



การพัฒนาโปรแกรมเพื่อแก้ปัญหา

วงจรการพัฒนาโปรแกรม

ความรู้พื้นฐานสำหรับการเขียนโปรแกรมภาษาไพธอน

การเขียนโปรแกรมแบบลำดับ

การเขียนโปรแกรมแบบทางเลือก

การเขียนโปรแกรมแบบวนซ้ำ

ฟังก์ชัน

วงจรการพัฒนาโปรแกรม

(Software Development Life Cycle: SDLC)



วงจรการพัฒนาโปรแกรม (Software Development Life Cycle : SDLC)

การพัฒนาโปรแกรมเป็นแนวทางในการพัฒนาระบบหรือซอฟต์แวร์ ซึ่งมีรูปแบบที่หลากหลาย ขึ้นอยู่กับบริบทของทีมและภาระงาน





วงจรการพัฒนาโปรแกรม (Software Development Life Cycle : SDLC)

1. วิเคราะห์ความต้องการ (Requirement Analysis)

2. ออกแบบ (Design)

3. พัฒนาโปรแกรม (Development)

4. ทดสอบ (Testing)

5. ติดตั้งและบำรุงรักษา (Deployment & Maintenance)

6. การปรับปรุงอย่างต่อเนื่อง (Continuous Improvement)

วงจรการพัฒนาโปรแกรม

ประกอบด้วยอะไรบ้าง ?





ความรู้พื้นฐานสำหรับการเขียนโปรแกรมภาษาไพธอน

1. ภาษาคอมพิวเตอร์

เป็นภาษาที่ใช้กับการทำงานของคอมพิวเตอร์หรือใช้เพื่อให้คอมพิวเตอร์เข้าใจ และสามารถทำงานตามจุดประสงค์ได้ ภาษาคอมพิวเตอร์ แบ่งเป็น 5 ประเภท ได้แก่

1.1 ภาษาเครื่อง

เป็นภาษาคอมพิวเตอร์ยุคแรก ๆ เริ่มใช้ 0 แทน Volt และใช้ 1 แทน แรงดันไฟฟ้า เช่น 5V คำสั่งของเครื่องประกอบด้วยตัวเลขในระบบฐานสอง ได้แก่ 0 และ 1

1.2 ภาษาระดับต่ำ

เป็นภาษาในยุคที่ 2 มีชื่อว่าภาษาแอสเซมบลี ที่เป็นการใช้สัญลักษณ์แทนกลุ่มของเลขฐานสอง เพื่อให้ง่ายต่อการเข้าใจในการเขียนโปรแกรมของมนุษย์มากขึ้น

การทำงานของเครื่องคอมพิวเตอร์จะเข้าใจเฉพาะภาษาเครื่องเท่านั้น ดังนั้นจึงต้องมีการแปลโปรแกรมภาษาแอสเซมบลีให้เป็นภาษาเครื่องโดยกระบวนการแปลภาษาที่เรียกว่า แอสเซมเบลอร์

```
//I*15;  
MOV R3,#15  
STR R3, [R11, #-8]  
  
//J*25;  
MOV R3,#25  
STR R3, [R11, #-12]  
  
//I*|*J;  
LDR R2, [R11, #-8]  
LDR R3, [R11, #-12]  
ADD R3, R2, R3  
STR R3, [R11, #-8]
```

ASSEMBLY LANGUAGE

ASSEMBLER

```
1100 1010 1011 0011  
1100 1010 1011 0011  
1100 1010 1011 0011  
1100 1010 1011 0011  
1100 1010 1011 0011  
1100 1010 1011 0011  
1100 1010 1011 0011
```

MACHINE CODE

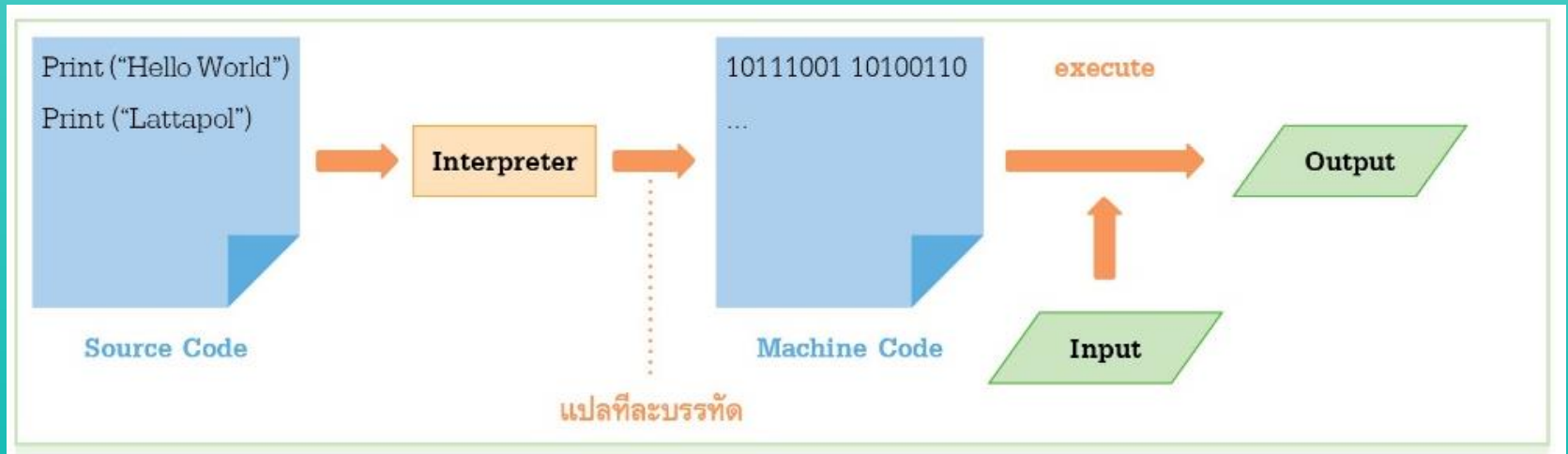
1.3 ภาษาระดับสูง

เป็นภาษารุ่นที่ 3 ที่ถูกสร้างขึ้นมาเพื่อให้สามารถเขียนและอ่านโปรแกรมได้ง่ายขึ้น โดยมีลักษณะเหมือนภาษาอังกฤษ จึงเป็นที่นิยมของโปรแกรมเมอร์ในการเขียนโปรแกรม ตัวอย่างของภาษานี้ เช่น ภาษาโคบอล ภาษาปาสคาล ภาษาเบสิก ภาษาซี และภาษาจาวา

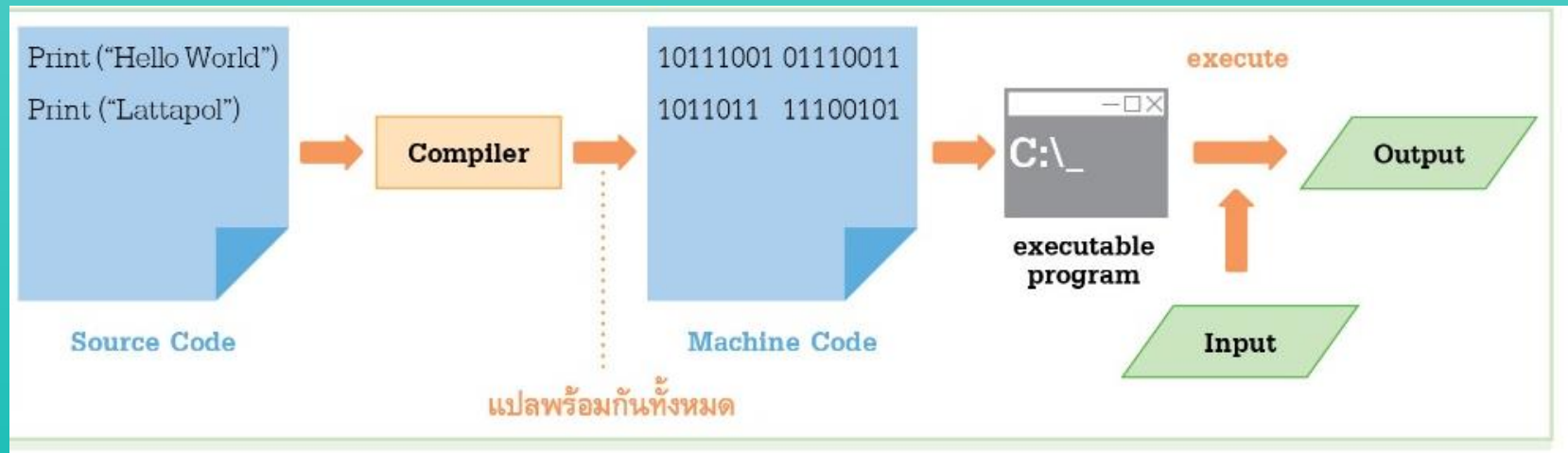


ตัวแปลภาษาระดับสูง

1) อินเทอร์พรีเตอร์ เป็นตัวแปลภาษาระดับสูงให้เป็นภาษาเครื่อง โดยจะแปลทีละบรรทัดและทำงานตามคำสั่งทันที เมื่อมีความผิดพลาดจะรายงานทันที โปรแกรมเมอร์จะต้องแก้ไขคำสั่งบรรทัดนั้นให้ถูกต้อง แล้วจึงสั่งให้โปรแกรมเริ่มทำงานใหม่ ภาษาที่ใช้อินเทอร์พรีเตอร์ เช่น ภาษาเบสิก ภาษาไพธอน

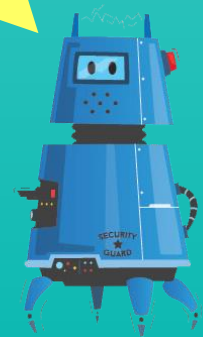


2) คอมไพเลอร์ เป็นโปรแกรมที่ใช้แปลภาษาระดับสูงให้เป็นภาษาเครื่อง และเก็บข้อมูลที่แปลได้ในรูปแบบภาษาเครื่อง ถ้ามีข้อผิดพลาดคอมไพเลอร์ จะบอกข้อผิดพลาดทั้งหมดที่มีในโปรแกรมออกมา ภาษาที่ใช้คอมไพเลอร์ เช่น ภาษาฟอร์แทรน ภาษาโคบอล ภาษาซี ภาษาจาวา



ตัวแปลภาษาแบบอินเทอร์พรีเตอร์กับ

คอมไพเลอร์แตกต่างกันอย่างไร ?

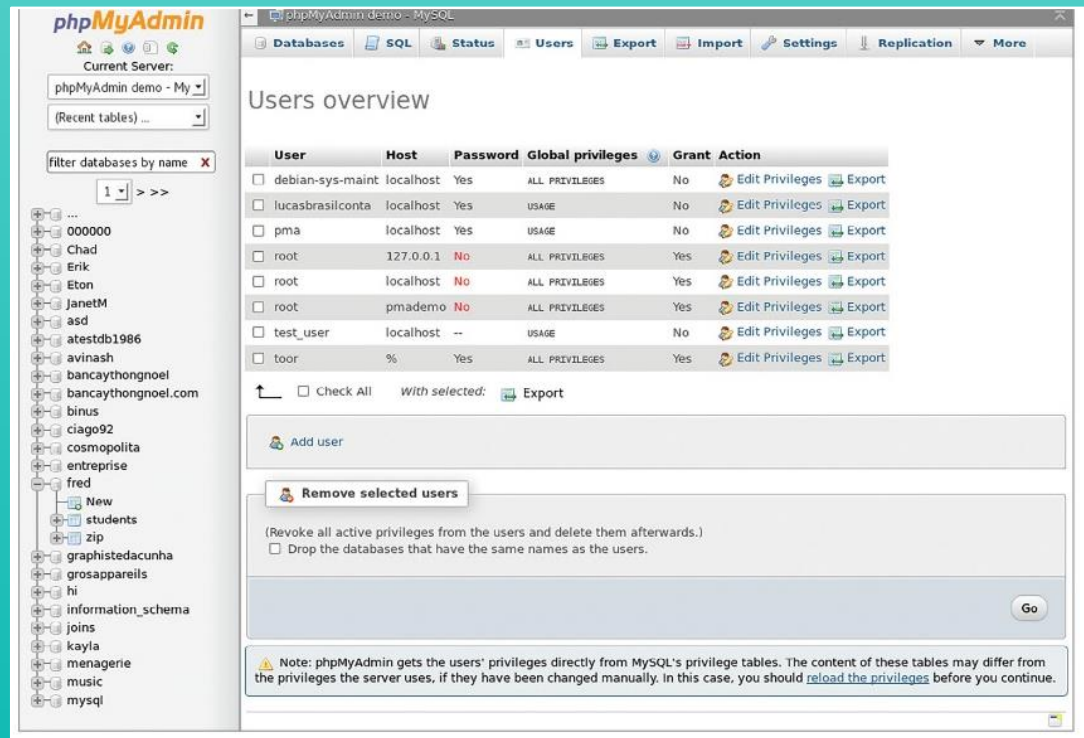


1.4 ภาษาระดับสูงมาก

เป็นภาษาโปรแกรมยุคที่ 4 มีลักษณะคล้ายภาษาพูดของมนุษย์จัดเป็นภาษาที่ไร้
กระบวนการคำสั่ง ผู้ใช้เพียงบอกว่าให้คอมพิวเตอร์ทำอะไร ใกล้เคียงกับภาษาของมนุษย์
ภาษาโปรแกรมที่เขียนได้แก่ ภาษาเอสคิวแอล (SQL)

```
SELECT *  
FROM Teacher  
WHERE Name = "Lattapol"
```

