกได้แทรกอยู่บนแต่ละแผ่น

พื้นที่ของแพล็ตเตอร์สามารถแบ่งออกได้เป็นดังนี้

1. Track – การแบ่งพื้นที่เป็นวงกลมเหมือนลู่วิ่ง
2. Sector – การแบ่ง Track เป็นส่วน ๆ เหมือนขนมเค้ก
3. Cylinder – Track เดียวกันแต่อยู่คนละ Platter
4. Cluster – กลุ่มของชิ้นส่วน Sector โดยจะเป็นหน่วยเล็กที่สุดที่เอาไว้บรรจุข้อมูล



Solid State Drive (SSD)

อุปกรณ์จัดเก็บข้อมูลที่ใช้ Flash memory จำนวนมากประกอบรวมกันทำให้มีความสามารถในการเข้าถึงข้อมูลได้รวดเร็วกว่า HDD มาก

- ข้อดี: อ่าน/เขียนข้อมูลได้เร็ว น้ำหนักเบา กินไฟน้อยกว่า ตกแล้วพังยากกว่า HDD (แต่ทางที่ดีไม่ควรทำตกนะ)
- ข้อเสีย: แพงกว่า, การกู้ข้อมูลจะยากกว่า HDD, หากไม่มีไฟเลี้ยงเป็นเวลานาน ข้อมูลอาจไม่สมบูรณ์

RAID (Redundant Array of Inexpensive Disk)

การนำเอาฮาร์ดดิสก์ตั้งแต่ 2 ตัวขึ้นไปมาทำงานร่วมกันเสมือนเป็นตัวเดียวกันโดยที่จะทำให้มีประสิทธิภาพสูงขึ้น หรือมีการสำรองข้อมูลเพื่อป้องกันการสูญหายในกรณีที่เกิดความผิดพลาดของอุปกรณ์

Software

Software คือ ส่วนชุดคำสั่งที่ใช้ในการจัดเก็บและประมวลผลข้อมูลในคอมพิวเตอร์ และนอกจากคอมพิวเตอร์ แล้วก็ยังสามารถใช้งานในอุปกรณ์อื่น ๆ ได้ เช่น หุ่นยนต์ในโรงงาน หรือเครื่องใช้ไฟฟ้า แบ่งได้ 2 ประเภท คือ

1. **ซอฟต์แวร์ระบบ (Operating System)** คือซอฟต์แวร์ที่สร้างขึ้นเพื่อมาเป็นตัวกลางระหว่างฮาร์ดแวร์และแอปพลิเคชันซอฟต์แวร์ โดยหน้าที่หลัก ๆ ก็คือ การจัดสรรทรัพยากรภายในเครื่อง การรับ-ส่งข้อมูล การจัดเก็บข้อมูล ฯลฯ

ลักษณะการทำงานของ OS สามารถแบ่งได้เป็น 2 รูปแบบ คือ

- Single tasking เป็น OS ที่ทำงานได้แค่ทีละโปรแกรม เช่น DOS
- Multitasking เป็น OS ทำงานได้หลายโปรแกรมพร้อมกัน เช่น OS ส่วนใหญ่ในปัจจุบันอย่าง Windows, macOS

2. **ซอฟต์แวร์ประยุกต์ (Application Software)** คือ Software ที่ถูกสร้างขึ้นมาสำหรับงานเฉพาะด้านที่ไม่เกี่ยวกับการควบคุมระบบ โดยซอฟต์แวร์ประยุกต์จะไม่สามารถทำงานด้วยตัวเองได้ จะต้องมียุทธรณ์ระบบมาช่วยจัดการ

โปรแกรมอรรถประโยชน์ (Utility Software)

คือซอฟต์แวร์ที่ช่วยในการดูแลระบบคอมพิวเตอร์ ส่วนใหญ่มักจะแถมมากับซอฟต์แวร์ระบบอยู่แล้ว เพียงแต่โปรแกรมอื่นนอกเหนือจากที่มีมาให้อาจจะมียุทธรณ์ที่มากกว่า เพื่อเป็นตัวเลือกรุ่นที่หลากหลยให้กับผู้ใช้ เช่น Anti-virus, โปรแกรมบีบอัดไฟล์, Disk cleaner เป็นต้น

Network

เครือข่ายคอมพิวเตอร์ (Computer Network) เป็นการเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์ตั้งแต่ 2 เครื่องขึ้นไปเข้าด้วยกันโดยใช้สื่อกลาง เพื่อให้คอมพิวเตอร์ สามารถสื่อสารกันได้ นอกจากการแลกเปลี่ยนข้อมูลแล้วยังสามารถใช้อุปกรณ์ต่าง ๆ ในเครือข่ายร่วมกันได้อีกด้วย

Network Devices

Server หรือ เครื่องแม่ข่าย เป็นเครื่องคอมพิวเตอร์หลักในเครือข่ายที่ทำหน้าที่จัดเก็บข้อมูลและให้บริการ Services หรือทรัพยากรกับคอมพิวเตอร์เครื่องอื่น ๆ ในเครือข่าย

Client หรือ เครื่องลูกข่าย เป็นคอมพิวเตอร์ธรรมดา สมาร์ทโฟนหรืออุปกรณ์ต่าง ๆ โดยที่ไคลเอนต์จะร้องขอ บริการและเข้าถึงไฟล์ ข้อมูลที่จัดเก็บในเซิร์ฟเวอร์

Hub หรือเรียกอีกอย่างว่า **Repeater** เป็นอุปกรณ์ที่ใช้เชื่อมต่อกลุ่มคอมพิวเตอร์ฮับมีหน้าที่รับส่งเฟรมข้อมูล ทุกเฟรมที่ได้รับจากพอร์ตใดพอร์ตหนึ่งไปยังพอร์ตที่เหลือ (รับข้อมูลมาจากเครื่องใดเครื่องหนึ่งและก็กระจาย ข้อมูลให้ทุกคน)

Switch เป็นอุปกรณ์เครือข่ายที่ทำหน้าที่ส่งข้อมูลที่รับมาจากพอร์ตหนึ่งไปพอร์ตเฉพาะที่เป็นปลายทาง เท่านั้น (ต่างจาก Hub ตรงที่ Switch จะรับข้อมูลมาแล้วส่งให้เฉพาะปลายทางที่ระบุไว้เท่านั้น) นิยมเชื่อมต่อ แบบนี้มากกว่าฮับเพราะลดปัญหาการชนกันของข้อมูล

Router เป็นตัวกลางในการส่งต่อข้อมูลไปยังเครือข่ายอื่น ทำหน้าที่คล้าย Switch ที่ทำให้เชื่อมต่อได้หลาย เครื่องพร้อมกันหน้าที่หลักของ Router คือการหาเส้นทางในการส่งผ่านข้อมูลที่ดีที่สุด

Bridge เป็นอุปกรณ์เชื่อมโยงเครือข่ายที่แยกจากกัน โดยจะทำหน้าที่เหมือนเป็นสะพานเชื่อมระหว่างวงแลน เข้าหากันทำให้วงแลน 2 วง สามารถเชื่อมต่อเข้าหากันได้ และสามารถเข้าถึงอุปกรณ์ของอีวงแลนหนึ่งได้

Gateway เป็นอุปกรณ์ฮาร์ดแวร์ที่เชื่อมต่อเครือข่ายต่างประเภทเข้าด้วยกัน เช่น การใช้ Gateway ในการ เชื่อมต่ออินเทอร์เน็ตจาก ISP เข้ากับเครือข่ายภายในบ้าน

Types of Computer Network

การแบ่งประเภทเครือข่ายคอมพิวเตอร์นั้นสามารถจำแนกออกได้เป็นหลายประเภทแล้วแต่เกณฑ์ที่ใช้แต่โดยทั่วไปจำแนกประเภทของเครือข่ายมีอยู่ 3 วิธี คือ

1. แบ่งตามขนาด ได้แก่

PAN (Personal Area Network) คือ เทคโนโลยีการเข้าถึงไร้สายในพื้นที่เฉพาะส่วนบุคคล โดยมีระยะทางไม่เกิน 1 เมตร และมีอัตราการรับส่งข้อมูลความเร็วสูงมาก (สูงถึง 480 Mbps)

LAN (Local Area Network) คือ ระบบเครือข่ายท้องถิ่นหรือภายในองค์กรต่าง ๆ ไม่ว่าจะเป็นโรงเรียน ภายในที่อยู่อาศัย เป็นเน็ตเวิร์คในระยะทางไม่เกิน 10 กิโลเมตร

MAN (Metropolitan Area Network) คือ ระบบเครือข่ายเมือง เป็นเน็ตเวิร์คที่จะต้องใช้โครงข่ายการสื่อสารขององค์กรโทรศัพท์ หรือการสื่อสารแห่งประเทศไทย

WAN (Wide Area Network) คือ ระบบเครือข่ายกว้างไกล หรือเรียกได้ว่าเป็น World Wide ของระบบเน็ตเวิร์ค โดยจะเป็นการสื่อสารในระดับประเทศ ข้ามทวีปหรือทั่วโลก เชื่อมต่อผ่านเครือข่าย International Internet Gateway

2. แบ่งตามลักษณะการทำงาน

เครือข่ายแบบเพียร์ทูเพียร์ (Peer-to-peer Network) คือ เครือข่ายที่ไม่มีการเลือกเครื่องใด เป็น Server หรือ Client ซึ่งทำให้แต่ละเครื่องสามารถเข้าถึงข้อมูล และสามารถเผยแพร่ข้อมูลร่วมกันได้ แต่มีปัญหาในด้านความปลอดภัย เพราะทุกคนสามารถเข้าถึงได้ทุกอย่างของกันและกัน

