

ทำความรู้จัก Python เบื้องต้น

ภาษาไพทอนเป็นภาษาระดับสูง โปรแกรมที่เขียนจึงต้องถูกแปลให้เป็นภาษาเครื่องก่อนที่จะใช้สั่งงานคอมพิวเตอร์ได้โดยตรง การแปลภาษาไพทอนนี้ต้องใช้ตัวแปลภาษาไพทอน (python interpreter) ประเภท **อินเทอร์พรีเตอร์**

การเขียนโปรแกรมภาษาไพทอน สามารถเขียนได้ 2 รูปแบบ คือ

1. การเขียนโปรแกรมไพทอนในโหมดอิมมีเดียท
2. การเขียนโปรแกรมไพทอนในโหมดสคริปต์

ซึ่งแต่ละรูปแบบมีข้อดี ข้อเสีย และลักษณะการเขียนที่ต่างกัน ดังนี้

การเขียนโปรแกรมไพทอนในโหมดอิมมีเดียท

คือการเขียนโปรแกรมในหน้าต่าง Python Shell ซึ่งจะมีการทำงานทุกครั้งทีกดปุ่ม Enter เหมาะกับการทดสอบคำสั่งทีละคำสั่ง หรือเขียนโปรแกรมที่มีขนาดเล็กและไม่ซับซ้อน

การเขียนโปรแกรมไพทอนในโหมดอิมมีเดียท

คือการเขียนโปรแกรมในหน้าต่าง Python Shell ซึ่งจะมีการทำงานทุกครั้งทีกดปุ่ม Enter เหมาะกับการทดสอบคำสั่งทีละคำสั่ง หรือเขียนโปรแกรมที่มีขนาดเล็กและไม่ซับซ้อน

การเขียนโปรแกรมไพทอนในโหมดสคริปต์

แม้ว่าการรันคำสั่งโปรแกรมไพทอนในโหมดอิมมีเดียทจะทำได้ง่ายและได้ผลลัพธ์ทันที แต่ก็ไม่เหมาะสมในกรณีที่ต้องการเขียนโปรแกรมซับซ้อนหรือมีขนาดใหญ่ ประกอบด้วยคำสั่งจำนวนมาก การรันคำสั่งจำนวนมากในโหมดอิมมีเดียทจะต้องป้อนคำสั่งทีละบรรทัด ทำให้ไม่สะดวก

ในโหมดสคริปต์ทำให้สามารถเขียนคำสั่งทั้งหมดแล้วบันทึกไว้เป็นไฟล์ จากนั้นรันทุกคำสั่งในไฟล์ต่อเนื่องกันเพื่อให้ได้ผลลัพธ์ได้โดยการสั่งเพียงคำสั่งเดียว

ค่าและชนิดของข้อมูล

ข้อมูลจัดเป็นสิ่งที่พื้นฐานที่สุดที่สามารถนำไปประมวลผลได้ ส่วนประกอบของข้อมูล เช่น ชนิดของข้อมูล ค่าของข้อมูล ขอบเขตของข้อมูล ชนิดของข้อมูล เช่น ตัวเลข ข้อความหรือสตริง (string) ตัวอย่างค่าของข้อมูล เช่น 16 หรือ “Hello”

ค่าของข้อมูล (value) แบ่งออกได้เป็นหลายชนิด (data type) หรือหลายคลาส (class) เช่น 16 เป็นชนิดจำนวนเต็ม (integer) และ “Hello” เป็น สตริง ในไพทอนมีฟังก์ชัน `type()` ที่จะสามารถบอกคลาสของค่าข้อมูลได้

การตรวจสอบคลาสของข้อมูลด้วยฟังก์ชัน type() ในโหมดอิมมีเดียท

```
>>> type("Hello")  
<class 'str'>  
>>> type(16)  
<class 'int'>  
>>> type(3.98)  
<class 'float'>
```

การตรวจสอบคลาสของข้อมูลด้วยฟังก์ชัน type() ในโหมดอิมมีเดียท

```
>>> type("Hello")  
<class 'str'>  
>>> type(16)  
<class 'int'>  
>>> type(3.98)  
<class 'float'>
```