ตัวอย่างการใช้ภาษา SQL
ในการเข้าถึงข้อมูล

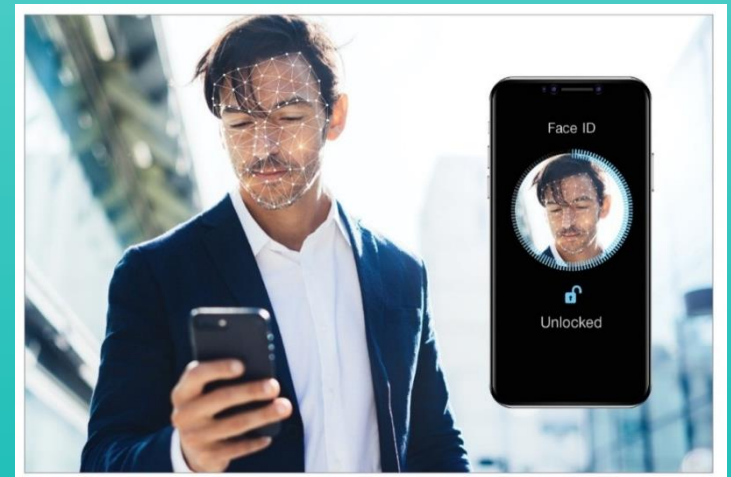


1.5 ภาษาธรรมชาติ

เป็นภาษาโปรแกรมยุคที่ 5 เป็นภาษาแบบ Non-procedural สามารถสั่งงานคอมพิวเตอร์ได้โดยใช้ภาษามนุษย์โดยตรง อาจมีรูปแบบที่ไม่แน่นอนตายตัว ถูกสร้างขึ้นจากเทคโนโลยีทางด้านระบบผู้เชี่ยวชาญปัญญาประดิษฐ์ ปัจจุบันมีการนำมาใช้เฉพาะด้าน เช่น การแพทย์ การพยากรณ์อากาศ การลงทุน



เว็บไซต์ Youtube ใช้ Machine Learning

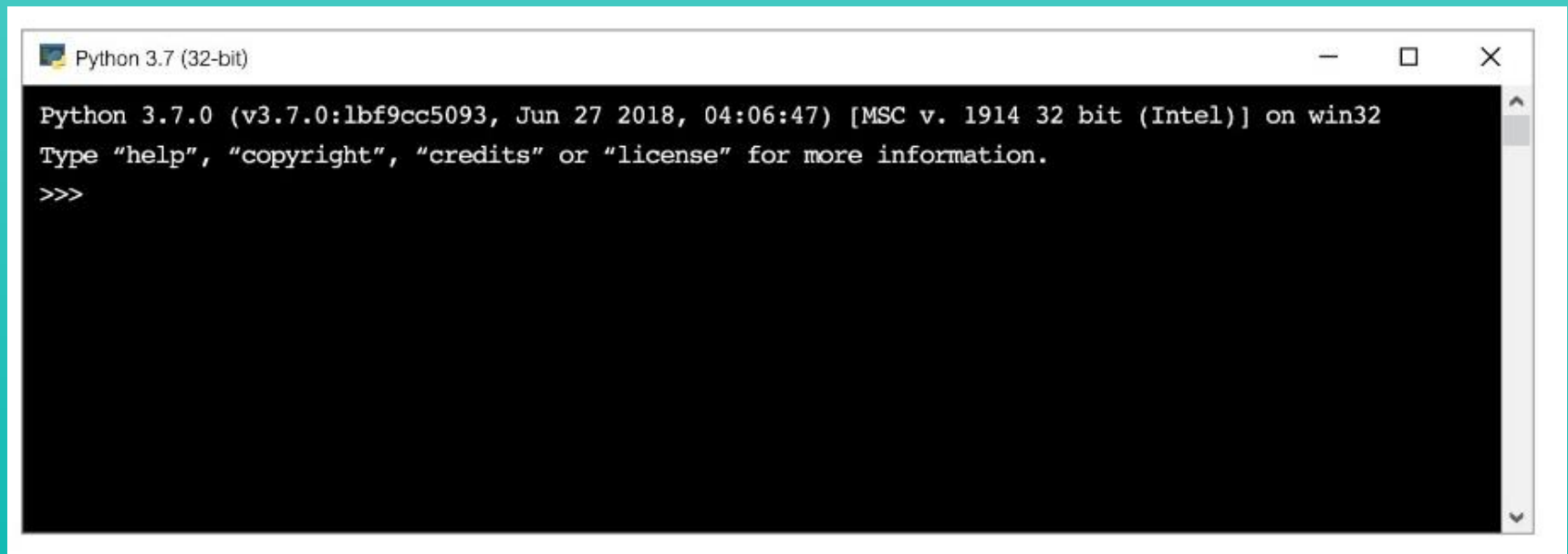


Face id

2. ภาษาไพธอน



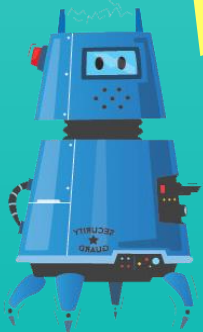
เป็นภาษาระดับสูงที่เหมาะสมกับการฝึกเขียนโปรแกรมสำหรับผู้เริ่มต้นฝึกเขียนโปรแกรม โดยมีโครงสร้างภาษาที่ไม่ซับซ้อน สามารถต่อยอดเพื่อพัฒนาชิ้นงานอื่น ๆ ได้อีกมากมาย เช่น การสร้างเกม การสร้างเว็บแอปพลิเคชัน

A screenshot of a Windows command prompt window titled "Python 3.7 (32-bit)". The window has standard Windows window controls (minimize, maximize, close) in the top right corner. The text inside the window is as follows:

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v. 1914 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

โปรแกรมภาษาไพธอนมีข้อดีอย่างไร

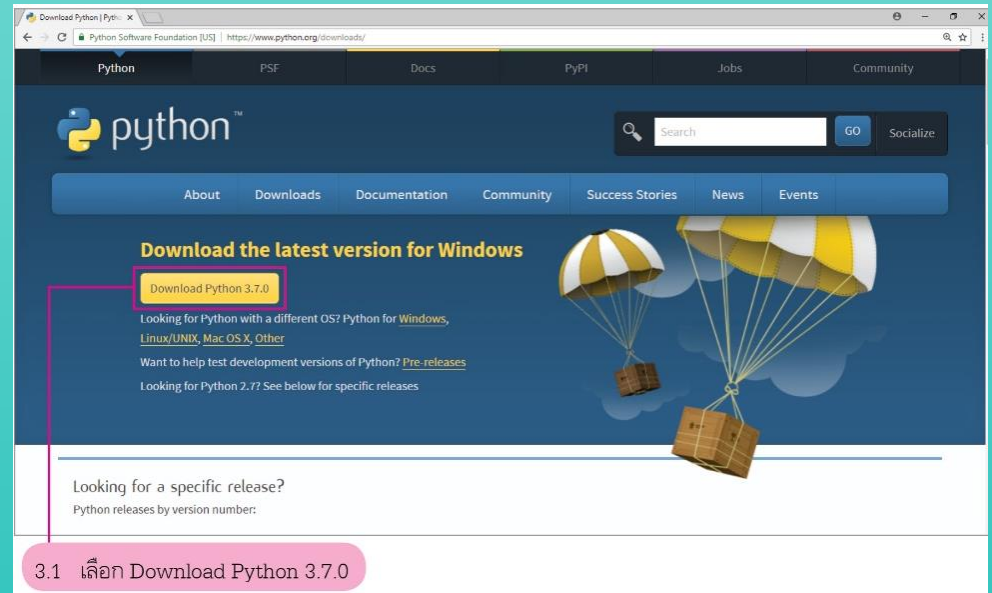
สำหรับผู้เริ่มฝึกเขียนโปรแกรม ?



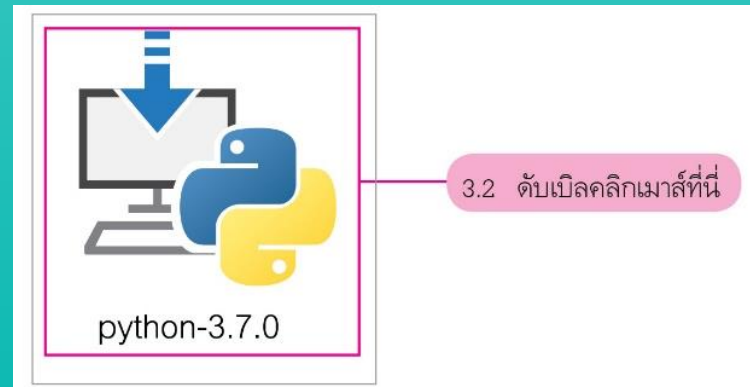
3. การติดตั้งอินเทอร์พรีเตอร์ภาษาไพธอน

3.1 ไปที่หน้าเว็บไซต์

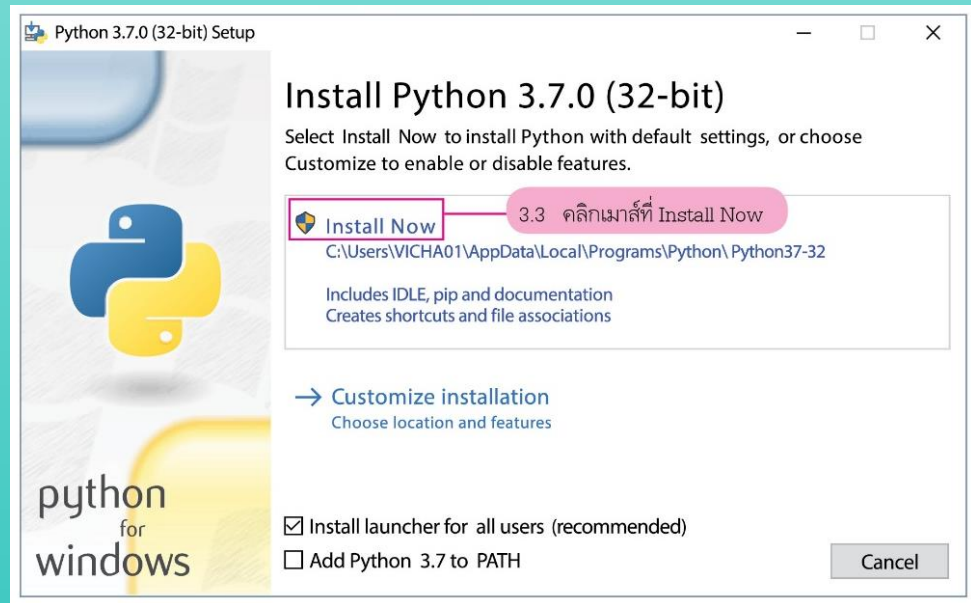
<https://www.python.org/downloads/>
เพื่อดาวน์โหลดตัวติดตั้ง



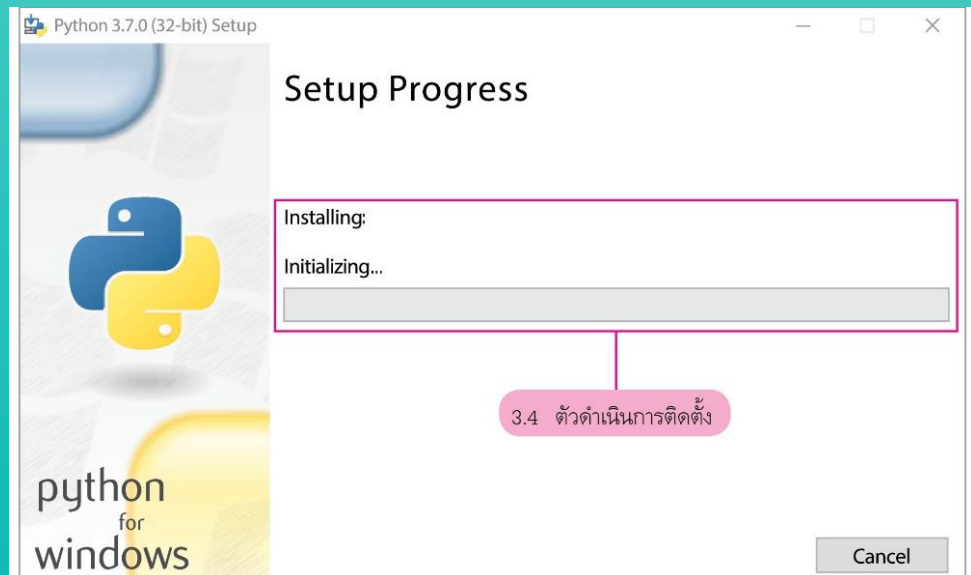
3.2 ดับเบิลคลิกเมาส์ที่ตัวติดตั้ง อินเทอร์พรีเตอร์ python 3.7.0



3.3 คลิกเมาส์ที่คำสั่ง Install Now



3.4 รอดำเนินการติดตั้ง

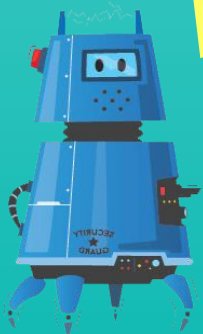


3.5 ติดตั้งสำเร็จ คลิกเมาส์ที่ปุ่ม close



อินเทอร์เน็ต

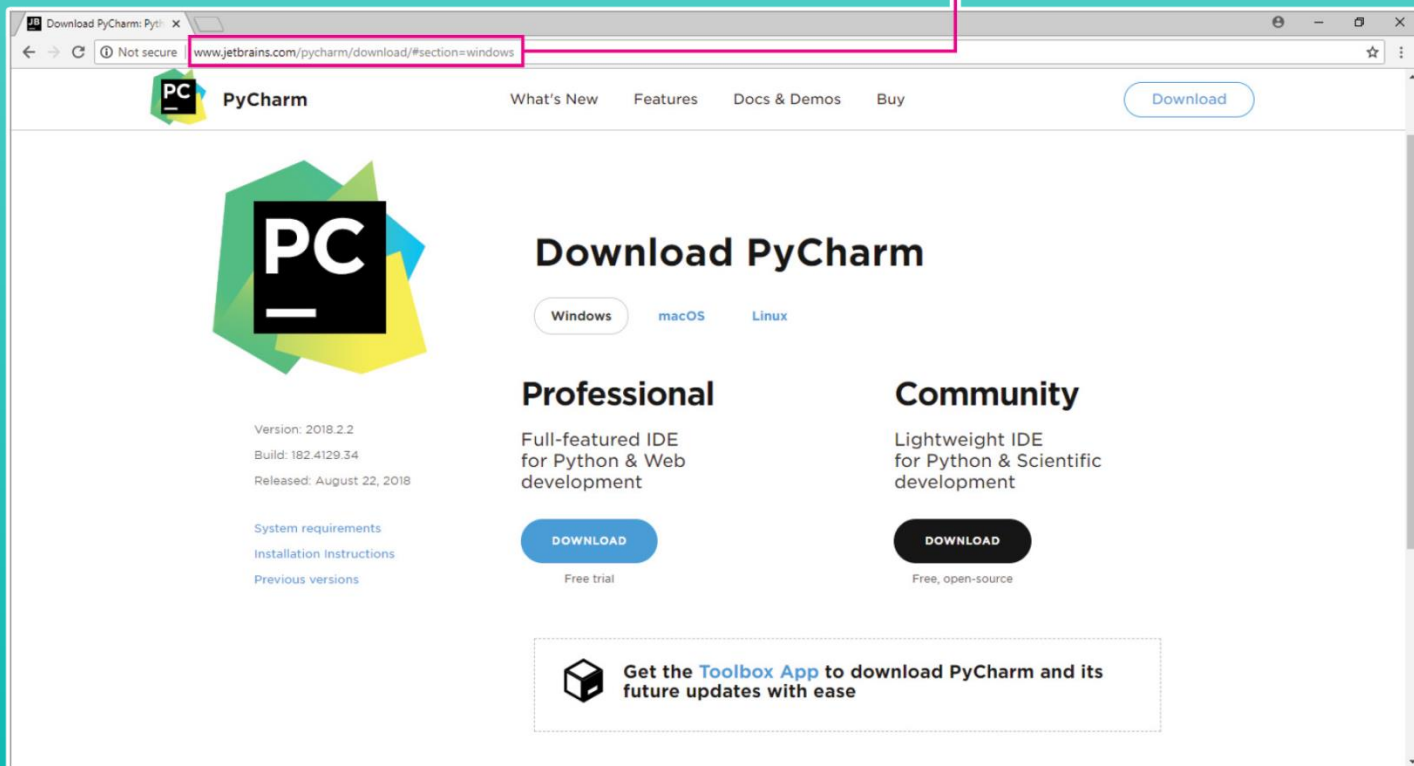
ติดตั้งเพื่อจุดประสงค์ใด ?