เครือข่ายแบบไคลเอนท์เซิร์ฟเวอร์ (Client/Server Network) คือ เครือข่ายที่มีคอมพิวเตอร์เครื่องหนึ่งทำหน้าที่เป็น Server (เปรียบเสมือนเครื่องแม่) โดยทำหน้าที่เป็นศูนย์กลางของระบบเครือข่ายและทรัพยากรต่าง ๆ ให้กับเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Client (เปรียบเสมือนเครื่องลูก) ซึ่งอยู่ในตัวอย่างเครือข่ายประเภทนี้ได้แก่ Web Server, Mail Server

3. แบ่งตามระดับความปลอดภัยของข้อมูล

อินเทอร์เน็ต (Internet) เป็นเครือข่ายที่ความปลอดภัยของข้อมูลเป็นศูนย์หากไม่มีการป้องกัน เนื่องจากอินเทอร์เน็ตถูกสร้างให้เป็นเครือข่ายสาธารณะ

อินทราเน็ต (Intranet) เป็นเครือข่ายที่มีความปลอดภัยของข้อมูลค่อนข้างสูงเนื่องจากเป็นเครือข่ายที่ใช้ภายในวงปิดเช่นองค์กร

เอ็กสทราเน็ต (Extranet) เป็นเครือข่ายกึ่งอินทราเน็ต สามารถอธิบายได้ในรูปของการเชื่อมต่อระหว่างองค์กรจะยอมให้ผู้ใช้งานสามารถเข้าถึงได้ตามที่อนุญาตเท่านั้น

Network Topologies

โครงสร้างเครือข่ายคอมพิวเตอร์ คือลักษณะการเชื่อมต่อทางกายภาพระหว่างอุปกรณ์คอมพิวเตอร์ต่าง ๆ ในระบบเครือข่าย ซึ่งหากจะแบ่งประเภทของโครงสร้างเครือข่ายกันจริง ๆ ตามหลักวิชาการที่ใช้กันมาตั้งแต่สมัยก่อน ๆ นั้น ก็สามารถแบ่งออกได้เป็น 5 แบบ คือ

1. โครงสร้างเครือข่ายแบบสตาร์ (Star Network) ลักษณะการเชื่อมต่อของโครงสร้างเครือข่ายแบบสตาร์จะคล้าย ๆ กับดาวกระจาย โดยมีอุปกรณ์ประเภท Hub หรือ Switch เป็นศูนย์กลางการเชื่อมต่อ
 - ข้อดี การติดตั้งและการดูแลรักษาทำได้ง่าย และเพิ่มอุปกรณ์ในเครือข่ายได้ง่าย
 - ข้อเสีย เสียค่าใช้จ่ายมากกับอุปกรณ์และสายเคเบิล
2. โครงสร้างเครือข่ายแบบบัส (Bus Network) คือลักษณะการเชื่อมต่อแบบอนุกรม โดยใช้สายเคเบิลเส้นยาวต่อเนื่องกันไป
 - ข้อดี การติดตั้งและการดูแลรักษาทำได้ง่าย ค่าใช้จ่ายน้อยและเพิ่มอุปกรณ์ในเครือข่ายได้ง่าย
 - ข้อเสีย สามารถเกิดข้อผิดพลาดในเครือข่ายได้ง่าย เช่น การชนกันของข้อมูล และตรวจหาอุปกรณ์ในเครือข่ายที่เสียได้ยาก
3. โครงสร้างเครือข่ายแบบวงแหวน (Ring Network) เป็นเครือข่ายที่เชื่อมต่อคอมพิวเตอร์ด้วยสายเคเบิลยาวเส้นเดียว ในลักษณะวงแหวน การรับส่งข้อมูลในเครือข่ายวงแหวนจะใช้ทิศทางเดียวเท่านั้น
 - ข้อดี ผู้ส่งสามารถส่งข้อมูลไปยังผู้รับได้หลายเครื่องพร้อมกัน ไม่มีการชนกันของ สัญญาณข้อมูลที่ส่งออกไป คอมพิวเตอร์ทุกเครื่องในเน็ตเวิร์คมีโอกาที่จะส่งข้อมูลได้อย่างทัดเทียมกัน
 - ข้อเสีย ถ้าหากมีเครื่องใดเครื่องหนึ่งในเครือข่ายเสียหาย ข้อมูลจะไม่สามารถส่งผ่านไปยังเครื่องต่อ ๆ ไปได้และจะทำให้เครือข่ายทั้งเครือข่ายหยุดชะงักได้ และจะต้องเสียเวลาในการตรวจสอบ
4. โครงสร้างเครือข่ายแบบตาข่าย (Mesh Network) เป็นการเชื่อมต่ออุปกรณ์ในเครือข่าย ทั้งหมดเข้าด้วยกัน มีลักษณะคล้ายตาข่ายที่อุปกรณ์ทุกตัวมีการเชื่อมโยงถึงกัน จึงทำให้อุปกรณ์ทุกชิ้นสามารถติดต่อกันได้โดยตรง ต่อให้มีอุปกรณ์หรือสายเส้นใดเสีย ระบบก็จะหาเส้นทางอื่นในการส่งข้อมูลให้ถึงเป้าหมายได้
 - ข้อดี มีความปลอดภัยสูงเนื่องจากเป็นกันส่สารกันระหว่าง 2 เครื่อง และด้วยเหตุนี้จึงทำให้สามารถใช้ Bandwidth ได้อย่างเต็มประสิทธิภาพ
 - ข้อเสีย เป็นรูปแบบการเชื่อมต่อเครือข่ายที่สิ้นเปลืองสายสื่อสารมากที่สุดทำให้มีราคาสูงและยากต่อการติดตั้ง/บำรุงรักษา
5. โครงสร้างเครือข่ายแบบผสม (Hybrid Network) เป็นเครือข่ายคอมพิวเตอร์ที่ผสมผสานระหว่างรูปแบบต่าง ๆ หลาย ๆ แบบเข้าด้วยกันคือจะมีเครือข่ายคอมพิวเตอร์ย่อย ๆ หลาย ๆ เครือข่ายเพื่อให้เกิดประสิทธิภาพสูงสุดในการทำงาน
 - ข้อดี ไม่ต้องเสียค่าใช้จ่ายในการวางสายเคเบิลมากนัก, สามารถขยายระบบได้ง่าย และเสียค่าใช้จ่ายน้อย
 - ข้อเสีย ถ้ามีเครื่องคอมพิวเตอร์จำนวนมาก ๆ อาจทำให้เกิดการคับคั่ง (คอขวด) ของเน็ตเวิร์กซึ่งจะทำให้ระบบช้าลงได้

Ethics

หรือ จริยธรรม คือ หลักเกณฑ์ที่มีการตกลงร่วมกัน เพื่อเป็นแนวทางปฏิบัติในสังคมใช้เพื่อจำแนกแยกแยะว่าสิ่งไหนถูก สิ่งไหนผิด โดยจริยธรรมที่เกี่ยวข้องกับคอมพิวเตอร์และเทคโนโลยีสารสนเทศ เรารู้จักกันในชื่อว่า PAPA

- ความเป็นส่วนตัว (Information Privacy)
- ความถูกต้อง (Information Accuracy)
- ความเป็นเจ้าของ (Information Property)
 - Copyright (ลิขสิทธิ์) – ห้ามนำผลงานไปใช้ทำซ้ำ โดยเด็ดขาด ต้องขออนุญาตจากเจ้าของ
 - Copyleft (นิรสิทธิ์) – อนุญาตให้นำผลงานไปใช้ต่อยอดได้ในบางกรณีเช่น แจกให้อ่านหรือแจกให้ดูฟรีแต่ห้ามทำเพื่อจัดจำหน่าย และต้องระบุชื่อเจ้าของที่แท้จริงด้วย
- Shareware – ซอฟต์แวร์ให้ทดลองใช้ได้ก่อนที่จะตัดสินใจซื้อ
- Freeware – ซอฟต์แวร์ใช้งานได้ฟรี คัดลอก และเผยแพร่ให้ผู้อื่นได้
- การเข้าถึงข้อมูล (Data Accessibility)
 - Hacker คือกลุ่มคนที่มีความสามารถด้านคอมพิวเตอร์และเครือข่ายเป็นอย่างดีโดยอาจจะมีทั้งแฮกเกอร์ที่ดีและไม่ดี

ประเภทของแฮกเกอร์

White hat - เป็นแฮกเกอร์ที่คอยทำหน้าที่ป้องกันระบบ โดยการเจาะระบบนั้นได้รับอนุญาตจากผู้ดูแลให้ทำการเจาะระบบแล้ว เพื่อทดสอบช่องโหว่และแจ้งผู้ดูแลให้นำไปแก้ไขหรือปรับปรุง

Gray hat - เป็นแฮกเกอร์ที่มีจุดประสงค์ไม่แน่นอน อาจจะเป็นการโจมตีระบบ หรือบางครั้งก็จะแจ้งกลับไปยังผู้ดูแลแล้วแต่จุดประสงค์หรือความต้องการ

Black hat – เป็นแฮกเกอร์ที่มุ่งทำลายระบบเพื่อผลประโยชน์ตัวเองซึ่งเป็นการกระทำในทางลบ