ค่าของข้อมูลที่ใช้ในการประมวลผลมีหลายคลาส เช่น **จำนวนเต็ม (int)**, **สตริง (str)**, **จำนวนจริง (float)**, และ **บูล (bool)** คลาสสามารถนำมาคำนวณรวมกันได้ เช่น คลาส int สามารถนำมาคำนวณร่วมกับคลาส float และผลที่ได้จะเป็นคลาส float แต่ตัวเลขหรือนิพจน์ที่อยู่ภายในเครื่องหมายคำพูดไม่สามารถนำมาใช้คำนวณได้ เพราะถือเป็นคลาส str ส่วนค่าของข้อมูลในคลาส bool จะมีค่าเป็นจริง (true) หรือ เท็จ (false) เท่านั้น

ตัวแปร

ในไพทอนผู้เขียนโปรแกรมสามารถสร้างตัวแปร (variable) ขึ้นมาเพื่ออ้างอิงหรือชี้ไปยังค่าของข้อมูลเพื่อใช้งานภายหลังได้ ในขณะใดขณะหนึ่งตัวแปรสามารถชี้ไปยังค่าของข้อมูลได้เพียงค่าเดียว โดยผู้เขียนโปรแกรมสามารถใช้คำสั่งกำหนดค่า (assignment statement) เพื่อใช้ในการกำหนดให้ตัวแปรชี้ไปยังค่าที่ต้องการ เช่น

กำหนดตัวแปร `c = 16` หรือ `msg = "hello world"`

สังเกตว่าในคำสั่งกำหนดค่าจะมีองค์ประกอบของคำสั่ง คือ มีชื่อตัวแปรอยู่ทางซ้ายของเครื่องหมายกำหนดค่า (เครื่องหมายเท่ากับ) และนิพจน์ที่ต้องการกำหนดให้กับตัวแปรอยู่ทางขวา

การตั้งชื่อตัวแปร มีหลักการดังนี้

1. ใช้อักขระ(ภาษาอังกฤษ) หรือตัวเลข และ _ (underscore)
2. ตัวแรกจะต้องเป็นอักขระหรือ _ เท่านั้น
3. ห้ามใช้คำสงวน (Reserved Words)
4. ตัวพิมพ์เล็กและตัวพิมพ์ใหญ่มีความหมายต่างกัน

ผู้เขียนโปรแกรมควรเลือกแนวทางการตั้งชื่อตัวแปรให้เป็นแนวทางเดียวกัน เพื่อง่ายต่อการทำความเข้าใจกับโปรแกรม และควรตั้งชื่อตัวแปรให้เหมาะสม เพราะเมื่อเขียนโปรแกรมขนาดใหญ่ต้องใช้ตัวแปรจำนวนมาก การตั้งชื่อตัวแปรที่มีความหมายจะทำให้เข้าใจโปรแกรมได้รวดเร็ว และลดความผิดพลาดในการเรียกใช้ตัวแปรได้ ดังนั้น ชื่อตัวแปรควรสื่อถึงวัตถุประสงค์ของข้อมูลที่เก็บด้วย เช่น ตัวแปรที่ชี้ไปยังค่าเงินเดือนอาจตั้งชื่อเป็น salary ตัวแปรที่ชี้ไปยังอายุอาจตั้งชื่อเป็น age เป็นต้น

ชื่อตัวแปรที่กำหนดขึ้น
ต้องไม่ซ้ำกับคำหลัก (keyword)
ที่ไพทอนใช้เป็นคำสั่ง เนื่องจาก
ไพทอนจะไม่สามารถแยกได้ว่าคำ
ใดเป็นค่าที่ผู้เขียนโปรแกรม
ต้องการให้เป็นตัวแปร หรือคำใด
ต้องการให้เป็นคำสั่ง คำหลักใน
ไพทอนมีดังนี้

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

สตริง

การกำหนดข้อความในไพทอนผู้เขียนโปรแกรมสามารถเลือกใช้เครื่องหมายอัญประกาศเดี่ยว (' ') หรือเครื่องหมายอัญประกาศคู่ (" ") ได้ตามความเหมาะสม เช่น “Navamin”