4. การติดตั้งโปรแกรม PyCharm สำหรับใช้เป็น IDE ภาษาไพธอน

4.1 ไปที่หน้าเว็บไซต์ <https://www.jetbrains.com/pycharm/download/> เพื่อดาวน์โหลดตัวติดตั้งโปรแกรม PyCharm

4.1 ลิงก์ URL ตัวติดตั้งโปรแกรม PyCharm



4.2 คลิกเมาส์ที่ปุ่ม Download



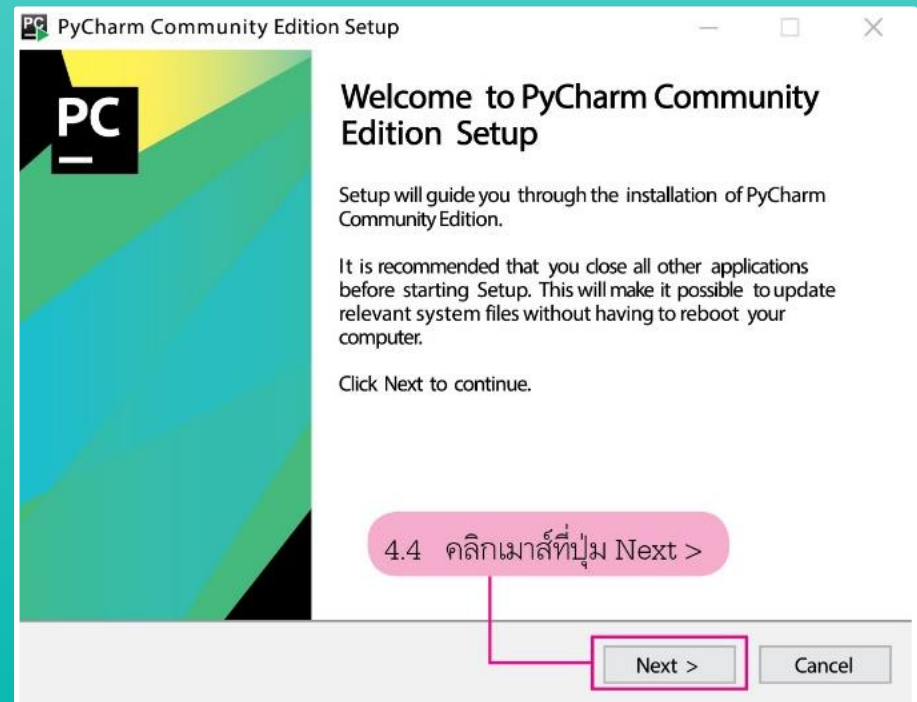
4.2 เลือก DOWNLOAD

4.3 ดับเบิลคลิกเมาส์ที่ตัวติดตั้ง

4.3 ตัวติดตั้ง pycharm-community-2018.2.2

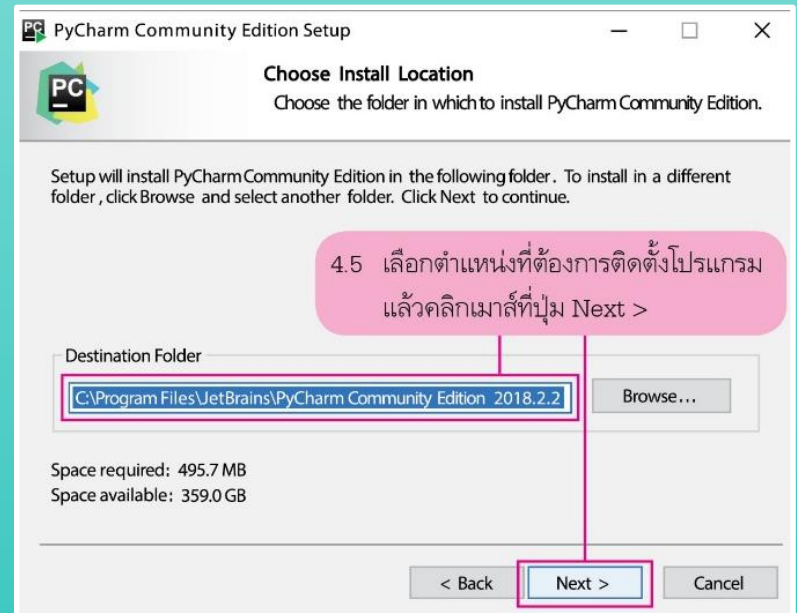


4.4 โปรแกรมจะแสดงหน้าจอ
ต้อนรับดังภาพ จากนั้นคลิกเมาส์
ที่ปุ่ม Next

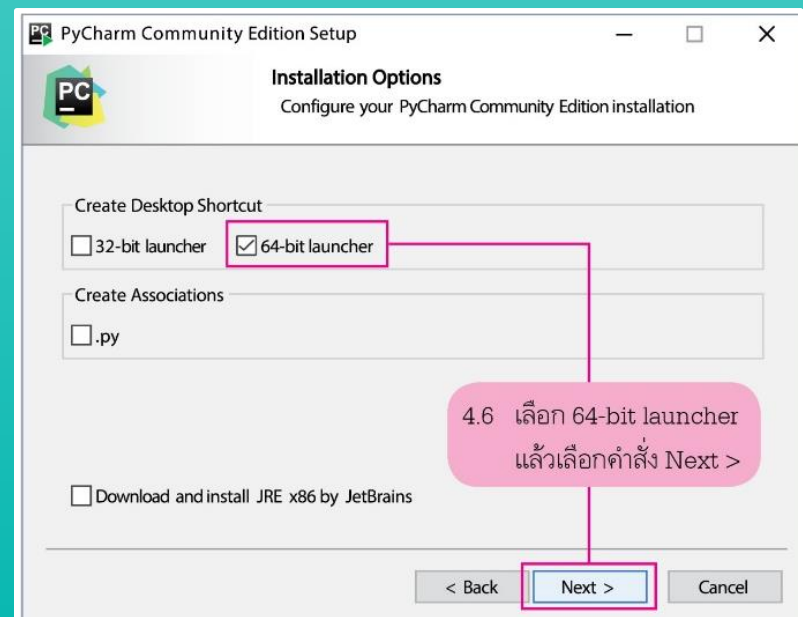


4.4 คลิกเมาส์ที่ปุ่ม Next >

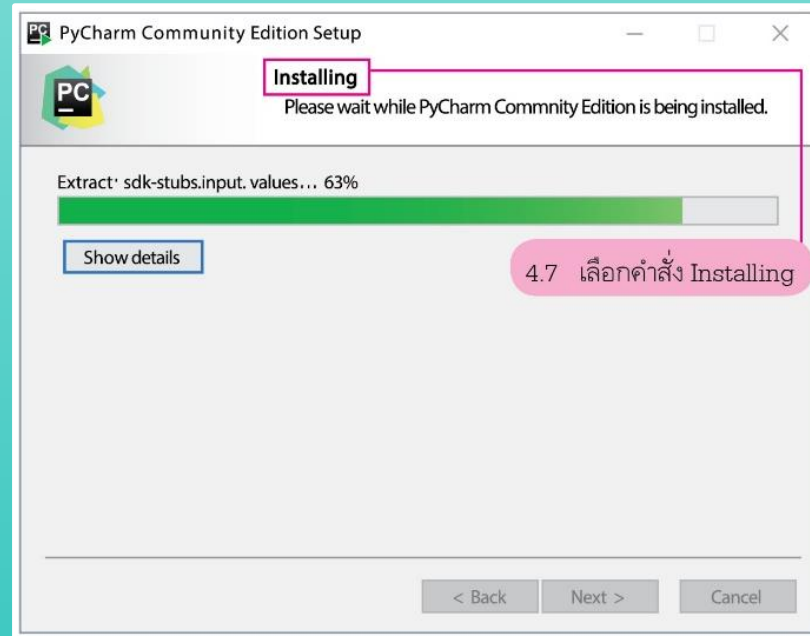
4.5 โปรแกรมสอบถามตำแหน่งติดตั้ง
ของโปรแกรกดังภาพ จากนั้นเลือก
ตำแหน่งที่ต้องการติดตั้ง แล้วคลิกเมาส์
ที่ปุ่ม Next



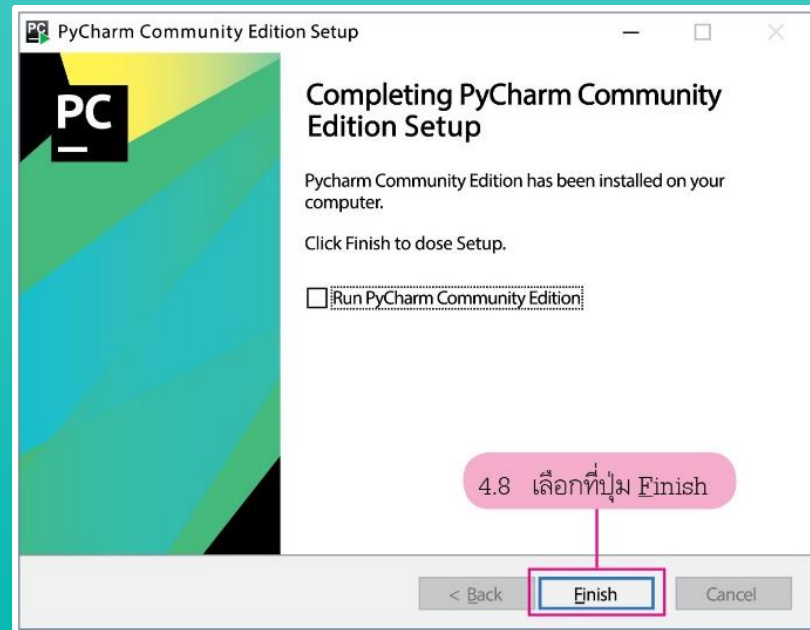
4.6 โปรแกรมจะสอบถามการสร้าง
icon บนหน้าจอ จากนั้นทำเครื่องหมาย
✓ ลงในช่องว่างหน้า 64-bit แล้วคลิก
เมาส์ที่ปุ่ม Next



4.7 คลิกเมาส์ที่ปุ่ม Install
เพื่อเริ่มการติดตั้ง

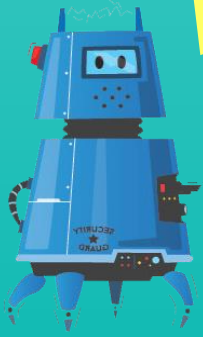


4.8 เมื่อติดตั้งเสร็จแล้ว
คลิกเมาส์ที่ปุ่ม Finish



หากการลงโปรแกรมไม่สมบูรณ์

นักเรียนมีวิธีแก้ไขอย่างไร ?