Malicious Software – Malware

1. Computer Virus หรือที่เราคุ้นเคยกันกับคำว่า "ไวรัส" ซึ่งเป็นชื่อที่เลียนแบบกับสิ่งมีชีวิตเพราะ โปรแกรมชนิดนี้จะสามารถแพร่กระจายได้เหมือนกับเชื้อไวรัส โดยโปรแกรมนี้อาจติดต่อกับไฟล์สื่ไฟล์ได้ ไม่ว่าจะเป็นจากในระบบเดียวกันหรือเคลื่อนย้ายข้ามระบบไปที่คอมพิวเตอร์เครื่องอื่นผ่านการฝังตัวเองไปตามโปรแกรมต่างๆ ก็ได้ ซึ่งเมื่อผู้ใช้เปิดใช้งานโปรแกรมไวรัสก็จะทำงาน โดยไวรัสจะสามารถทำลายได้ทั้ง Hardware, Software และข้อมูล
2. Worms (เวิร์ม) เป็น Malware ชนิดหนึ่งที่มีคุณลักษณะพิเศษคือ ไม่ต้องอาศัยตัวกลาง เช่น ไฟล์หรือโปรแกรม ในการแพร่กระจาย เนื่องจาก Worms สามารถจำลองตัวเองขึ้นมาได้ นอกจากนี้ Worms บางชนิดไม่จำเป็นต้องอาศัยผู้ใช้งานในการแพร่กระจายตัวมันเองอีกด้วย (ผู้ใช้ไม่จำเป็นต้องเปิดโปรแกรม Worms ก็สามารถทำงานได้ด้วยตัวเอง) Worms มีความสามารถในการทำลายระบบคอมพิวเตอร์สูง ซึ่งหากยิ่งกระจายตัวเยอะเท่าไรความสามารถในการทำลายก็เยอะขึ้นมากเท่านั้น

3. Trojan House (ม้าโทรจัน) เป็น Malware ที่ดูเหมือนจะไม่ใช่พิษเป็นภัยหรืออาจจะเป็นประโยชน์ต่อผู้ใช้เลยด้วยซ้ำ แต่ข้างในโปรแกรมจะแฝงส่วนที่เป็นอันตรายเอาไว้ ซึ่งหากผู้ใช้งานโปรแกรมขึ้นมาก็เสี่ยงต่อการถูกลักลอบขโมยข้อมูล หรืออาจทำให้ระบบถูกทำลายได้
4. Spyware (สปายแวร์) เป็น Malware ที่ถูกเขียนมาเพื่อสอดส่องและเก็บข้อมูลการใช้งานของผู้ใช้ เช่น ข้อมูลส่วนตัว ที่อยู่ เบอร์โทร Email รวมถึงข้อมูลสำคัญ เช่น รหัสผ่านหรือข้อมูลบัตรเครดิต เป็นต้น
5. Ransomware (แรนซัมแวร์) เป็น Malware ที่กำลังแพร่หลายมากในปัจจุบัน โดย Ransomware จะถูกออกแบบมาเพื่อทำการเข้ารหัสข้อมูลของผู้ใช้งานหรือระบบคอมพิวเตอร์ เช่น ทำให้ผู้ใช้ไม่สามารถเปิดไฟล์ที่สำคัญได้, ไม่สามารถใช้งาน Website ได้ เป็นต้น ซึ่งถ้าหากอยากถอดรหัสไฟล์นั้น ๆ ก็จะต้องจ่ายเงินให้กับ

W.S.U. คอมฯ

คือพระราชบัญญัติที่ว่าด้วยการกระทำผิดเกี่ยวกับคอมพิวเตอร์ ซึ่งคอมพิวเตอร์ที่ว่านี้ก็เป็นได้ทั้ง PC, Notebook, Smartphone รวมถึงระบบต่าง ๆ ที่ถูกควบคุมด้วยระบบคอมพิวเตอร์ด้วย

8 เรื่องที่ห้ามทำผิดกฎหมายใน W.S.U. คอมพิวเตอร์ฯ

ฉบับล่าสุดได้มีการประกาศใช้เมื่อเดือนพฤษภาคม พ.ศ. 2560 ซึ่งเป็น พ.ร.บ. คอมพิวเตอร์ฯ ฉบับ 2

1. เข้าถึงระบบ หรือข้อมูลของผู้อื่นโดยไม่ชอบ (มาตรา 5-8)
2. แก้ไข ดัดแปลง หรือทำให้ข้อมูลผู้อื่นเสียหาย (มาตรา 9-10)
3. ส่งข้อมูลหรืออีเมลก่อกวนผู้อื่น หรือส่งอีเมลสแปม (มาตรา 11)
4. เข้าถึงระบบ หรือข้อมูลทางความมั่นคงโดยมิชอบ (มาตรา 12)
5. จำหน่ายหรือเผยแพร่ชุดคำสั่งเพื่อนำไปใช้กระทำความผิด (มาตรา 13)
6. นำข้อมูลที่ผิด พ.ร.บ. เข้าสู่ระบบคอมพิวเตอร์ (มาตรา 14)
7. ให้ความร่วมมือ ยินยอม รู้เห็นเป็นใจกับผู้ร่วมกระทำความผิด (มาตรา 15)
8. ตัดต่อ เติม หรือดัดแปลงภาพ (มาตรา 16)

สามารถดูมาตราอื่นๆพร้อมโทษจำคุกและโทษปรับได้ที่:

<http://www.ratchakitcha.soc.go.th/DATA/PDF/2560/A/010/24.PDF>



Introduction to Web Technology

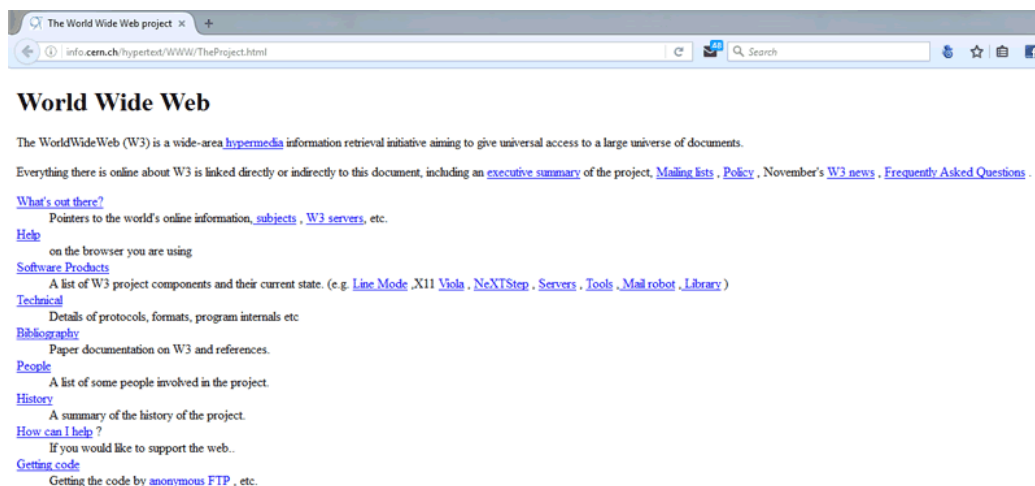
วิวัฒนาการของเว็บไซต์

เมื่อพูดถึงเรื่องของเว็บไซต์หลาย ๆ คนอาจจะรู้จักกันดีอยู่แล้วแต่ก็ยังมีเรื่องราวอีกมากมายที่น่าค้นหาเกี่ยวกับวิวัฒนาการของเว็บไซต์ที่ปัจจุบันเว็บไซต์นั้นก็เปรียบเสมือนกับจักรวาลที่กำลังขยายตัวอย่างต่อเนื่องในทุก ๆ วัน

โดยมีทั้งส่วนที่ผู้ใช้งานมองเห็นและมองไม่เห็น นั่นก็คือการทำงานร่วมกันระหว่างเทคโนโลยีของเว็บไซต์และเว็บเบราว์เซอร์ ที่ผู้ใช้งานทั่วไปไม่สามารถมองเห็นได้ ซึ่งรวมถึงเว็บแอปพลิเคชันจำนวนมาก ที่มีเนื้อหาแบบอินเตอร์แอคทีฟ ทั้งวิดีโอ รูปภาพ และอื่น ๆ

กล่าวโดยรวมนั้น เว็บไซต์ ก็คือการนำเว็บเพจหลาย ๆ เพจมาเชื่อมต่อกันจนกลายเป็นเว็บไซต์เว็บหนึ่งที่มีการเชื่อมโยงลิงก์ถึงกันในแต่ละหน้า โดยที่ไฟล์ของเว็บเพจนั้นจะถูกวางเอาไว้บนเว็บเซิร์ฟเวอร์ (Server-side) และเปิดให้ฝั่งผู้ใช้งาน (Client-side) สามารถเรียกดูเว็บไซต์นั้นได้ผ่านเว็บเบราว์เซอร์ แต่! หลาย ๆ คนอาจจะลืมนึกคิดไปว่า ก่อนที่เว็บไซต์ในปัจจุบันจะถือกำเนิดขึ้นมานั้น ยุคก่อนหน้า เว็บไซต์มีหน้าตาเป็นอย่างไร เรามาย้อนเวลากันครับ