นิพจน์และตัวดำเนินการ

นิพจน์ (expression) ในไพทอน ได้แก่ ค่าของข้อมูลหรือค่าคงตัว (literal) ตัวแปร หรือการรวมกลุ่มของสิ่งดังกล่าวกับตัวดำเนินการ (operation) นิพจน์มีหลายประเภท เช่น นิพจน์คณิตศาสตร์ นิพจน์เปรียบเทียบ และนิพจน์ตรรกะ



นิพจน์และตัวดำเนินการ

นิพจน์คณิตศาสตร์

นิพจน์คณิตศาสตร์เป็นนิพจน์ที่ใช้ตัวดำเนินการคณิตศาสตร์ (arithmetic operation) ในการประมวลผล เพื่อให้ได้ผลลัพธ์เป็นจำนวน

ตัวดำเนินการทางคณิตศาสตร์ที่ใช้ในภาษาไพทอน ได้แก่

ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน	ผลลัพธ์
()	จัดกลุ่ม	$(2 + 9) * 5$	55
**	ยกกำลัง	$5 ** 2$	25
*	คูณ	$10 * 3$	30
/	หารจำนวนจริง	$25 / 5$	5.0
//	หารปัดเศษทิ้ง	$8 // 3$	2
%	หารเอาเศษ	$28 \% 24$	4
+	บวก	$5 + 7$	12
-	ลบ	$10 - 2$	8

ตัวดำเนินการคณิตศาสตร์มีลำดับความสำคัญที่ใช้ในการประมวลผล ถ้าหากในกรณีที่มีตัวดำเนินการหลายตัวอยู่ในนิพจน์เดียวกัน การตัดสินใจว่าตัวดำเนินการใดจะทำงานก่อนหรือหลังขึ้นอยู่กับลำดับความสำคัญ โดยตัวดำเนินการที่มีลำดับความสำคัญสูงจะทำงานก่อน ตัวดำเนินการแต่ละตัวมีลำดับความสำคัญแตกต่างกัน โดยเรียงจากลำดับความสำคัญสูงไปยังลำดับความสำคัญต่ำ ดังนี้

1. () จัดกลุ่ม มีความสำคัญสูงสุด
2. ** ยกกำลัง มีความสำคัญรองลงมา
3. * / // % คูณหารหารปิดเศษทิ้ง และหารเอาเศษ มีความสำคัญเท่ากันโดยจะกระทำตามลำดับจากซ้ายไปขวาในนิพจน์
4. + - บวกและลบ มีความสำคัญต่ำสุด โดยจะกระทำจากซ้ายไปขวา

นิพจน์เปรียบเทียบ

นิพจน์เปรียบเทียบ (relational expression) เป็นนิพจน์ที่ใช้ตัวดำเนินการเปรียบเทียบ (relational operation) ในการเปรียบเทียบค่าของนิพจน์คณิตศาสตร์สองนิพจน์ ซึ่งในภาษาไพทอนจะได้ผลลัพธ์เป็นคลาส bool ซึ่งมีค่าเป็น True หรือ False เท่านั้น

ตัวดำเนินการทางเปรียบเทียบที่ใช้ในภาษาไพทอน โดยกำหนด $x = 5$ และ $y = 7$

ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน	ผลลัพธ์
==	เท่ากัน	$x == y$	False
!=	ไม่เท่ากัน	$x != y$	True
>=	มากกว่าหรือเท่ากับ	$x >= y$	False
<=	น้อยกว่าหรือเท่ากับ	$x <= y$	True
>	มากกว่า	$x > y$	False
<	น้อยกว่า	$x < y$	True

การใช้นิพจน์เปรียบเทียบ

```
>>> a = 20
>>> a < 50
True
>>> a > 25
False
>>> a == 20
True
>>> type(a != 20)
<class 'bool'>
```

ผลลัพธ์ที่ได้จะพบว่า เมื่อกำหนดค่าตัวแปร a ให้มีค่าเท่ากับ 20 และนำตัวแปร a ไปใช้ตัวดำเนินการเปรียบเทียบจะได้ผลลัพธ์เป็นค่า จริง (True) หรือเท็จ (False) เท่านั้น ซึ่งมีชนิดเป็น bool นั่นเอง