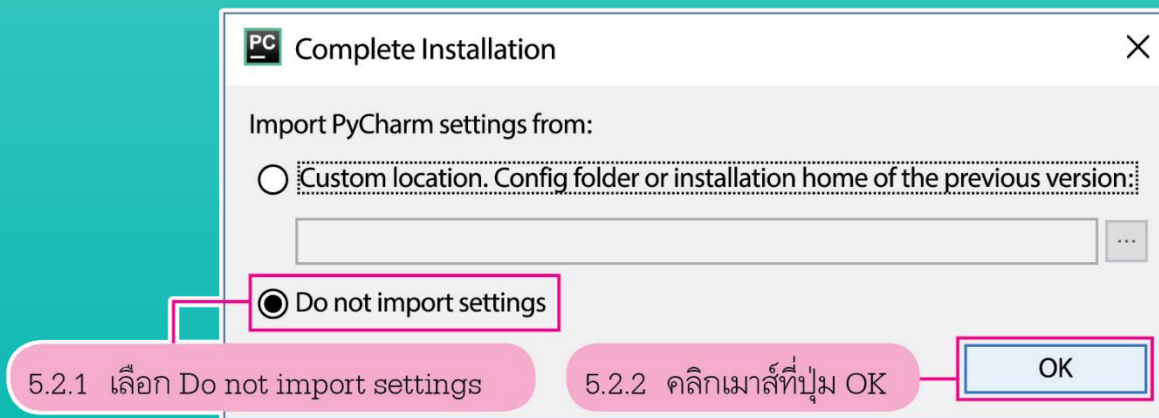


5. การใช้งานโปรแกรม PyCharm เบื้องต้น

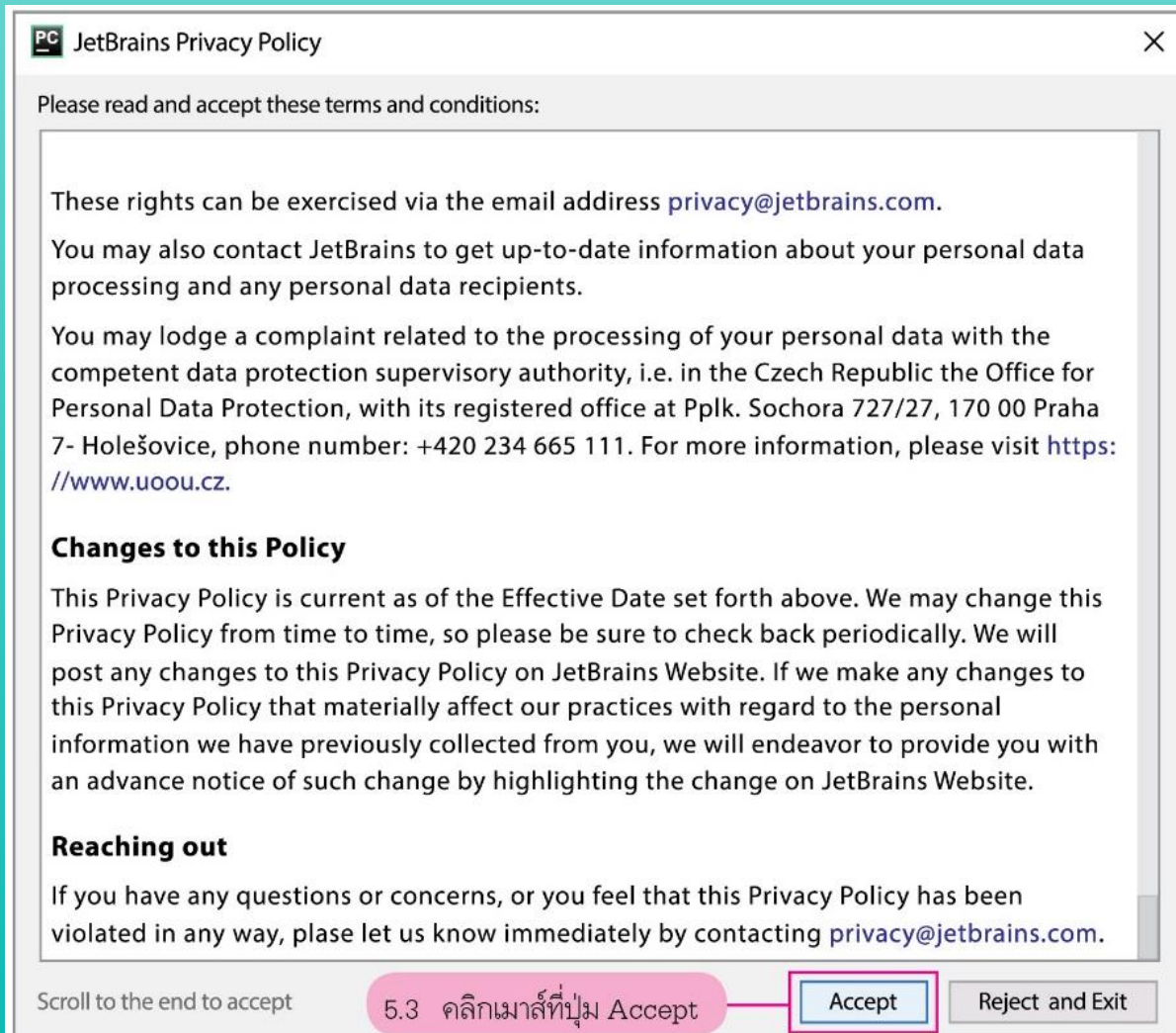
5.1 คลิกเมาส์ที่ icon โปรแกรมบนหน้า Desktop



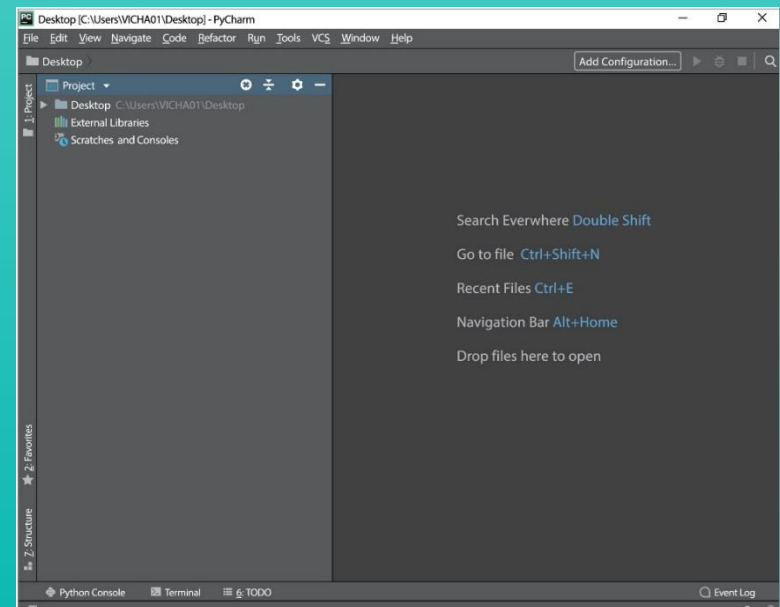
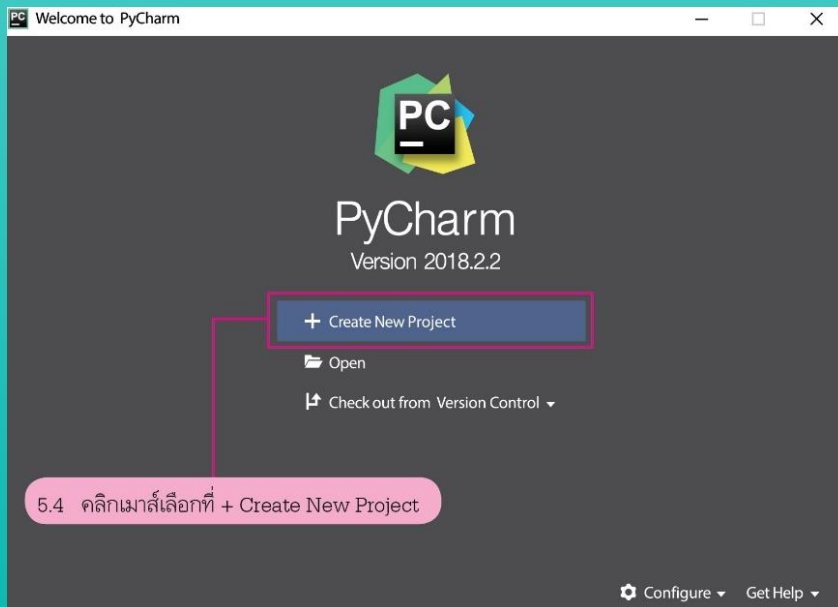
5.2คลิกเมาส์เลือกหน้าคำสั่ง Do not import settings จากนั้นคลิกเมาส์ปุ่ม OK



5.3 อ่านข้อตกลง แล้วคลิกเมาส์ปุ่ม Accept



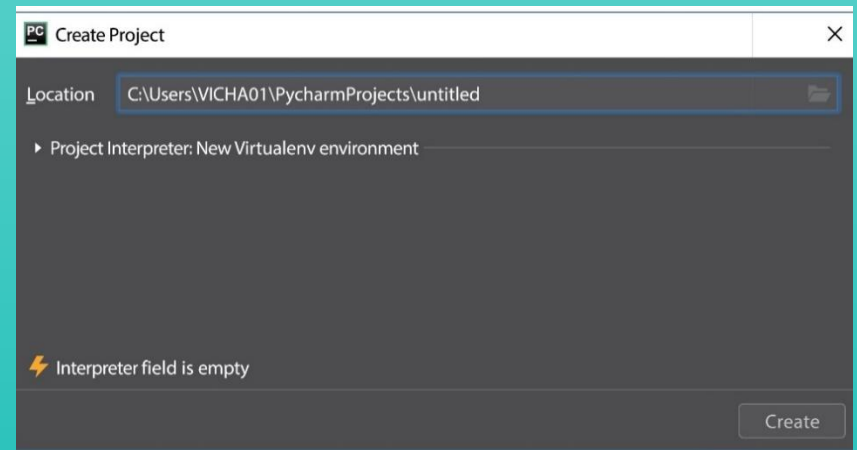
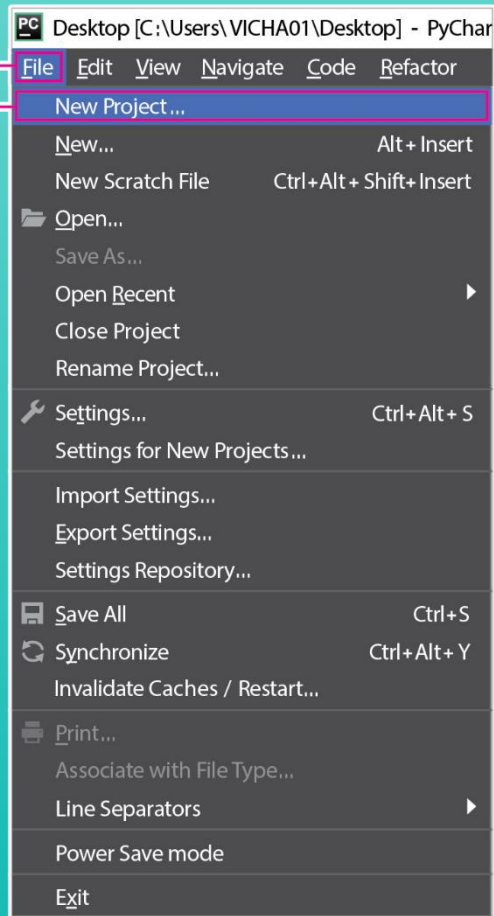
5.4 เริ่มเข้าใช้งานโปรแกรม แล้วคลิกเมาส์ที่ Create New Project



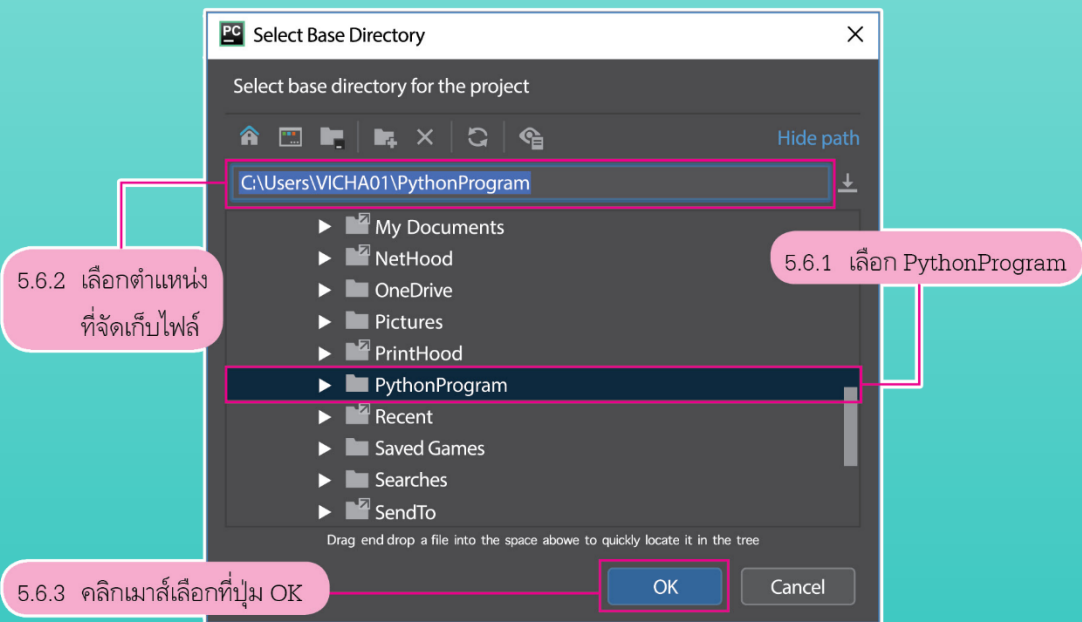
5.5 เริ่มสร้างโปรเจกต์ โดยคลิกเมาส์เลือก File เลือก New Project

5.5.1 เลือกที่เมนู File

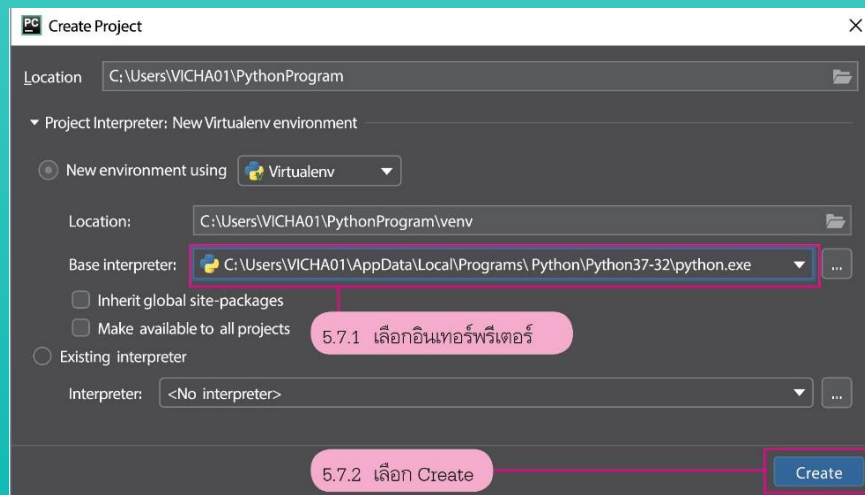
5.5.2 เลือก New Project...