เว็บไซต์ยุคบุกเบิก หรือยุคแรกเริ่ม



เว็บไซต์แรกของโลก <http://info.cern.ch/hypertext/WWW/TheProject.html>

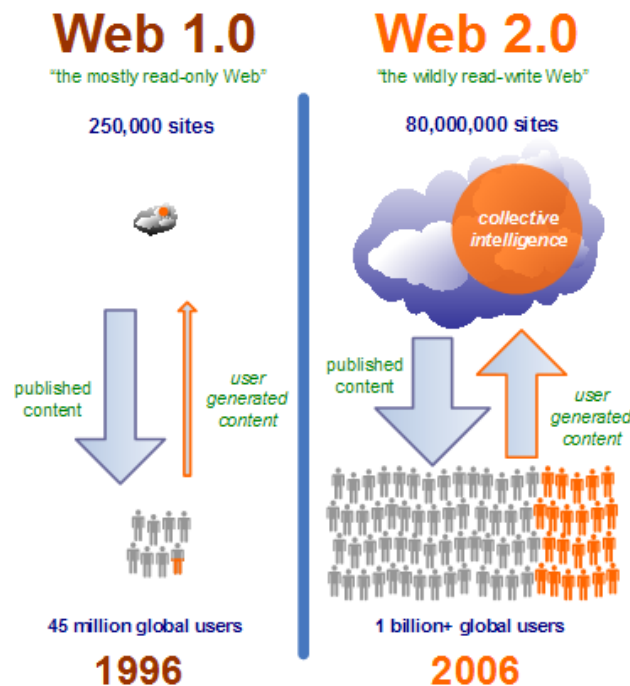
มาทำความเข้าใจเว็บไซต์ยุค 1.0 (WEB 1.0)

เว็บไซต์ยุค 1.0 เป็นเว็บยุคเริ่มต้นบน WWW และยังคงมีให้เห็นอยู่บ้างในปัจจุบันแต่มีจำนวนน้อย โดยผู้ใช้งานไม่สามารถมีส่วนร่วมกับเว็บไซต์ได้ กล่าวคือ ผู้ใช้งานจะสามารถอ่านข้อมูลบนเว็บไซต์ได้อย่างเดียว ไม่สามารถแสดงความคิดเห็นบนเว็บไซต์ หรือตอบโต้แบบอื่น ๆ ได้ โดยเรากันเรียกว่าการสื่อสารแบบทางเดียว หรือ One Way Communication

ซึ่งเว็บ 1.0 นั้นมักจะมีเนื้อหาคล้ายกับหนังสือที่เราอ่านกันอยู่ในปัจจุบัน และมีข้อมูลที่คงที่ ไม่มีการเปลี่ยนแปลง เชื่อมโยงกันด้วยเว็บเพจธรรมดาในรูปแบบ HTML ซึ่งสรุปง่าย ๆ นั่นก็คือ Web 1.0 คือเว็บไซต์ที่สร้างขึ้นมาให้ผู้เข้าชมอ่านได้อย่างเดียว มีไว้เผยแพร่ข้อมูลเพียงอย่างเดียว และผู้ใช้งานไม่สามารถมีส่วนร่วมได้ หรือที่เรียกกันว่า Static web pages

เว็บไซต์ยุคใหม่

เว็บไซต์ยุค 2.0 (WEB 2.0)



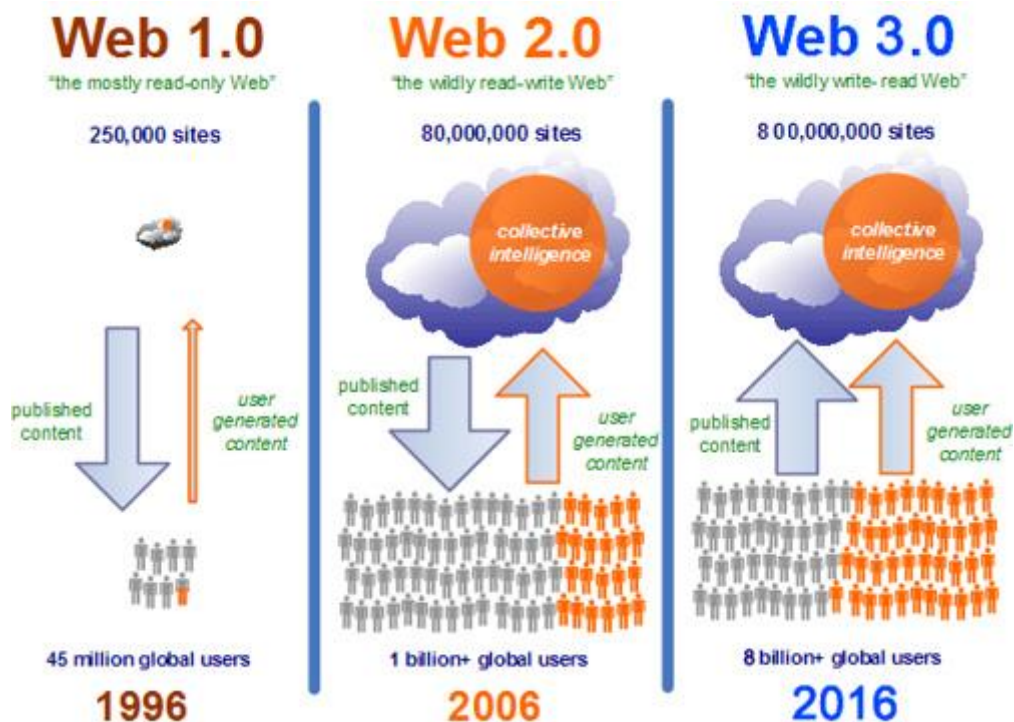
ที่มา <https://sites.google.com/site/thawatchaiarunthong/30/thailand1-0-4-0/theconomy-1g-5g/5g/web-1-0---web-4-0>

มาถึงในช่วงยุคเว็บไซต์ 2.0 กันบ้างนะครับ สำหรับเว็บไซต์ในยุคนี้เป็นเว็บไซต์ที่ยังคงมีให้เห็นอยู่เยอะพอสมควรในยุคปัจจุบัน โดยเว็บ 2.0 เป็นเว็บที่ผู้ใช้งานสามารถมีส่วนร่วมในการโต้ตอบกับเว็บไซต์ได้ (Interactive) อย่างเช่น การเขียนบล็อก, การแชร์รูปภาพ, การแสดงความคิดเห็น และการพูดคุยถกเถียงต่าง ๆ ทั้งจากผู้ใช้งานเว็บไซต์หรือจากเจ้าของเว็บไซต์เอง

ซึ่งในยุคนี้ ผู้เข้ามามียังสามารถสร้างบทความด้วยตัวเอง หรือโพสต์สิ่งต่าง ๆ ได้ อีกด้วย เช่น Social networkชื่อดังอย่าง Twitter รวมถึงเว็บไซต์ให้บริการด้านความบันเทิงทางวิดีโออย่าง YouTube และเว็บไซต์ที่ให้ข้อมูลแก่ชาวโลกอย่างเรา ๆ อย่าง Wikipedia จนกลายเป็นเว็บไซต์ที่มีรูปแบบการสื่อสารแบบสองทาง หรือที่เรียกว่า Two Way Communication

กล่าวโดยสรุปนั่นก็คือเว็บ 2.0 นั้น ผู้ใช้งานจะสามารถ อ่าน - เขียน และมีส่วนร่วมกับข้อมูลที่มีการเปลี่ยนแปลงได้ โดยเชื่อมต่อให้ผู้คนได้เข้าหากัน ที่เรียกกันว่า Dynamic web pages รวมถึงใช้บริการบนหน้าเว็บได้นั้นได้ (Web Service)

เว็บไซต์ยุค 3.0 (WEB 3.0)



ที่มา <https://sites.google.com/site/thawatchaiarunthong/30/thailand1-0-4-0/theconomy-1g-5g/5g/web-1-0---web-4-0>

สำหรับเว็บไซต์ในยุค 3.0 นี้ จะมีวิวัฒนาการต่อมาจากยุค 2.0 โดยจะเน้นไปที่ความฉลาดของอัลกอริทึมเว็บไซต์ที่จะมีส่วนร่วมกับผู้ใช้งานมากขึ้น โดยจะมีการวิเคราะห์พฤติกรรมของผู้ใช้งาน คาดเดาความต้องการของมนุษย์ ซึ่งเมื่อได้ข้อมูลนั้น ๆ มา ระบบจะทำการประมวลผลตามสถานการณ์ และสร้างสิ่งที่ผู้ใช้งานต้องการออกมา

โดยเว็บไซต์ในยุคนี้ส่วนใหญ่จะรองรับการเข้าชมผ่านมือถือได้หมดแล้ว หรือที่เรียกกันว่า Mobile Friendly ทำให้สามารถเข้าถึงข้อมูลได้ง่าย ได้รับข้อมูลที่มีประโยชน์ และตรงต่อความต้องการมากขึ้น จากการวิเคราะห์ข้อมูลและดูแนวโน้มความต้องการของผู้ใช้งาน เช่น Google Maps และอัลกอริทึมอันชาญฉลาดของ Facebook

อ่านเพิ่มเติมได้ที่:

<https://www.f5buddy.com/basic-definitions-of-web-1-0-web-2-0-and-web-3-0/>

เทคโนโลยีของเว็บไซต์

การที่เว็บ ๗ หนึ่งจะถือกำเนิดขึ้นมาได้นั้น นักพัฒนาเว็บไซต์ทั้งหลาย จะต้องเรียนรู้ถึงเทคนิคต่าง ๆ ในการพัฒนา ซึ่งถ้าหากน้อง ๆ อยากรู้มีเว็บไซต์ และมีฟังก์ชันตามที่ตัวเองต้องการ การศึกษาเกี่ยวกับเทคโนโลยีของเว็บไซต์ จะสามารถทำให้น้องมีเว็บไซต์เป็นของตัวเองได้ตามเป้าหมาย

ซึ่งก่อนที่จะเราจะพัฒนาเว็บไซต์ได้นั้น ก็ควรที่จะเรียนรู้ 3 ภาษาหลัก ๆ ที่เป็นพื้นฐานในการพัฒนาเว็บไซต์กันก่อน นั่นก็คือ HTML, CSS และ JavaScript ภาษาเหล่านี้ ถ้าได้เรียนเป็นครั้งแรก น้อง ๆ อาจจะคิดว่ามันดูซับซ้อนยิ่งนัก... แต่ถ้าหากได้ทำความเข้าใจและพยายามที่จะเรียนรู้ รวมถึงสนุกไปกับมัน การที่น้องจะเข้าใจเทคโนโลยีที่ใช้ในการพัฒนาเว็บไซต์ ก็จะไม่ใช่ว่าเรื่องยากอีกต่อไป

เทคโนโลยีเว็บคืออะไร...?