นิพจน์ตรรกะ

นิพจน์ตรรกะ (logical expression) เป็นนิพจน์ที่ใช้ตัวดำเนินการตรรกะ (logical operation) ในการประมวลผลตัวถูกดำเนินการที่อยู่ในคลาส bool

ตัวดำเนินการตรรกะที่ใช้ในภาษาไพทอน มีดังนี้

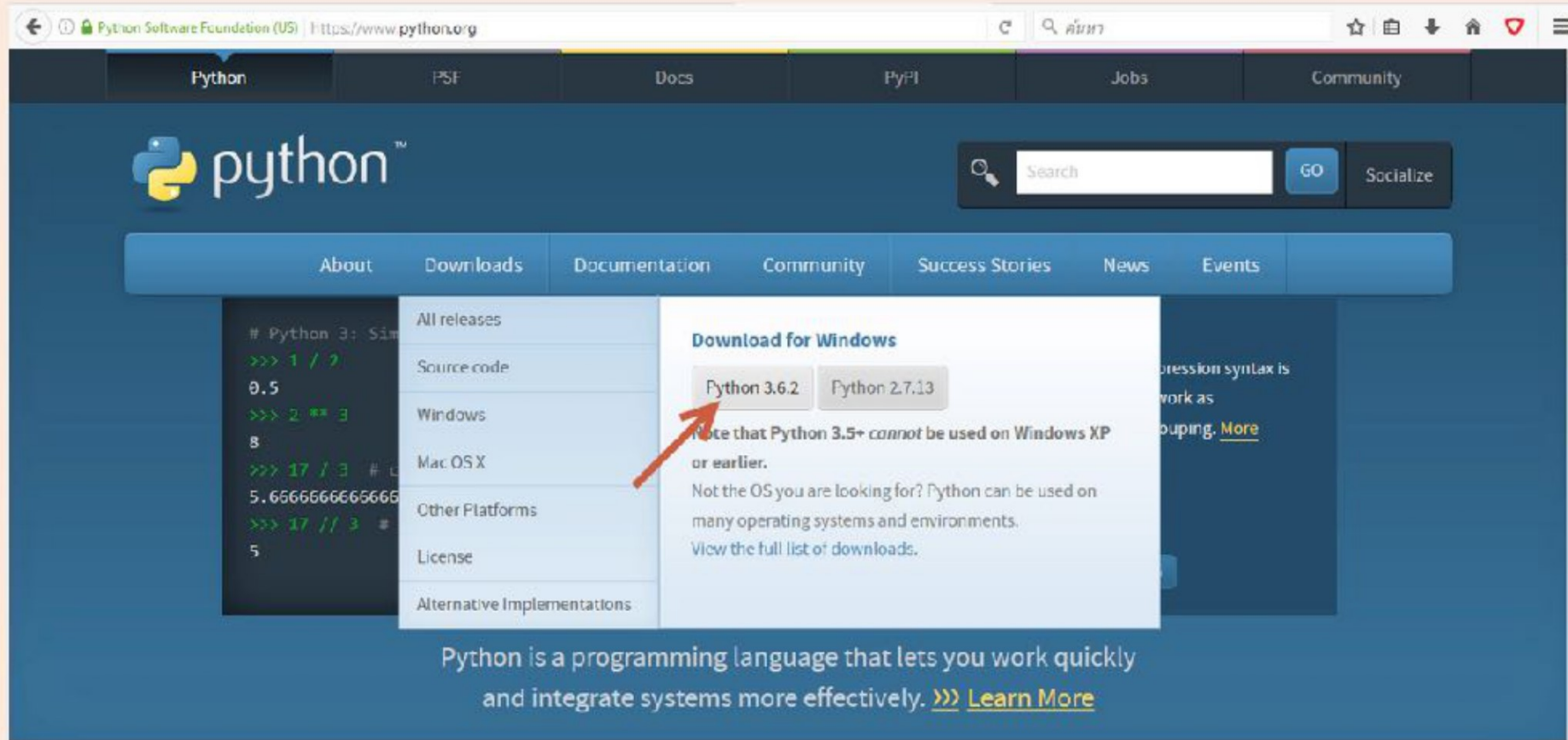
ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน	ผลลัพธ์
not	นิเสธ	not 5 > 6	True
and	และ	3 < 4 and 5 < 6	False
or	หรือ	3 < 4 or 5 < 6	True