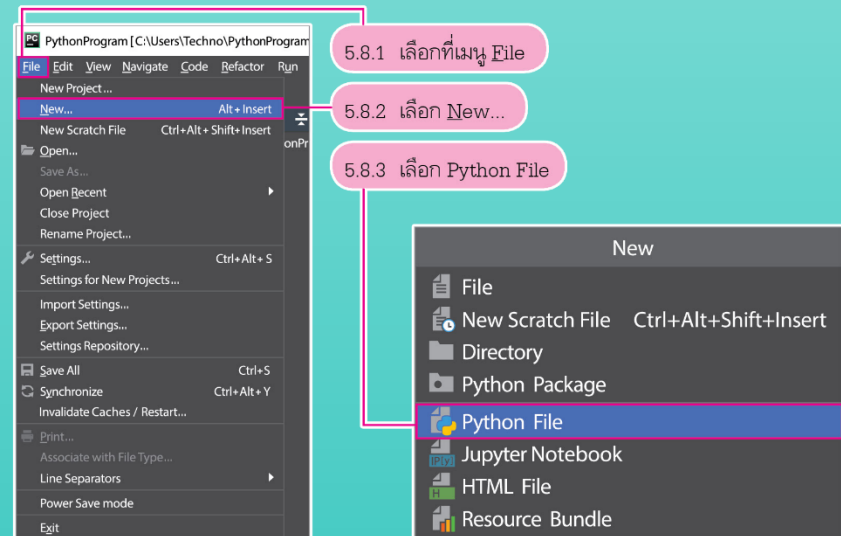
5.6 คลิกเมาส์ที่ปุ่ม โฟลเดอร์
เพื่อกำหนดตำแหน่งที่จะ
บันทึก แล้วคลิกเมาส์ปุ่ม OK



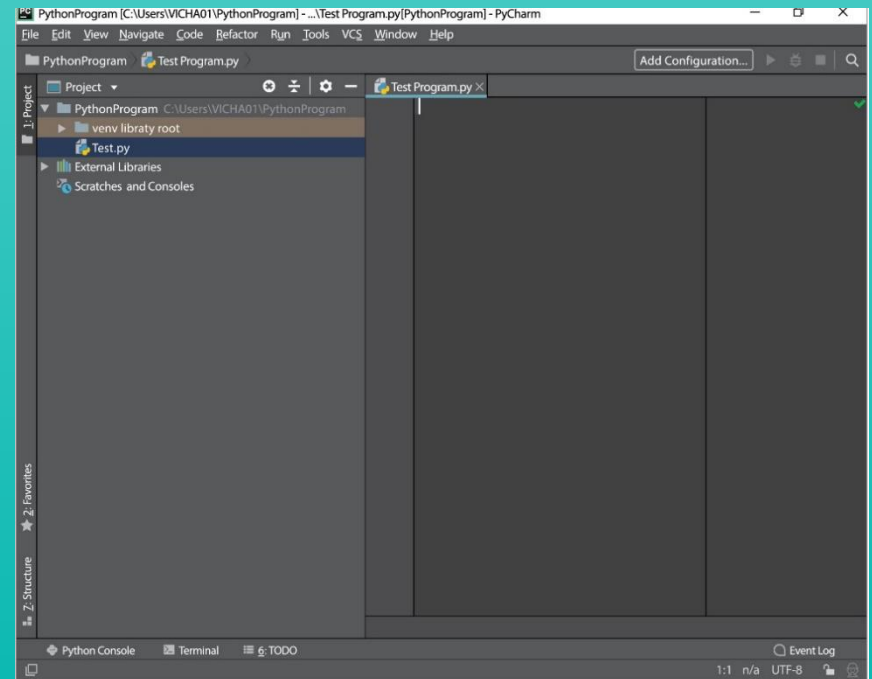
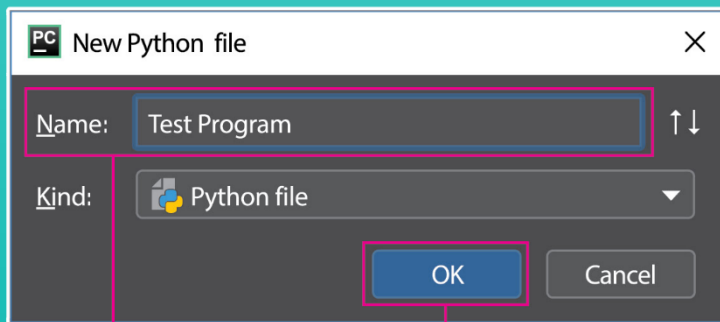
5.7 หากมีข้อความเตือน
Interpreter field is empty
ให้เลือกอินเทอร์พรีเตอร์ดังภาพ
แล้วคลิกเมาส์ปุ่ม Create



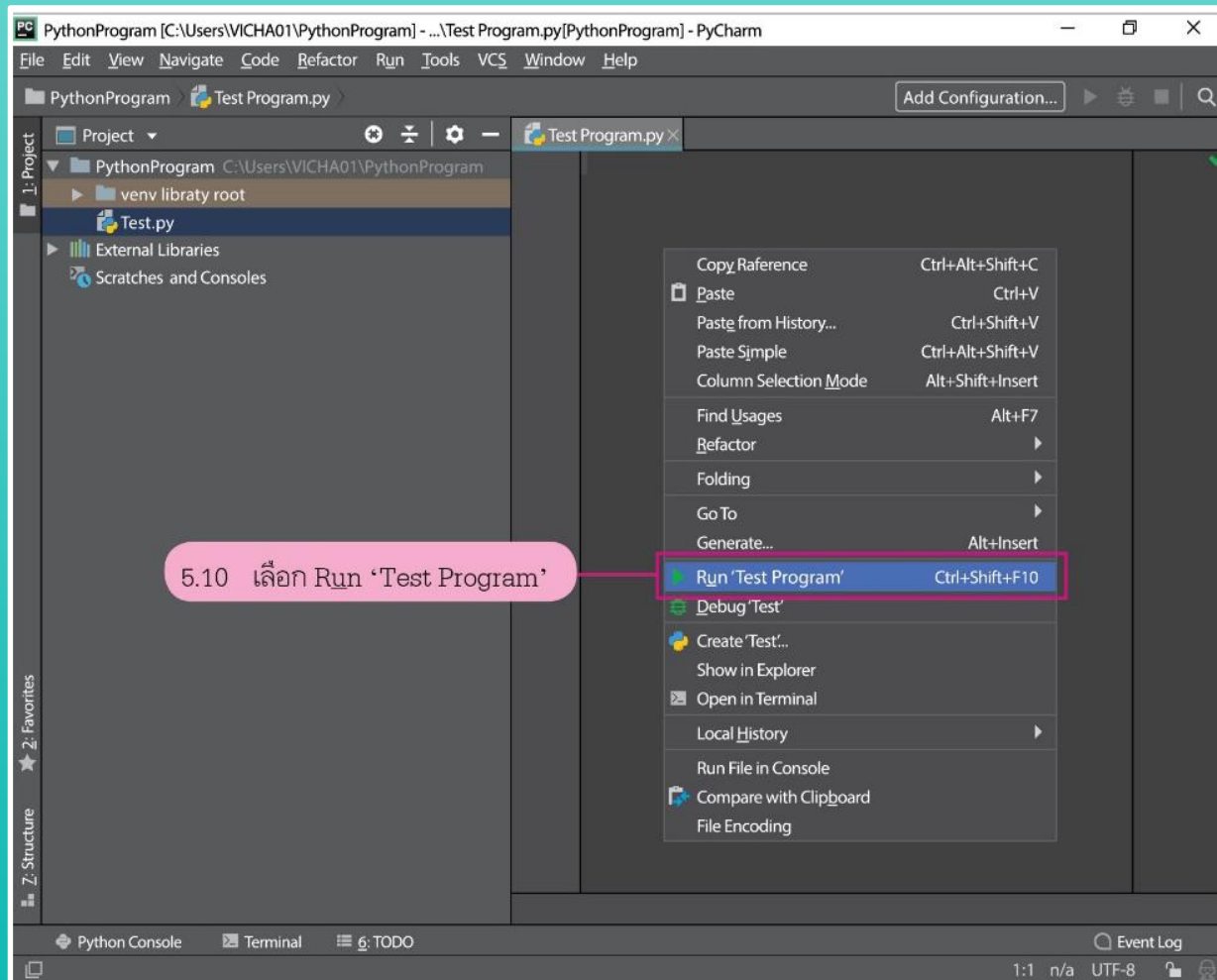
5.8 เริ่มสร้างไฟล์งานคลิกเมาส์ เลือก File เลือก New แล้วเลือก ชนิดไฟล์ Python File



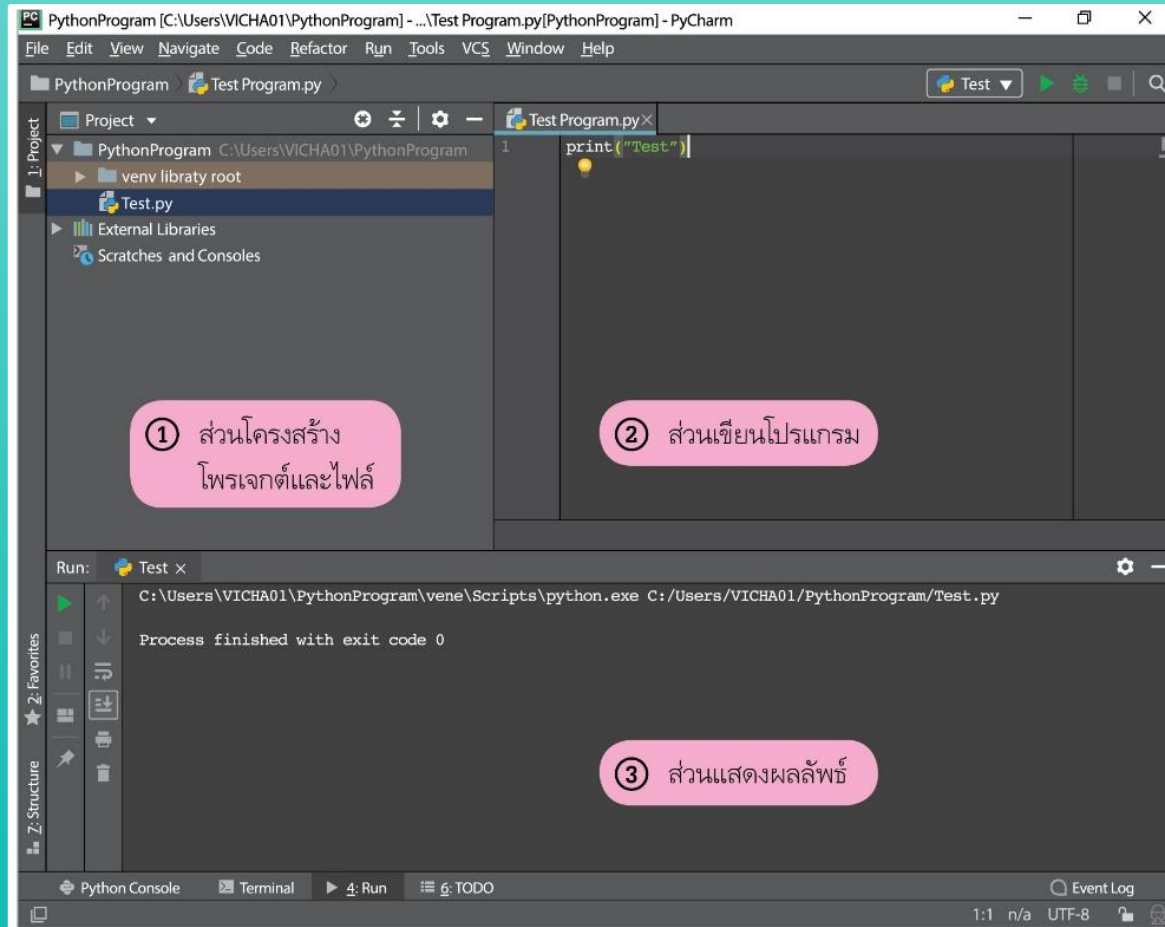
5.9 ตั้งชื่อไฟล์งาน แล้วคลิกเมาส์ปุ่ม OK



5.10 ทดสอบการเขียนโปรแกรมด้วยคำสั่ง print (“test”) จากนั้นเลือกเมนู Run ‘test’



5.11 โปรแกรมจะแสดงผลลัพธ์ของโปรแกรม ดังภาพ



ถ้านักเรียนทดสอบโปรแกรมแล้ว
เกิดปัญหาโปรแกรม
ไม่แสดงผลข้อมูลที่นักเรียนเขียน
จะมีวิธีการตรวจสอบอย่างไร?