น้อง ๆ เคยได้ยินเทคโนโลยีที่ใช้ในการพัฒนาเว็บไซต์กันมาก่อนหรือเปล่า..? แล้วรู้หรือไม่ว่าความหมายที่แท้จริงของมันคืออะไร? วันนี้เราจะมาเฉลยกันครับ

ตั้งแต่ยุคแรกเริ่ม คอมพิวเตอร์ ไม่สามารถที่จะสื่อสารกับมนุษย์โดยตรงได้ จึงทำให้ ภาษาคอมพิวเตอร์ ได้ถือกำเนิดขึ้นมา เพื่อเป็นตัวกลางในการสื่อสารระหว่างมนุษย์และคอมพิวเตอร์ให้มีความเข้าใจซึ่งกันและกันได้

โดยเช่นเดียวกันกับเทคโนโลยีของเว็บ ที่จะใช้ ภาษามาร์กอัพ (Markup Languages) หรือภาษาที่จะแสดงทั้งข้อมูลและรูปแบบเข้าด้วยกัน เป็นหลัก ซึ่งภาษาที่เรารู้จักกันนั้นก็คือ HTML ที่ใช้ในการทำโครงสร้างเว็บเพจทั้งหมด รวมถึงสื่อมัลติมีเดียต่าง ๆ ที่ใช้ในการสื่อสารกับมนุษย์ผ่านทางเว็บเบราว์เซอร์

ทำความรู้จักกับ Web Browsers

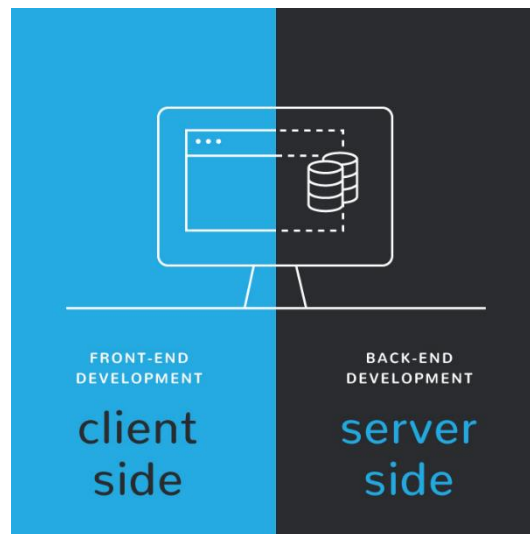


เว็บเบราว์เซอร์ เป็นสิ่งที่พวกเราต่างก็คุ้นเคยกันดี โดยการที่พวกเราจะเข้าถึงเว็บไซต์ได้นั้น จะต้องเรียกดูผ่านทางเว็บเบราว์เซอร์ ซึ่งทางเว็บเบราว์เซอร์จะทำการร้องขอและเข้าถึงข้อมูลที่อยู่บน Web Server ผ่านทาง URL และแสดงผลออกมาให้พวกเราได้เห็นกันในรูปแบบของเว็บเพจ

โดยเปรียบเทียบให้เว็บเบราว์เซอร์นั้นเป็นเสมือนกับตัวแปลภาษาระหว่างเว็บไซต์และมนุษย์ ให้แสดงผลออกมาในรูปแบบที่เราเข้าใจ จากที่เป็นเพียงภาษามาร์กอัพธรรมดา

ตัวอย่างของเว็บเบราว์เซอร์ที่นิยมในปัจจุบัน ได้แก่ **Google Chrome, Edge, Firefox, Safari**

Client-side และ Server-side?



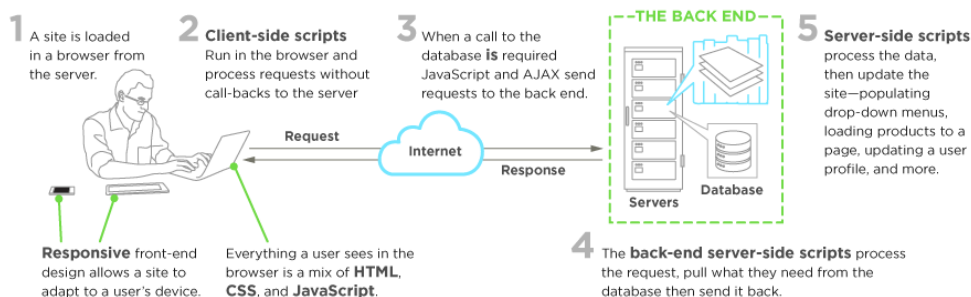
ที่มา <https://blog.devmountain.com/a-programmers-best-friends-the-most-useful-languages-for-web-development/>

น้อง ๆ รู้หรือไม่ว่าเว็บไซต์ทุกเว็บไซต์นั้น ทำงานอยู่บน 3 องค์ประกอบหลักๆ นั่นก็คือเซิร์ฟเวอร์ (Server), ฐานข้อมูล (Database) และฝั่งผู้ใช้งาน (Client) โดยในวันนี้เราจะมาทำความรู้จักกับสิ่งที่พี่ได้กล่าวไว้ข้างต้น

Client-side (Front-end)

FRONT-END DEVELOPMENT

upwork



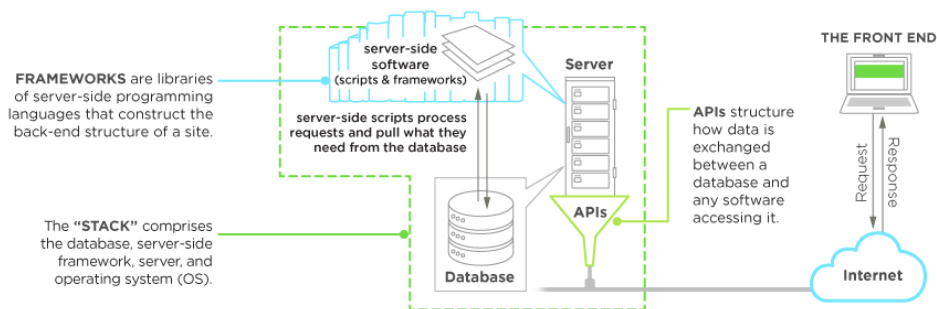
ที่มา: <https://www.upwork.com/hiring/development/front-end-developer/>

ทุกสิ่งที่นี่น้องได้เห็น, ตอบโต้ และมีส่วนร่วมกับสิ่งที่อยู่บนหน้าเว็บ นั่นคือการทำงานของฝั่ง Client-side หรือฝั่งผู้ใช้งานนั่นเอง และนี่คือสิ่งที่เราเรียกกันอีกอย่างหนึ่งสำหรับนักพัฒนาว่า การพัฒนาเว็บไซต์ในฝั่ง Front-end ที่เราจะได้เรียนกันในบทถัดไป

ซึ่งกล่าวง่าย ๆ Client-side นั่นก็คือการที่ผู้ใช้งานได้เปิดเว็บเบราว์เซอร์ขึ้นมา และเรียกดูเว็บไซต์นั้น โดยขณะเดียวกัน เว็บเบราว์เซอร์ก็จะทำการประมวลผลจากสคริปต์ที่ถูกติดตั้งเอาไว้บนเว็บไซต์และแสดงผลออกมานั่นเอง

Server-side (Back-end)

BACK-END DEVELOPMENT & FRAMEWORKS IN SERVER SIDE SOFTWARE



ที่มา <https://www.upwork.com/hiring/development/back-end-web-developer/>

Server-side หรือฝั่ง Back-end เป็นส่วนที่ผู้ใช้งานทั่วไปจะไม่สามารถมองเห็นได้ เพราะในส่วนนี้จะ เป็นกลไกการทำงานของเว็บไซต์ ที่ใช้ในการติดต่อกับฐานข้อมูลที่อยู่บน Server ซึ่งทุกสิ่งเป็นการทำงานในส่วน ของเบื้องหลังเว็บไซต์ทั้งหมด และยังเป็นตัวช่วยในการจัดการและขับเคลื่อนเว็บไซต์ให้สามารถทำงานได้อย่าง มีประสิทธิภาพโดยทำให้เกิดการแลกเปลี่ยนข้อมูลซึ่งกันและกันระหว่าง Server-side และ Client-side

ความแตกต่างระหว่าง Server-side และ Client-side

กล่าวโดยสรุปนั้นทั้ง Server-side และ Client-side จะทำงานร่วมกันอย่างเป็นระบบ โดยที่ Server อาจจะถูกติดตั้งไว้ที่ใดก็ได้บนโลก และเก็บข้อมูลของเว็บไซต์เอาไว้บนฐานข้อมูลเพื่อรอการประมวลผลอยู่ในฝั่ง ของ Back-end โดยจะถูกเรียกใช้จากการประมวลผลคำสั่งจากทางฝั่งของผู้ใช้งาน (Client) และตอบกลับไป เป็นเว็บเพจบนเว็บเบราว์เซอร์ นั่นเอง

ทั้งนี้สคริปต์หรือชุดคำสั่งที่เป็นการทำงานของ Server-side จะถูกวิเคราะห์โดย Web server ของ เว็บไซต์นั้น ๆ เช่นภาษา PHP ที่ใช้ติดต่อกับฐานข้อมูล เป็นต้น และสคริปต์ในฝั่งผู้ใช้งาน (Client-side) ไม่ว่าจะเป็น JavaScript หรือ HTML ก็จะถูกวิเคราะห์และแสดงผลออกมาโดยเว็บเบราว์เซอร์เสมอ

โครงสร้างเว็บไซต์พื้นฐาน

HTML และ CSS คืออะไร?