6. การเขียนคำอธิบาย

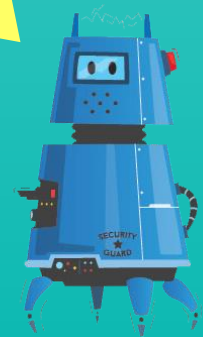
การเขียนคอมเมนต์ใช้เมื่อต้องการอธิบายการทำงานของโปรแกรม จะใช้เครื่องหมายชาร์ป (#) นำหน้าข้อความ ดังตัวอย่าง

	โปรแกรม	ผลลัพธ์
1	<code>x = 10</code>	
2	<code>#x = x + 10</code>	
3	<code>x = x - 10</code>	
4	<code>print(x)</code>	0

หรือหากต้องการเขียนคอมเมนต์พร้อมกันหลายบรรทัด สามารถใช้เครื่องหมาย Double Quote 3 คู่ ครอบคลุมที่ต้องการคอมเมนต์ ดังตัวอย่าง

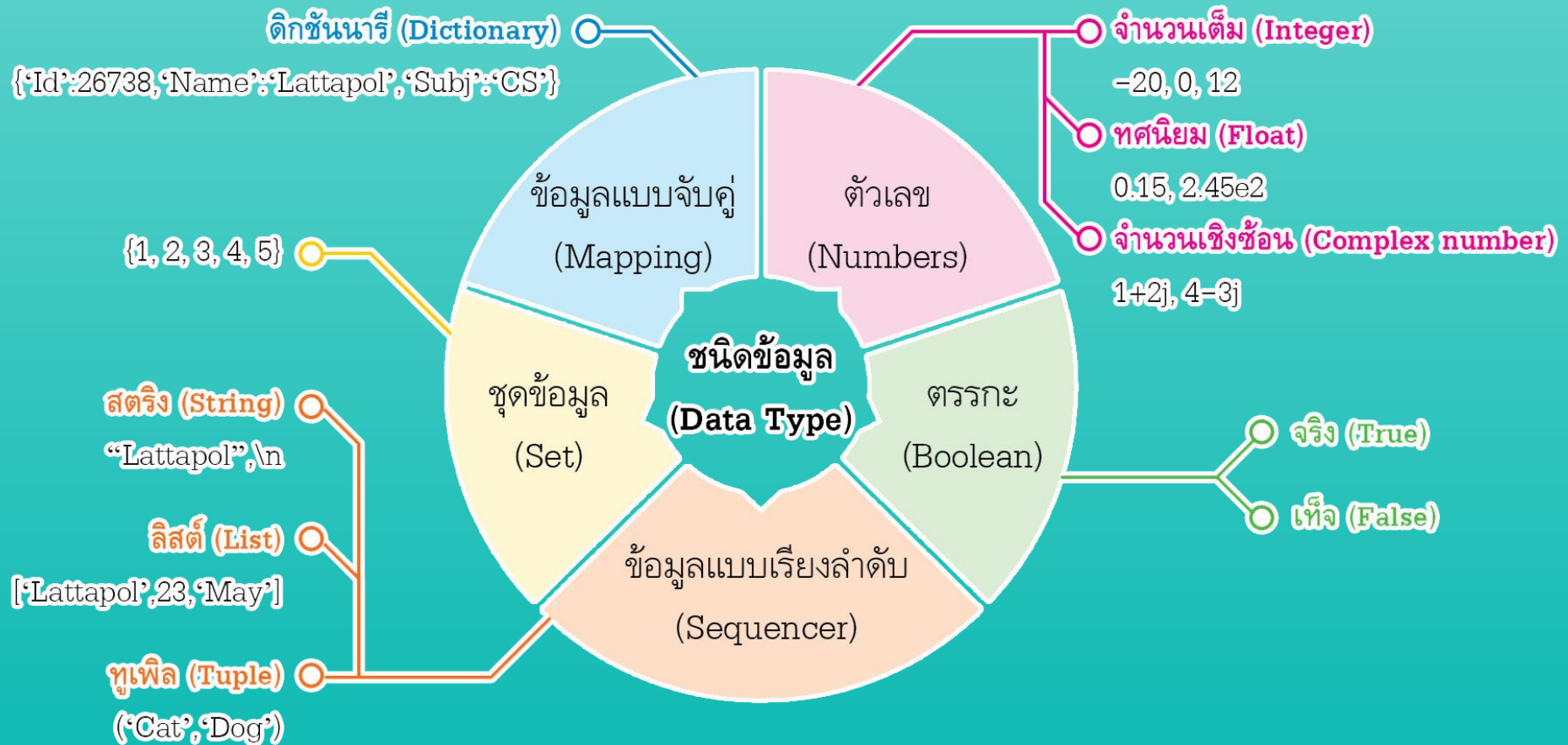
	โปรแกรม	ผลลัพธ์
1	<code>x = 10</code>	
2	<code>"""x = x + 10</code>	
3	<code>x = x - 10"""</code>	
4	<code>print(x)</code>	10

ทำไมจึงต้องเขียนคำอธิบาย?



7. ชนิดข้อมูล (Data type)

ข้อมูลจะถูกจัดเก็บในหน่วยความจำ ซึ่งจะมีวิธีเก็บข้อมูลแตกต่างกันไป
ชนิดข้อมูลทั้งหมดในภาษาไพธอนประกอบด้วย



7.1 จำนวนเต็ม (Integer) มีขนาด 8 Bytes (64bits) สามารถกำหนดเป็นเลขฐานต่าง ๆ ได้ โดยการใส่เลข 0 นำหน้าแล้วตามด้วยอักษรประจำฐาน ได้แก่

เลขฐานสอง	ใช้เลขโดด 0 และ 1	ตัวอักษรประจำฐาน b
เลขฐานแปด	ใช้เลขโดด 0-7	ตัวอักษรประจำฐาน o
เลขฐานสิบหก	ใช้เลขโดด 0-9 และ A-F	ตัวอักษรประจำฐาน x
เลขฐานสิบ	ให้พิมพ์เป็นตัวเลขปกติ ดังตัวอย่าง	

	โปรแกรม	ผลลัพธ์
1	<code>print(0b010110)</code>	22
2	<code>print(0o205)</code>	133
3	<code>print(0x92A)</code>	2346
4	<code>print(501)</code>	501

7.2 จำนวนทศนิยม (Floating-Point Number) จำนวนทศนิยม เช่น 12.45

นอกจากนี้ยังสามารถกำหนดทศนิยมในรูปสัญกรณ์วิทยาศาสตร์ ดังตัวอย่าง

	โปรแกรม	ผลลัพธ์
1	num = 2018	#เป็นจำนวนเต็ม
2	print(num)	2018
3	num = 2018.0	#เป็นทศนิยม
4	print(num)	2018.0
5	num = 2018e5	#เป็น 2018×10^5
6	print(num)	201800000.0
7	num = 2018e-5	#เป็น 2018×10^{-5}
8	print(num)	0.02018

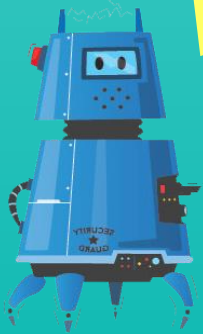
7.3 ตรรกะ (Boolean) เป็นข้อมูลที่ประกอบด้วย 2 ค่า คือ

- จริง (True) หรือตัวย่อ T
- เท็จ (False) หรือตัวย่อ F



ทำไมการเขียนโปรแกรม

จึงต้องมีตัวแปร ?



8. ตัวแปร (Variable)

ตัวแปร คือ การจองพื้นที่ในหน่วยความจำเพื่อใช้ในการเก็บข้อมูลไว้ใช้ประมวลผลขั้นต่อไป ซึ่งในภาษาไพธอนจะเป็นตัวแปรแบบ Dynamic คือไม่ต้องประกาศตัวแปรและชนิดข้อมูล

8.1 การกำหนดชื่อตัวแปร

กฎการตั้งชื่อตัวแปรในภาษาไพธอน มีดังนี้

- 1) ตั้งชื่อให้สื่อความหมายกับข้อมูลที่จัดเก็บ
- 2) อักษรตัวพิมพ์เล็กและพิมพ์ใหญ่จะมีความหมายต่างกัน
- 3) เริ่มต้นด้วยตัวอักษรภาษาอังกฤษ ภาษาไทย หรือเครื่องหมาย _
- 4) หลังจากตัวแรกจะเป็นตัวอักษรภาษาอังกฤษ ภาษาไทย ตัวเลข หรือเครื่องหมาย _ ห้ามมีเว้นวรรค จุด หรือสัญลักษณ์พิเศษ
- 5) ไม่เป็นคำสงวนในภาษาไพธอน

คำสงวนในภาษาไพธอน

and	exec	not
assert	finally	or
break	for	pass
class	from	print
continue	global	raise
def	if	return
del	import	try
elif	in	while
else	is	with
except	lambda	yield

8.2 การประกาศตัวแปร

เป็นการสร้างตัวแปรสำหรับเก็บข้อมูลก่อนนำไปประมวลผล โดยมีรูปแบบการประกาศตัวแปร ดังนี้

ชื่อตัวแปร = ข้อมูล
ชื่อตัวแปร = “ ”

เครื่องหมาย = เป็นเครื่องหมายที่แทนการกำหนดค่า ซึ่งหมายถึง นำข้อมูลด้านขวาของ = มากำหนดทางด้านซ้าย ดังตัวอย่าง

	โปรแกรม	
1	text = "Python"	#ประกาศตัวแปรชื่อ text เก็บข้อมูลสายอักขระ (string)
2	number = 23	#ประกาศตัวแปรชื่อ number เก็บข้อมูลจำนวนเต็ม (integer)
3	score = 16.25	#ประกาศตัวแปรชื่อ score เก็บข้อมูลทศนิยม (float)
4	blank = ""	#ประกาศตัวแปรชื่อ blank เก็บค่าว่าง

	โปรแกรม	ผลลัพธ์
1	print (5<0)	False
1	print (0<5)	True

8.3 สตริง (String)

เป็นข้อมูลแบบอักขระที่รองรับภาษาต่าง ๆ และสัญลักษณ์อื่น ๆ สตริงมีรูปแบบ ดังนี้

- 1) อยู่ภายในเครื่องหมาย “ ” หรือ ‘ ’
- 2) ต้องการให้มีเครื่องหมาย “ ” ประกอบในข้อความ ต้องปิดข้อความทั้งหมดด้วย ‘ ’
หรือต้องการให้มีเครื่องหมาย ‘ ’ ประกอบในข้อความ ต้องปิดข้อความทั้งหมดด้วย “ ”
- 3) ใส่ \n ในข้อความ เพื่อขึ้นบรรทัดใหม่
- 4) ใช้เครื่องหมาย “ ” หรือ ” ปิดข้อความที่มีหลายบรรทัด

	โปรแกรม	ผลลัพธ์
1	<code>name = 'Lattapol'</code>	
2	<code>lastname = "'Dansakul'"</code>	
3	<code>print(name, lastname)</code>	Lattapol 'Dansakul'
4	<code>address = '60/362 Senaniwet2,</code> <code>\nPraserdmanukid road,\nBKK, 10230'</code>	
5	<code>print(address)</code>	60/362 Senaniwet2, Praserdmanukid road, BKK, 10230
6	<code>address = '''60/362 Senaniwet2,</code> <code>Praserdmanukid road,</code> <code>BKK, 10230'''</code>	
7		
8		
9	<code>print(address)</code>	60/362 Senaniwet2, Praserdmanukid road, BKK, 10230

9. การแสดงผลออกทางหน้าจอด้วยคำสั่ง `print()` รูปแบบการใช้งานดังนี้

`print` (ข้อมูลที่ต้องการแสดงผล)

ข้อมูลที่ต้องการแสดงผล ได้แก่ ตัวแปร ค่าคงที่ นิพจน์ หรือข้อความ ซึ่งถ้าเป็นข้อความ จะต้องอยู่ภายใน “ ” หรือ ‘ ’ ดังตัวอย่าง

	โปรแกรม	ผลลัพธ์
1	<code>print(1)</code>	1
2	<code>print(12+11)</code>	23
3	<code>x = 20</code>	
4	<code>print(x)</code>	20
5	<code>print(x + 5)</code>	25
6	<code>print("Lattapol Dansakul")</code>	Lattapol Dansakul

โดยทั่วไปแล้วคำสั่ง `print()` จะแสดงผลการขึ้นบรรทัดหลังจบคำสั่ง แต่หากต้องการปรับแต่งการจบโดยยังไม่ให้ขึ้นบรรทัดใหม่สามารถใส่คำสั่ง `end = "..."` เพื่อกำหนดรูปแบบของการจบคำสั่ง ดังตัวอย่าง

	โปรแกรม	ผลลัพธ์
1	<code>print(1, end=" ")</code>	1 2
2	<code>print(2)</code>	3
3	<code>print(3)</code>	




เด็กควรรู้

สัญลักษณ์การจัดรูปแบบข้อมูลสตริงที่ควรทราบ ได้แก่

- `%c` จัดรูปแบบตัวอักขระ (Character)
- `%d` จัดรูปแบบตัวเลขจำนวนเต็มฐานสิบ (Decimal Number)
- `%f` จัดรูปแบบตัวเลขทศนิยม (Floating Number)
- `%s` จัดรูปแบบสายอักขระ (String)

โครงสร้างการจัดรูปแบบการแสดงผลจะมีการอ้างอิงตำแหน่ง ซึ่งค่าที่อยู่ในอาร์กิวเมนต์จะต้องสัมพันธ์กับสัญลักษณ์ที่อยู่ในส่วนของการจัดรูปแบบแสดงผลจริง และมีเครื่องหมาย % คั่น



```
print("ข้อความ %s, %d, %f" % (var1, var2, var2))
```

	โปรแกรม	ผลลัพธ์
1	num1 = 12; num2 = 15.2; text = "Python"; ch = "A"	
2	print("first is %d" % num1)	first is 12
3	print("second is %f" % num2)	second is 15.2
4	print("third is %s" % text)	third is Python
5	print("fourth is %c" % ch)	fourth is A

10. การรับข้อมูลเข้าด้วยคำสั่ง input()

```
ตัวแปร = input ("ข้อความที่แสดงก่อนรับข้อมูลเข้า")
```

ตัวแปร : เป็นตัวแปรที่ใช้เก็บข้อมูลที่รับเข้ามาทางคีย์บอร์ด

ข้อความที่แสดงก่อนรับข้อมูลเข้า : คำอธิบายที่ต้องการแสดงเพื่อชี้แจงกับผู้ใช้งาน
เช่น

	โปรแกรม
1	<pre>name = input ("Enter your name:")</pre>

เมื่อมีการเรียกใช้คำสั่ง `output()` โปรแกรมแสดงข้อความชี้แจง

ผลลัพธ์
Enter your Var_Name:

แล้วรอการป้อนข้อมูลนำเข้าจากผู้ใช้ผ่านคีย์บอร์ด แล้วตามด้วยปุ่ม Enter เช่น

ผลลัพธ์
Enter your name: Lattapol

ซึ่งโปรแกรมจะนำข้อมูล Lattapol ไปเก็บในตัวแปร name ดังตัวอย่าง

	โปรแกรม	ผลลัพธ์ พร้อมตัวอย่างข้อมูลนำเข้า
1	<code>name = input("Enter your name: ")</code>	Enter your name: Lattapol
2	<code>age = int(input("Enter your age: "))</code>	Enter your age: 28
3	<code>print("_____")</code>	
4	<code>print("Your name is", name, ".")</code>	Your name is Lattapol .
5	<code>print("You are", age, "years old.")</code>	You are 28 years old.

การเขียนโปรแกรมแบบลำดับ

การเขียนโปรแกรมแบบลำดับ เป็นลักษณะการเขียนคำสั่งให้โปรแกรมทำงานตามลำดับ จากคำสั่งที่ 1 คำสั่งที่ 2 ไปเรื่อย ๆ จนครบทุกคำสั่ง



ในการเขียนโปรแกรมแบบลำดับมักมีการคำนวณ ดังนั้นต้องทำความเข้าใจเกี่ยวกับนิพจน์ ตัวดำเนินการทางคณิตศาสตร์ ลำดับความสำคัญของตัวดำเนินการให้แม่นยำ เพื่อให้สามารถออกแบบขั้นตอนวิธีและเขียนโปรแกรมแบบลำดับได้อย่างถูกต้อง

1. นิพจน์ (Expression)

เป็นการดำเนินการเพื่อคำนวณหรือทำการเปรียบเทียบค่าต่าง ๆ ตามที่ต้องการในการดำเนินการประกอบด้วย

1.1 ตัวถูกดำเนินการ ได้แก่ ค่าคงที่หรือตัวแปร

1.2 ตัวดำเนินการ เช่น $+$ $-$ $*$ $/$ $//$ $**$ $\%$ $<$ $>$ $=$

```
total = (num1 + num2 + num3) / 3
```

2. ตัวดำเนินการทางคณิตศาสตร์(Arithmetic Operator)

เป็นตัวดำเนินการข้อมูลด้วยการคำนวณทางคณิตศาสตร์

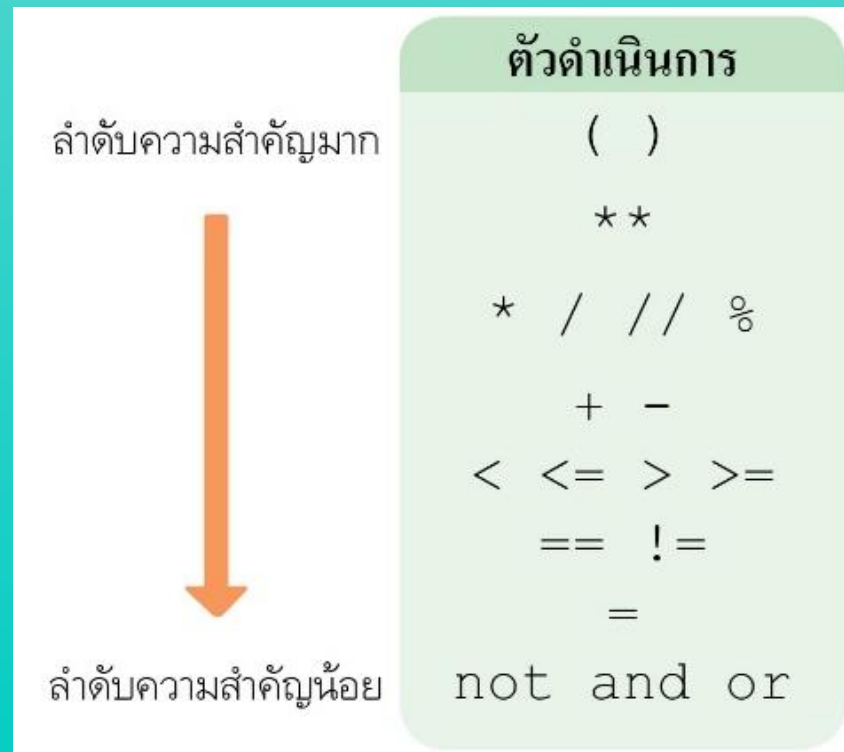
ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน
+	บวก (Addition)	$7 + 4 = 11$
-	ลบ (Subtraction)	$7 - 4 = 3$
*	คูณ (Multiplication)	$7 * 4 = 28$
/	หาร (Division)	$7 / 4 = 1.75$
//	หารปัดเศษ (Floor Division)	$7 // 4 = 1$
%	หารเอาเฉพาะเศษ (Modulo)	$7 \% 4 = 3$
**	ยกกำลัง (Exponent)	$7 ** 4 = 2401$

ตัวอย่าง

	โปรแกรม	ผลลัพธ์
1	<code>x = 11</code>	
2	<code>y = 3</code>	
3	<code>print("x + y =", x + y)</code>	<code>x + y = 14</code>
4	<code>print("x - y =", x - y)</code>	<code>x - y = 8</code>
5	<code>print("x * y =", x * y)</code>	<code>x * y = 33</code>
6	<code>print("x / y =", x / y)</code>	<code>x / y = 3.6666666666666665</code>
7	<code>print("x // y =", x // y)</code>	<code>x // y = 3</code>
8	<code>print("x % y =", x % y)</code>	<code>x % y = 2</code>
9	<code>print("x ** y =", x ** y)</code>	<code>x ** y = 1331</code>

3. ลำดับก่อน/หลังของตัวดำเนินการทางคณิตศาสตร์

เมื่อมีการใช้ตัวดำเนินการมากกว่า 1 ตัว การดำเนินการจะต้องเป็นไปตามลำดับ เช่น $1 + 4 \times 5$ จะได้ผลลัพธ์ 21 เพราะอินเทอร์พรีเตอร์จะทำ 4×5 ก่อน แล้วจึง $+ 1$ แต่ถ้าต้องการให้ $1 + 4$ ก่อน จะต้องเขียนนิพจน์ใหม่เป็น $(1 + 4) \times 5$ ตัวดำเนินการทางคณิตศาสตร์จึงมีลำดับในการทำงานก่อนหรือหลังตามกฎต่อไปนี้



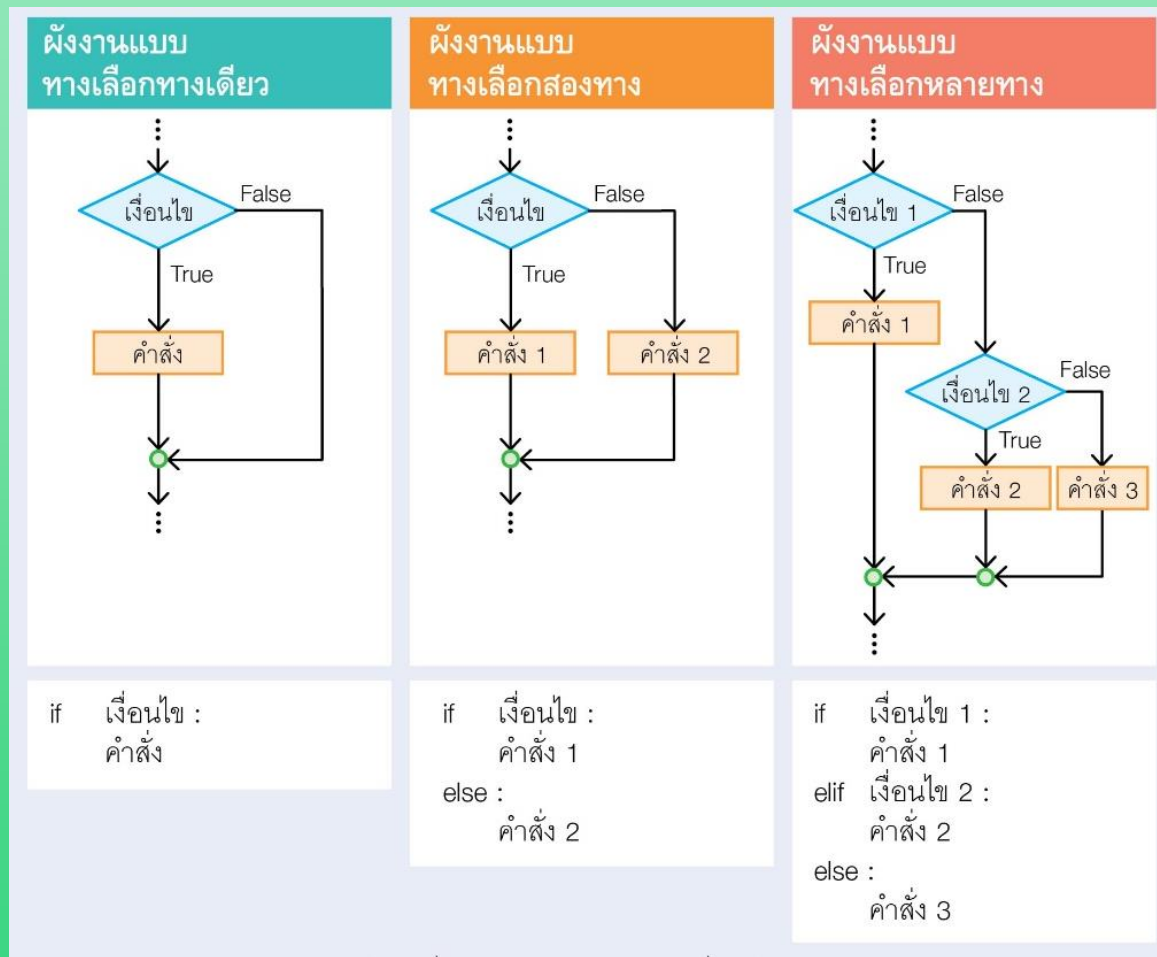


การเขียนโปรแกรมแบบลำดับ

มีลักษณะอย่างไร ?

การเขียนโปรแกรมแบบทางเลือก

รูปแบบโครงสร้างโปรแกรมทางเลือกมี 4 ประเภท ดังนี้



และแบบทางเลือกเชิงซ้อน
เป็นรูปแบบการทำงานที่
ซับซ้อน

การเขียนโปรแกรม

แบบทางเลือกคืออะไร ?



1. ตัวดำเนินการเปรียบเทียบ

เป็นตัวการที่นำข้อมูลมาเปรียบเทียบกัน ผลลัพธ์ที่ได้จะเป็นค่า boolean ได้แก่
จริง (True) หรือเท็จ (False)

ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน
==	เท่ากับ (Equal)	7 == 4 = False
!=	ไม่เท่ากับ (Not Equal)	7 != 4 = True
<	น้อยกว่า (Less than)	7 < 4 = False
<=	น้อยกว่าหรือเท่ากับ (Less than or equal)	7 <= 4 = False
>	มากกว่า (More than)	7 > 4 = True
>=	มากกว่าหรือเท่ากับ (More than or equal)	7 >= 4 = True

ตัวอย่าง

	โปรแกรม	ผลลัพธ์
1	x = 3	
2	y = 11	
3	print("x == y =", x == y)	x == y = False
4	print("x != y =", x != y)	x != y = True
5	print("x < y =", x < y)	x < y = True
6	print("x <= y =", x <= y)	x <= y = True
7	print("x > y =", x > y)	x > y = False
8	print("x >= y =", x >= y)	x >= y = False

2. ตัวดำเนินการตรรกะ

เป็นตัวการที่นำข้อมูลมาเปรียบเทียบกับ ผลลัพธ์ที่ได้จะเป็นค่า boolean ได้แก่
จริง (True) หรือเท็จ (False)

ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน
and	และ	False and False = False
or	หรือ	False or False = False
not	นิเสธ	Not False = True

p	q	p and q	p or q	not p	not q
True	True	True	True	False	False
True	False	False	True	False	True
False	True	False	True	True	False
False	False	False	False	True	True

ตัวอย่าง

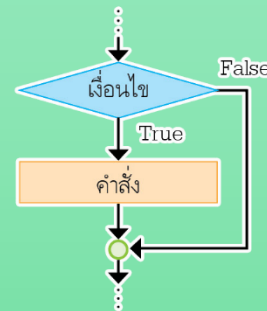
	โปรแกรม	ผลลัพธ์
1	<code>x = True</code>	
2	<code>y = False</code>	
3	<code>print("x and y =", x and y)</code>	x and y = False
4	<code>print("x or y =", x or y)</code>	x or y = True
5	<code>print("not x =", not x)</code>	not x = False
6	<code>print("not y =", not y)</code>	not y = True

3. โครงสร้างแบบทางเลือกทางเดียว

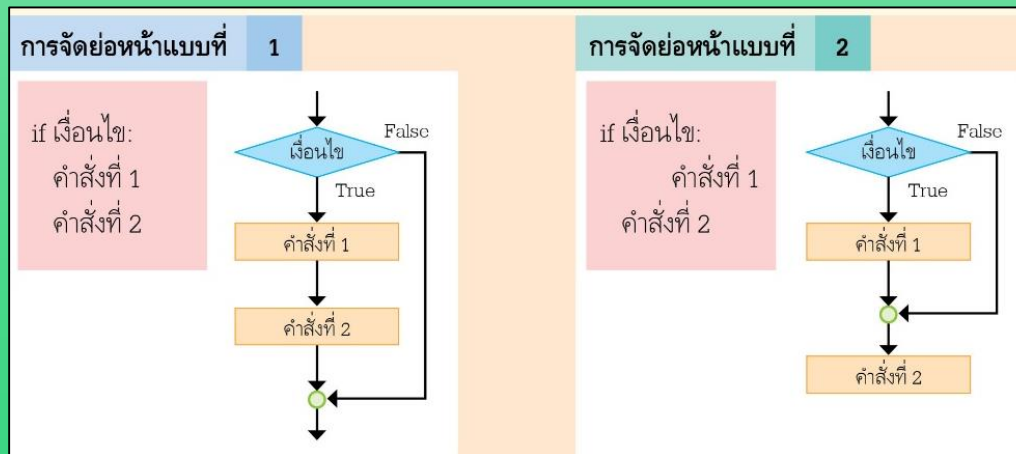
รูปแบบการเขียนโปรแกรม
แบบทางเลือกทางเดียว (if)

if เงื่อนไข:
คำสั่ง

ผังงานแสดงรูปแบบการทำงาน
แบบทางเลือกทางเดียว



ในภาษา Python จะมีการย่อหน้า เพื่อแบ่งบล็อกของโปรแกรมแต่ละส่วน ซึ่งจะต้องกำหนดย่อหน้าให้ตรงกันและถูกต้องตามลักษณะการทำงาน