HTML เป็นสิ่งแรก ที่น้อง ๆ ควรจะเรียนรู้เป็นอย่างแรกก่อนที่จะสร้างเว็บ ๆ หนึ่งขึ้นมา โดยต้องขอบคุณเจ้าภาษา HTML เป็นอย่างมาก ที่ทำให้เว็บเบราว์เซอร์รับรู้ว่าจะแสดงผลอะไรออกมาให้พวกเราได้ชม หลังจากที่เราส่งคำร้องขอไป

สิ่งที่ควรรู้นั้นก็คือ HTML นั้นก็มีเวอร์ชันของมันด้วย ซึ่งปัจจุบันเว็บไซต์ทั่วโลกนั้นกำลังใช้งาน HTML5 ถ้าหากถามว่าแตกต่างจากรุ่นก่อน ๆ ยังไง ก็ต้องบอกก่อนเลยว่าสมัย HTML4 เป็นรุ่นที่มีการใช้งานมาเป็นสิบปีแล้ว เมื่อเวลาผ่านไป HTML4 ก็ได้ถูกปรับเปลี่ยนมาเป็น HTML5 เพื่อให้ตอบสนองความต้องการของผู้ใช้งาน และง่ายต่อการพัฒนามากขึ้น

ก่อนที่จะทำการเขียน HTML Document ในแต่ละหน้านั้น จะต้องเริ่มต้นด้วยการอธิบายก่อนว่า Document นี้มันคืออะไร? โดยใช้

```
<!DOCTYPE html>
```

เป็นจุดเริ่มต้นในการเขียน HTML ทุกครั้ง จากนั้นในทุก Document จะต้องมีการเปิด HTML tag <html> และปิด HTML tag </html> ด้วยทุกครั้ง ส่วนที่จะถูกนำมาแสดงผลบนหน้าเว็บนั้น จะเป็นส่วนที่อยู่ใน tag <body> เท่านั้น ที่ผู้ใช้งานสามารถมองเห็นเว็บออกมาเป็นรูปเป็นร่างได้

นอกจากนี้ยังมีส่วนที่ไม่ถูกนำมาแสดงผลบนหน้าเว็บด้วยเช่นกัน นั่นก็คือส่วนที่อยู่ใน Heading tag <head> หรือเป็นส่วนหัวของเว็บ ซึ่งเนื้อหาที่ถูกบรรจุอยู่ข้างในนั้นจะเรียกรวม ๆ ว่า Metadata ที่จะคอยบอกถึงข้อมูลต่าง ๆ บน HTML Document นั้น ว่ามีอะไรบ้าง และเกี่ยวกับอะไร เช่น ชื่อของเว็บเพจนั้น ๆ และข้อมูลอื่น ๆ ที่ผู้ใช้งานอาจจะไม่สามารถมองเห็นได้โดยตรง เช่น คำอธิบายเว็บไซต์, ชื่อเจ้าของเว็บ เป็นต้น โดยเจ้า Heading tag นั้นจะถูกรวบรวมไว้ระหว่าง <html> และ <body> หรือส่วนเป็นส่วนหัวของเว็บนั่นเอง

ถ้าหากเราอยากเขียน HTML ให้เว็บเป็นรูปเป็นร่างขึ้นมานั้น เราจะต้องมารู้จักกับ Element กันเสียก่อน ..โดยเจ้า HTML Element คือสิ่งที่มักจะบรรจุเอา Start tag และ End tag เอาไว้ด้วยกัน จนกลายเป็นส่วนประกอบต่าง ๆ ของเว็บไซต์ เช่น ปุ่ม, ลิงก์, รูปภาพ และอื่น ๆ โดยมีลักษณะดังนี้

```
<tagname>element content</tagname>
```

HTML ELEMENT

โดยที่ Tags เปิดและปิดนั้น จะบรรจุ Element content ที่เราต้องการเอาไว้ จนรวมเป็นหนึ่ง Element ซึ่งนั่นก็คือเนื้อหาทุกอย่างที่อยู่ภายใต้ Start Tag ไปจนถึง End Tag แต่ทั้งนี้ ในบาง Tag ของ HTML อาจจะไม่จำเป็นต้องมี End Tag ก็ได้ เช่น
 ที่ใช้ในการเว้นบรรทัด เป็นต้น และพอเรามี HTML Element หรือส่วนประกอบต่าง ๆ บนหน้าเว็บกันแล้ว ทีนี้..เรามากำหนดคุณสมบัติของมันด้วย Attribute กันดีกว่า!

HTML Attribute นั้นสามารถนำไปใช้ได้กับทุก HTML Element โดยจะเป็นการกำหนดคุณสมบัติเพิ่มเติมของส่วนประกอบนั้น (Element) ว่าจะให้มันมีคุณสมบัติเป็นอย่างไรบ้าง ตัวอย่างเช่น

```
<a href="https://www.kmitl.ac.th">KMITL</a>
```

```
<element attribute="value">element content</element>
```

ตัวอย่างด้านบนนั้นจะเป็นการสร้างลิงก์ภายใน HTML Document โดยใช้ <a> Tag ซึ่งได้มีการระบุ Attribute href เพื่อบอกตำแหน่งของลิงก์ที่จะเชื่อมโยงไป และมี Element content เป็นคำว่า KMITL นั่นเอง

สิ่งใหม่ ๆ ใน HTML5

- มี tag ใหม่ ๆ เช่น Video, Audio, Canvas
- มี tag ที่ใช้จัด Layout เว็บไซต์ได้มากขึ้นว่าเนื้อหาในนั้นคืออะไร เช่น section, article, aside (ใน HTML4 เราใช้ div กับทุกอย่าง แล้วกำหนด id, class ให้กับ div นั้น ๆ)
- Input แบบใหม่ ๆ เช่น datepicker, email, url, number ฯลฯ (HTML4 มีแค่ text textarea select, radio, checkbox และอีกไม่กี่ตัว อยากได้แปลก ๆ ต้องใช้ JavaScript ช่วย) และมี input placeholder เพิ่มมาด้วย

ภาษา HTML ย่อมาจากคำว่า Hypertext Markup Language ซึ่งเป็นโครงสร้างหลัก หากให้เปรียบกับบ้าน เจ้า HTML ก็เป็นเหมือนกับโครงสร้าง ที่เป็นเสาหลักของบ้าน โดยเมื่อน้อง ๆ ได้รู้จักกับภาษา HTML แล้วต่อไปเราจะมาทำความรู้จักกับเจ้า CSS ที่จะคอยตกแต่งให้บ้านของเราสวยงามขึ้น

CSS เป็นสิ่งที่ควรเรียนรู้เป็นลำดับต่อไป ต่อจากภาษา HTML โดย CSS ย่อมาจากคำว่า Cascading Style Sheets ซึ่งมันจะคอยทำหน้าที่บอกกับ HTML ว่าควรแสดงผล Element วัตถุต่าง ๆ บนหน้าเว็บออกมาเป็นรูปร่างหน้าตาอย่างไรดี

ซึ่งถ้าให้เปรียบเทียบ ๆ CSS ก็เหมือนกับการตกแต่งบ้านให้มีความสวยงามมากขึ้น เช่น การทาสีบ้าน, ขนาดของประตู, ตำแหน่งของเก้าอี้ เราก็สามารถกำหนดได้ เช่นเดียวกันกับสีของเว็บไซต์, ขนาดฟอนต์, ตำแหน่งของวัตถุบนเว็บไซต์ และการตกแต่งอื่น ๆ ก็สามารถทำได้เช่นกัน โดยในปัจจุบันรุ่นของ CSS ที่เว็บเบราว์เซอร์รองรับนั้นก็คือ CSS3



รู้จักกับ Web Development Framework

ปัจจุบันการทำเว็บไซต์สามารถทำได้หลายวิธีและหลายช่องทางมาก ๆ โดยขณะเดียวกันก็ยังมีตัวช่วยที่จะทำให้การพัฒนาเว็บไซต์นั้นมีแบบแผนและมีความเป็นระบบระเบียบมากขึ้น นั่นก็คือการพัฒนาโดยใช้ Web Framework ที่ปัจจุบันมีทั้งฝั่งของ Front-end Framework และ Back-end Framework

...แล้ว Framework มันคืออะไรล่ะ?