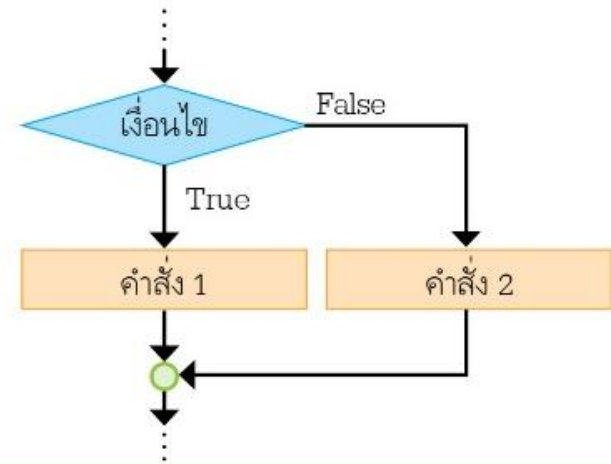


4. โครงสร้างแบบทางเลือกสองทาง

รูปแบบการเขียนโปรแกรม
แบบทางเลือกสองทาง (if-else)

```
if เงื่อนไข:  
    คำสั่ง1  
else:  
    คำสั่ง2
```

ผังงานแสดงรูปแบบการทำงาน
แบบทางเลือกสองทาง

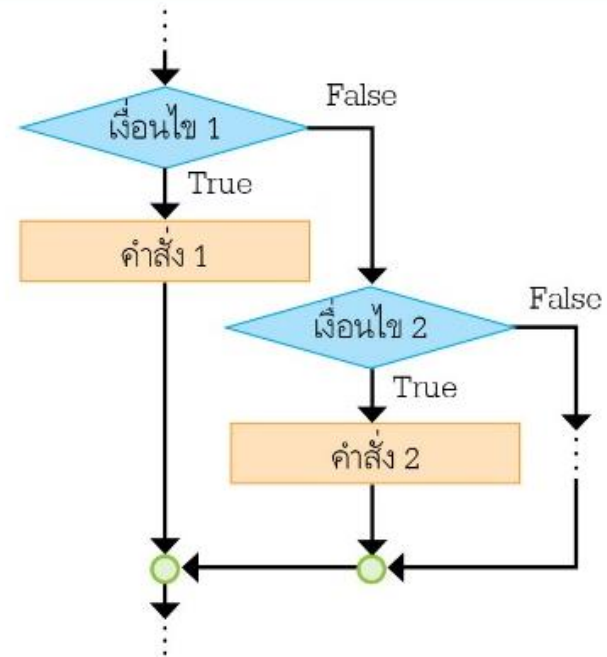


5. โครงสร้างแบบทางเลือกหลายทาง

รูปแบบการเขียนโปรแกรม
แบบทางเลือกหลายทาง (if-elif-else)

```
if เงื่อนไข1:  
    คำสั่ง1  
elif:  
    คำสั่ง2  
...  
else:  
    คำสั่งn
```

ผังงานแสดงรูปแบบการทำงาน
แบบทางเลือกหลายทาง



การเขียนโปรแกรมแบบวนซ้ำ

เป็นลักษณะการเขียนโปรแกรมให้ทำงานซ้ำตามจำนวนรอบหรือตามเงื่อนไขที่กำหนด เรียกว่า วนลูป (Loop) ซึ่งมี 2 รูปแบบ คือ แบบ while และแบบ for

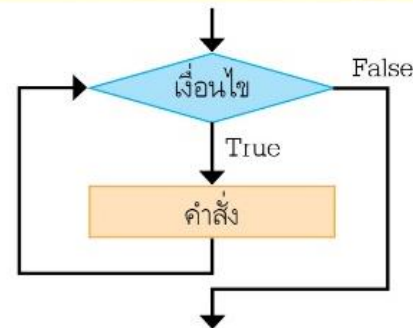
1. โครงสร้างวนซ้ำแบบ while (while loop)

เป็นคำสั่งที่กำหนดให้โปรแกรมทำซ้ำขณะที่เงื่อนไขเป็นจริง ให้โปรแกรมทำงานคำสั่งหลัง while ถ้าเงื่อนไขเป็นเท็จจะออกจากการทำซ้ำแล้วไปทำคำสั่งอื่น คำสั่ง while เหมาะกับงานที่ไม่ทราบรอบของการทำซ้ำที่แน่นอน

รูปแบบการเขียนโปรแกรม
แบบวนซ้ำด้วย while

```
while เงื่อนไข:  
    คำสั่ง
```

ผังงานการแสดงรูปแบบตรวจซ้ำ
ด้วย while (while loop)



2. การใช้คำสั่ง range()

`range()` เป็นฟังก์ชันที่ใช้สร้างลำดับเลขตามขอบเขตที่กำหนด มีรูปแบบการใช้ดังนี้

```
range(x)           # เริ่มจาก 0 ถึง x
range(x, y)        # เริ่มจาก x ถึง y
range(x, y, z)     # เพิ่ม/ลดทีละ z เริ่มจาก x ถึง y
```

โปรแกรมและตัวอย่างผลลัพธ์

	โปรแกรม	หมายถึง
1	<code>range(10)</code>	0 1 2 3 4 5 6 7 8 9
2	<code>range(5, 10)</code>	5 6 7 8 9
3	<code>range(5, 10, 2)</code>	5 7 9
4	<code>range(5, -10, -2)</code>	5 3 1 -1 -3 -5 -7 -9

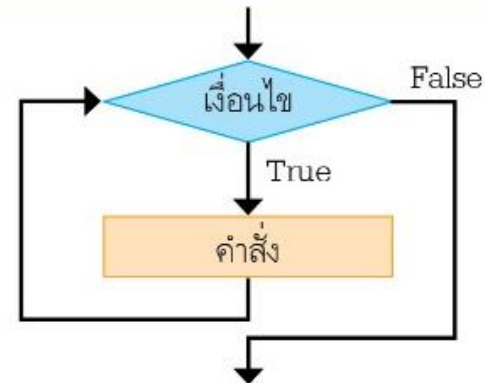
3. โครงสร้างวนซ้ำแบบ for (for loop)

คำสั่ง for เป็นคำสั่งที่กำหนดให้โปรแกรมทำซ้ำด้วยจำนวนรอบที่แน่นอน ถ้าผลการตรวจสอบเงื่อนไขเป็นจริงจะทำคำสั่งหลัง for ถ้าเงื่อนไขเป็นเท็จจะออกจากการทำซ้ำแล้วไปทำงานคำสั่งอื่น

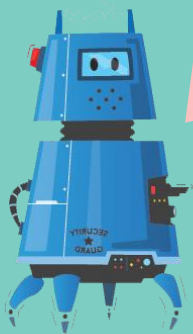
รูปแบบการเขียนโปรแกรม
แบบวนซ้ำด้วย for

```
for ตัวแปร: in ลำดับ:  
    คำสั่ง
```

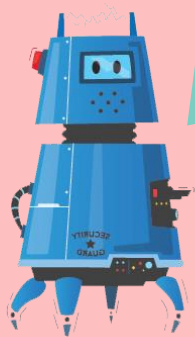
ผังงานแสดงการวนซ้ำ
ด้วยคำสั่ง for (for loop)



การวนซ้ำมีลักษณะไหนบ้าง ?



ฟังก์ชันคืออะไร ?





ฟังก์ชัน

ฟังก์ชัน (Function) เป็นชุดคำสั่งที่มีหน้าที่ทำงานอย่างใดอย่างหนึ่ง สามารถเรียกใช้งานกี่ครั้งก็ได้ โดยไม่ต้องเขียนโปรแกรมขึ้นมาใหม่

ประเภทของฟังก์ชัน

1. ฟังก์ชันจากไลบรารีมาตรฐาน เป็นฟังก์ชันที่ถูกสร้างขึ้นโดยทีมผู้พัฒนาภาษาไพธอน แบ่งเป็น 2 ส่วน คือ

1) Built-in Functions เป็นฟังก์ชันที่ถูกเก็บในตัวแปลภาษา จึงสามารถเรียกใช้ได้ทันที โดยไม่ต้องอ้างอิง

2) Library Functions เป็นฟังก์ชันที่สร้างโดยทีมพัฒนาโปรแกรมแล้วจัดเก็บเป็น โมดูลในไลบรารี ซึ่งก่อนจะนำมาใช้งานต้องเรียกโมดูลของฟังก์ชันที่ต้องการเข้ามาก่อน ถึงจะสามารถอ้างอิงถึงโมดูลและเรียกชื่อฟังก์ชันที่ต้องการใช้งาน

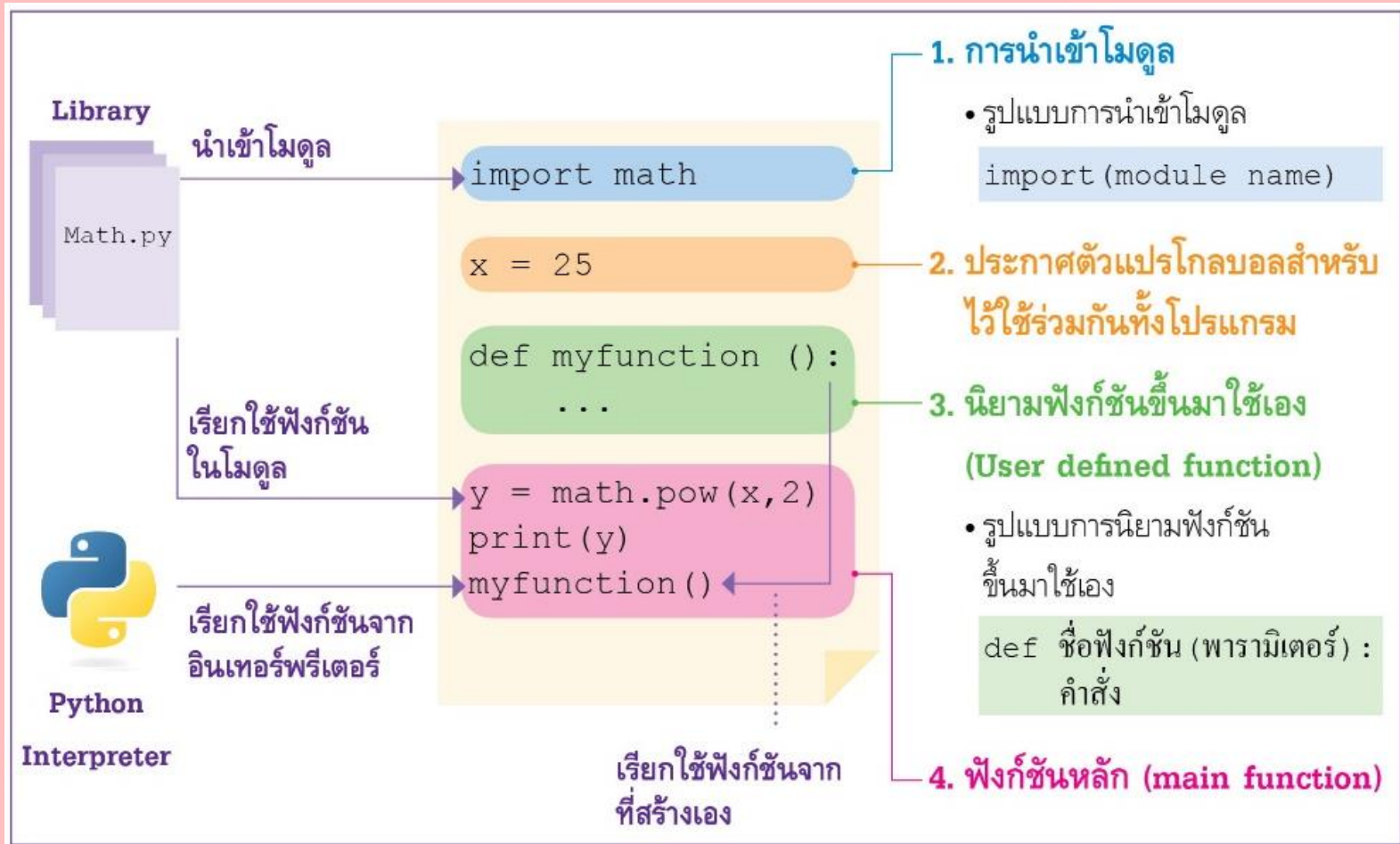
ตัวอย่างการเรียกใช้โมดูล math ในไลบรารี มีดังนี้

```
import math
```

การเรียกใช้ฟังก์ชัน sqrt() จะต้องมีชื่อโมดูลตามด้วย . แล้วใส่ชื่อฟังก์ชันที่ต้องการ

```
math.sqrt(x)
```

2. ฟังก์ชันที่ผู้เขียนโปรแกรมสร้างขึ้นมาเอง เป็นฟังก์ชันที่นิยามขึ้นมาเพื่อใช้เอง จากนั้นจึงเรียกใช้ฟังก์ชันที่สร้างขึ้นในตำแหน่งถัดมา



การกำหนดฟังก์ชัน เป็นการสร้างชุดคำสั่งเพื่อนำไปเรียกใช้ในส่วนใดส่วนหนึ่งของโปรแกรมได้บ่อย ๆ โดยมีรายละเอียดดังนี้

รูปแบบการนิยามฟังก์ชัน มีดังนี้



```
def myfunction(x) :  
    คำสั่ง
```



การเรียกใช้ฟังก์ชันสามารถเรียกใช้ได้ในส่วนต่าง ๆ ของโปรแกรม ฟังก์ชันที่ไม่มีการส่งผ่านค่าเมื่อเรียกชื่อฟังก์ชันโปรแกรมจะทำงานทันที แต่ถ้าฟังก์ชันนั้นเป็นฟังก์ชันที่ต้องส่งผ่านค่าเข้าไปในฟังก์ชัน เมื่อเรียกใช้จะต้องส่งค่าเข้าไปด้วย เช่น

```
myfunction(5)
```

```
myfunction(x)
```

ตัวอย่างนี้บรรทัดแรกจะส่งค่า 5 เข้าไปในฟังก์ชัน บรรทัดที่สองจะส่งค่าตัวแปร x เข้าไปในฟังก์ชันที่สร้างขึ้น



สำหรับฟังก์ชันที่มีการส่งค่ากลับจะต้องประกาศตัวแปรสำหรับค่าที่ได้จากฟังก์ชันด้วย โดยมีรูปแบบดังนี้

```
ตัวแปร = myfunction(x)
```

ตัวอย่าง

```
a = myfunction(x)
```