อธิบายแบบง่าย ๆ Framework ก็เหมือนกับการที่มีคนเขียนสคริปต์หรือเขียนโปรแกรมเอาไว้อยู่แล้ว โดยที่มีฟังก์ชันต่าง ๆ ถูกสร้างเอาไว้มากมายโดยผู้พัฒนา และเราก็สามารถนำเอาไปใช้งานได้ ซึ่งอาจจะสะดวกกว่าการที่จะมานั่งเขียนฟังก์ชันนั้น ๆ ขึ้นมาใช้เอง

โดยการที่เราจะนำ Framework มาพัฒนาเว็บไซต์ของเราต่อนั้นก็ยังคงยึดภาษาของ Framework นั้น ๆ เป็นภาษาหลักในการพัฒนา ตามปกติ เช่น Laravel Framework จะใช้ภาษา PHP ในการพัฒนาเว็บไซต์ส่วนของ Back-end และ Bootstrap Framework จะใช้งานอยู่สามภาษาหลัก ๆ คือ HTML, CSS และ JavaScript

ทำความเข้าใจกับ Protocols

Protocol คืออะไร? ถ้าจะลองอธิบายด้วยจินตภาพ สมมติคอมพิวเตอร์ 2 เครื่อง ที่ต้องการให้มันมีการรับ-ส่ง ข้อมูลระหว่างกันและกัน แต่คอมพิวเตอร์ทั้งสองเครื่องอาจจะมีทั้ง Hardware และ Software ที่แตกต่างกัน มันจะคุยกันรู้เรื่องได้อย่างไรถ้ามันพูดคนละภาษากัน เหมือนกับน้องกับเพื่อน ๆ ที่บางครั้งก็น้อง ๆ พูดกับเพื่อนบางคนไม่รู้เรื่อง อาจเกิดจากน้องและเพื่อนใช้ Protocol ไม่ตรงกัน ทำให้ Protocol เข้ามาทำหน้าที่ในการเป็นมาตรฐานกลางในการสื่อสาร เพื่อให้ทั้งผู้รับและผู้ส่งข้อมูล ได้รับข้อมูลที่ถูกต้องและตรงกัน

โดยระบบ Internet จะใช้ Protocol สื่อสารมาตรฐานที่ชื่อว่า TCP/IP เป็นภาษาหลัก ดังนั้นหากเครื่องคอมพิวเตอร์ไม่ว่าจะเป็นเครื่องระดับไมโครคอมพิวเตอร์ มินิคอมพิวเตอร์ หรือเมนเฟรมคอมพิวเตอร์ ก็สามารถเชื่อมโยงเข้าสู่อินเทอร์เน็ตได้

TCP/IP

คือชุดของโปรโตคอลที่จะถูกใช้ในการสื่อสารผ่านเครือข่ายอินเทอร์เน็ต โดยมีวัตถุประสงค์เพื่อให้สามารถสื่อสารจากต้นทางข้ามเครือข่ายไปยังปลายทางได้ และสามารถหาเส้นทางที่จะส่งข้อมูลไปได้เองโดยอัตโนมัติ

TCP ย่อมาจากคำว่า **Transmission Control Protocol**

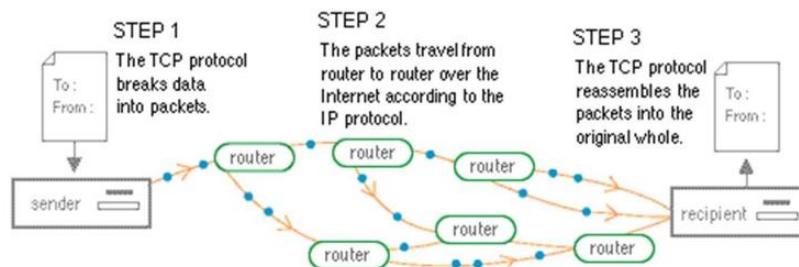
IP ย่อมาจากคำว่า **Internet Protocol**

โดย TCP และ IP จะมีหน้าที่ต่างกัน คือ

- **TCP** จะคอยแยกข้อมูลเป็นส่วนๆ หรือเรียกว่า **Package** ส่งออกไป และ **TCP** ปลายทางจะรวบรวมข้อมูลแต่ละส่วนเข้าด้วยกัน และนำไปประมวลผล ซึ่งในระหว่างทางนั้นข้อมูลจะถูกตรวจสอบความถูกต้อง และหากข้อมูลผิดพลาด **TCP** ปลายทางจะทำการร้องขอไปยัง **TCP** ต้นทางให้ส่งข้อมูลมาใหม่
- **IP** จะทำหน้าที่จัดส่งข้อมูลจากเครื่องต้นทางไปยังปลายทาง โดยดูจาก **IP Address** หรือที่อยู่ของ **IP**

การทำงานของ TCP/IP

How TCP/IP Works



ที่มา <https://www.colocationamerica.com/blog/history-of-ip-address-part-2-tcp-ip>

DNS

DNS ย่อมาจาก **Domain Name Server** โดยหน้าที่ของ **DNS** คือจะทำการแปลงชื่อ **Domain name** เป็น **IP Address** เพื่อให้ง่ายต่อการจดจำ

โดยให้น้อง ๆ ลองคิดภาพตามว่าหากเว็บไซต์เรา แทนที่จะเป็น **www.google.co.th** เราต้องพิมพ์ **IP Address: 216.58.196.35** นี้เข้าไปแทน มันจะทำให้ชีวิตลำบากไหม? บอกเลยว่ามาก! แล้วคิดว่า **Website** จำนวนเท่าไรบนโลกนี้ อาจจะต้องมีสมุดจดกันเลยทีเดียว จุดนี้เลยเป็นสาเหตุให้เกิด **DNS** ขึ้นมา ทำให้ชีวิตเราง่ายขึ้นเยอะมาก

แล้ว Domain name คืออะไร?

Domain name เป็นสิ่งที่สำคัญมากสำหรับเว็บไซต์ ซึ่งจะคอยทำหน้าที่บอกที่อยู่ของเว็บไซต์นั้นให้กับเรา โดยไม่ต้องเข้าใช้งานผ่าน IP Address โดยตรง เพื่อให้่ายต่อการจดจำ แต่ Domain name ก็จะถูกแปลงกลับไปเป็น IP Address ด้วยระบบ DNS ดังที่ได้กล่าวไปอยู่ดี เพื่อให้ระบบสามารถเชื่อมต่อและสื่อสารกันได้

ซึ่ง Domain name สามารถแบ่งออกมาได้เป็น 2 ประเภทคือ โดเมน 2 ระดับ และโดเมน 3 ระดับ โดยที่โดเมน 2 ระดับ จะประกอบไปด้วย

www . ชื่อโดเมน . ประเภทของโดเมน เช่น **www.youtube.com**

- .com คือ บริษัท หรือ องค์กรพาณิชย์
- .edu คือ สถาบันการศึกษา
- .gov คือ องค์กรของรัฐบาล
- .mil คือ องค์กรทางทหาร
- .net คือ องค์กรที่เป็นเกตเวย์ หรือ จุดเชื่อมต่อเครือข่าย
- .org คือ องค์กรเอกชนที่ไม่แสวงผลกำไร

ส่วนโดเมน 3 ระดับ จะประกอบไปด้วย

www . ชื่อโดเมน . ประเภทของโดเมน . ประเทศ เช่น **www.google.co.th**

- .co คือ บริษัท หรือ องค์กรพาณิชย์
- .ac คือ สถาบันการศึกษา
- .go คือ องค์กรของรัฐบาล
- .net คือ องค์กรที่ให้บริการเครือข่าย
- .or คือ องค์กรเอกชนที่ไม่แสวงผลกำไร

สรุปออกมาจับใจความง่าย ๆ และหวังว่าน้อง ๆ จะเข้าใจว่า Protocol ก็คือ ระเบียบ แบบแผนสำหรับการสื่อสารข้อมูล ซึ่ง Protocol มีอยู่หลายตัวด้วยกัน เช่น http, https, ftp และอื่น ๆ อีกมากมาย แล้วแต่จะเลือกใช้ตามสถานการณ์ และชนิดของข้อมูลด้วยกัน ซึ่งที่จะอธิบายอยู่ 4 อย่างหลัก ๆ ที่จำเป็นต้องรู้ด้วยกัน ได้แก่...

HTTP และ HTTPS

Http ย่อมาจาก **Hypertext Transfer Protocol** เป็น **Protocol** ใช้สำหรับการเผยแพร่ข้อมูล ซึ่งมันได้เป็นต้นกำเนิดของ **World Wide Web** ซึ่งมีโครงสร้างเป็นตัวอักษรต่าง ๆ เป็น **Protocol** ที่ใช้ในการเข้าสู่เว็บทั่วไป แต่ในปัจจุบันมี **Protocol** ตัวใหม่ที่มีความปลอดภัยกว่ามาแทน นั่นก็คือ **https** นั่นเอง

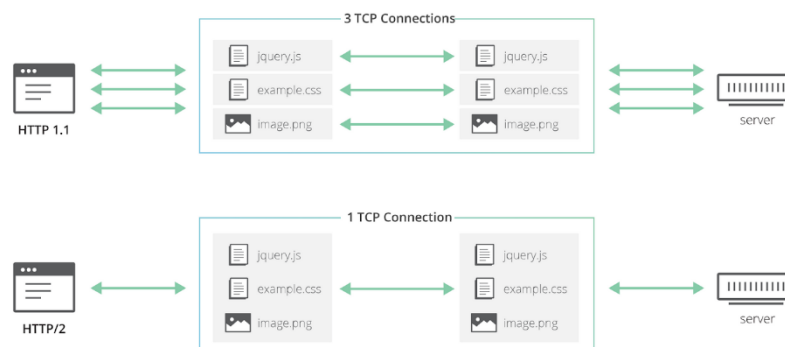
Https ย่อมาจาก **HTTP over Transport Layer Security (TLS)** อธิบายกันง่าย ๆ มันก็คือ **Http** ที่ถูกส่งผ่านชั้นที่ปลอดภัยกว่านั่นเอง ซึ่งเว็บไซต์ที่ใช้ **Https** นั้นทำงานเหมือนกับ **Http** เกือบทุกประการ แต่จะมีการเข้ารหัสข้อมูลก่อนส่งไปที่ **Server** ทำให้ข้อมูลมีความปลอดภัยมากขึ้นนั่นเอง และในอนาคตจะถูกนำมาบังคับใช้ใน **Browser** หลาย ๆ ตัว เช่น **Google Chrome** และ **Firefox**

ประวัติโดยย่อของ HTTP

HTTP/1.0 เป็น **Protocol** ที่เกิดมาในยุคแรก ถูกนำออกมาเป็นมาตรฐานกลางในการใช้งานในปี **1996** ซึ่งเป็นจุดเริ่มต้น ต่อมา ได้มีการสร้าง **SSL** หรือ **TLS** ขึ้นมา พุดง่าย ๆ นั่นก็คือ **https** มาเข้ารหัสเว็บในสมัยนั้นนั้นแหละ แต่เนื่องจากการ **handshake** มันช้าเกินไปจึงมีการคิดค้น **HTTP/1.1** ต่อไป

HTTP/1.1 พัฒนาขึ้นมาเพื่อที่จะลดระยะเวลาในการ handshake ในปี 1997

HTTP/2.0 ผ่านไป 18 ปี (อย่ารู้ว่าปีอะไรให้ลองบวกเลขดู) ได้มีการคิดค้น HTTP/2.0 ขึ้นเนื่องจากปัญหาที่ว่า “ช้า” ในสมัยก่อนเว็บไซต์ยังมีขนาดไม่ใหญ่เลย เมื่อเทียบกับเทคโนโลยีเว็บในปัจจุบัน ทำให้รูปแบบของ HTTP/1.1 ที่สามารถส่งข้อมูลได้เพียงครั้งละ 1 ไฟล์ ไม่ตอบโจทย์เว็บในปัจจุบัน ซึ่ง HTTP/2.0 ทำให้เราสามารถส่งไฟล์ได้หลายไฟล์พร้อมกันได้ผ่านท่อส่งข้อมูลท่อเดียว



ที่มา <https://blog.cloudflare.com/http3-the-past-present-and-future/>

HTTP/3 นั้นออกมาเพื่อแก้ปัญหาของวิธีส่งข้อมูล ซึ่งมีอยู่สองแบบ นั่นก็คือ TCP และ UDP โดยเจ้า TCP เวลาส่งข้อมูล จะมีการตรวจสอบเสมอว่าปลายทางได้รับข้อมูลหรือไม่ ซึ่งทำให้เกิดความล่าช้า ส่วน UDP มันส่งข้อมูลแบบขอไปที ส่งแล้วส่งเลย ไม่สนใจว่าจะเป็นตายเป็นอย่างไร แต่มันเร็วมาก

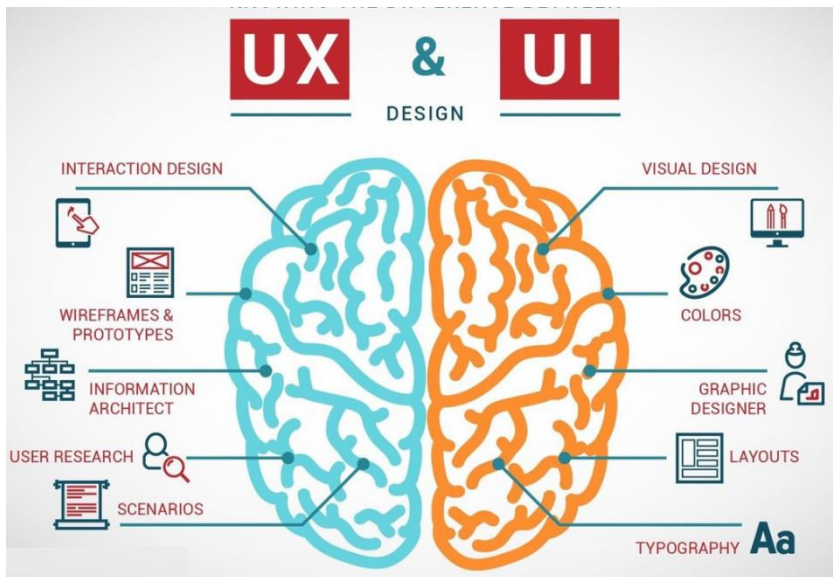
ซึ่งในเว็บปัจจุบันมันมีข้อมูลที่มากขึ้นทั้งขนาด และจำนวน ซึ่งเรายังใช้ TCP อยู่ใน HTTP/2.0 เหมือนกับที่บอกไปข้างต้น มันช้ากว่ามากครับ จึงมีการคิดรูปแบบ Protocol ใหม่ที่ใช้ UDP ส่งข้อมูลแทน ซึ่งทั้งหมดนี้มันเป็นเรื่องของอนาคตอันใกล้นี้ครับ หากน้องบวกเลขตามที่พี่บอกไปก่อนหน้านี้จริง ๆ น้องจะรู้ว่า HTTP/2.0 พี่เริ่มใช้เมื่อปี 2015 นี้เอง HTTP/3 ตอนนี้อยู่ยังไม่มีการใช้งานอย่างเป็นทางการเป็นจริงเป็นจังเพราะมันเพิ่งถูกเสนอเข้าเป็นมาตรฐานใหม่ แต่ว่าในหลาย ๆ แพลตฟอร์มก็เริ่มรองรับการใช้งานแล้วเช่น Cloudflare, Nginx ซึ่งจริง ๆ หลังจากที่พี่เขียนชีทนี้เสร็จ อาจจะมีอัปเดตจาก Google Chrome หรือ Firefox ออกมารองรับแล้วก็ได้ครับ รอตัวพี่ในอนาคตพูดในท้องแล้วกัน

ฝากด้วยนะ ตัวฉันอีกคนหนึ่ง!

ลิงก์อ่านเรื่องนี้เพิ่มเติม

- <https://www.blognone.com/node/112422>
- <https://medium.com/@DarkDragOnite/http-2-คืออะไร-331b3e7b5866>
- <https://blog.nextzy.me/http-d3254cf1dcdd>
- <https://blog.cloudflare.com/http3-the-past-present-and-future/> (ภาษาอังกฤษ)

UI และ UX คืออะไร?



ที่มา <https://uxdesign.cc/why-ux-and-ui-should-remain-separate-7d6e3adbd46f>

UI Design และ UX Design เป็นสองคำที่ผู้พัฒนาอาจจะสับสนอยู่บ่อย ๆ หรืออาจจะไม่มีความหมายจริง ๆ ของมัน โดยทั้งสองก็ต่างเป็นเงื่อนไขหลักและเป็นสิ่งสำคัญที่ควรนึกถึงในการออกแบบเว็บไซต์หรือแอปพลิเคชันในแต่ละครั้งอยู่แล้ว และมักจะเป็นสิ่งที่อยู่คู่กันในการออกแบบเว็บไซต์เสมอ

และหากจะอธิบายให้เข้าใจอย่างง่าย ๆ UI หรือ User Interface นั้นก็คือ ชุดขององค์ประกอบบนหน้าเว็บต่าง ๆ ที่เราสามารถมองเห็นหรือมีส่วนร่วมได้ เช่น รูปภาพ, ปุ่ม และไอคอนต่าง ๆ รวมถึงการวางตำแหน่งของวัตถุนบนหน้าเว็บ เป็นต้น เปรียบเสมือนกับการจัดวางสิ่งของต่าง ๆ ในร้านค้าไว้ตามตำแหน่งที่ต้องการเพื่อให้ลูกค้าหรือ User ได้มองเห็นนั่นเอง

ในส่วนของ UX หรือ User experience นั้น คือประสบการณ์ของผู้ใช้งานเมื่อได้เห็นหรือมีส่วนร่วมกับหน้าเว็บไซต์นั้น ที่ควรจะได้รับประสบการณ์ที่ดีกลับไป ซึ่งหากมีการดีไซน์ UI หรือจัดวางองค์ประกอบบนหน้าเว็บที่ดี ก็จะทำให้ผู้ใช้งานได้รับ UX ที่ดีกลับไปเช่นเดียวกัน เหมือนกับการที่เราเข้าไปในร้านค้าและเห็นการจัดการร้านค้าอย่างเป็นระเบียบ ทำให้ค้นหาสินค้าได้ง่ายขึ้น เป็นต้น

สรุปโดยรวมนั้น UX ก็คือการมุ่งเน้นไปที่การจัดการการใช้งานของผู้ใช้ เพื่อแก้ปัญหาและตอบสนองสิ่งต่าง ๆ ตามที่ผู้ใช้งานต้องการ โดยให้มีปัญหาน้อยที่สุด ส่วน UI นั้นจะมุ่งเน้นไปที่หน้าตาและการทำงานของเว็บเพจเป็นหลักนั่นเอง