ตัวดำเนินการ and และ or ต้องการตัวถูกดำเนินการสองตัว แต่ตัวดำเนินการ not ต้องการตัวถูกดำเนินการเพียงตัวเดียว โดยตัวดำเนินการตรรกะมีลำดับความสำคัญเรียงจากลำดับความสำคัญสูงไปยังลำดับความสำคัญต่ำ คือ not and or

การใช้นิพจน์ตรรกะ

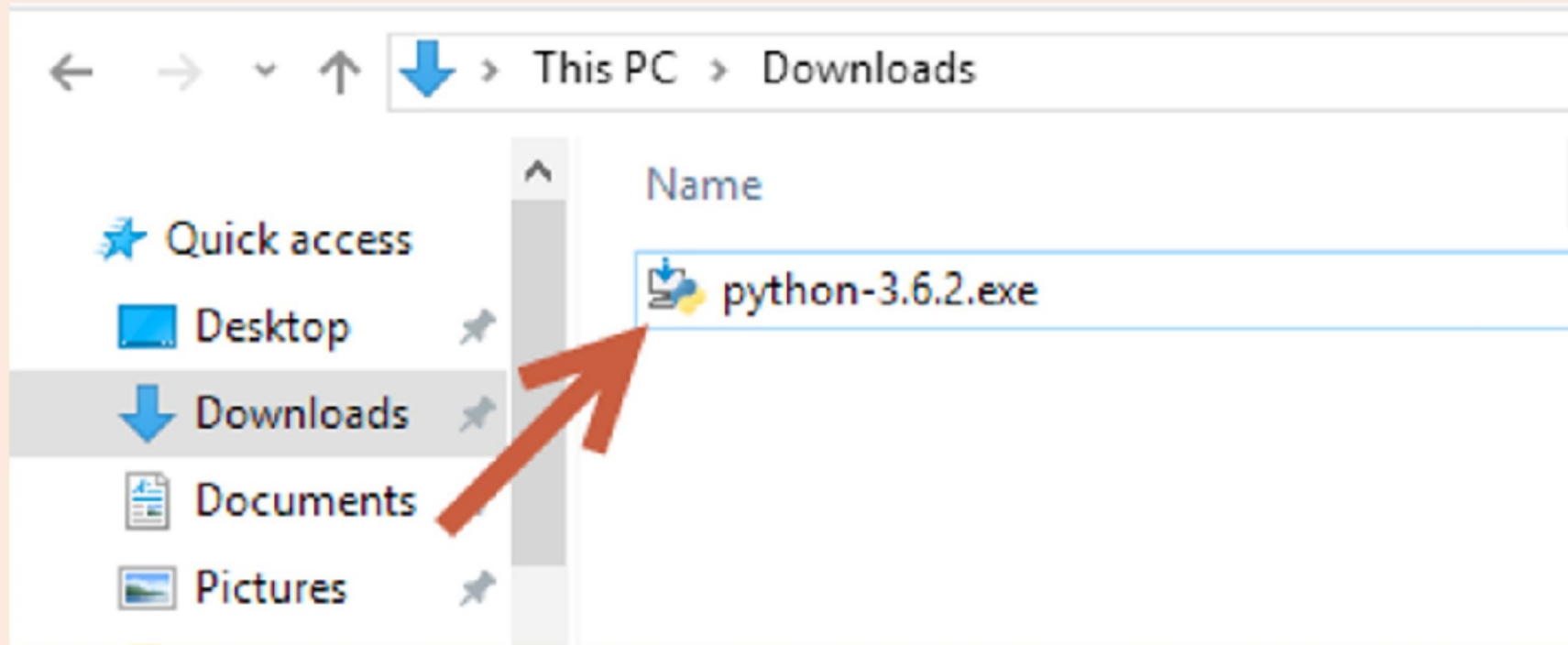
```
>>> a = 30
>>> a <= 25 and a >= 30
False
>>> a <= 25 or a >= 30
True
>>> not a != 30
True
```

1. ดาวน์โหลดโปรแกรมจากเว็บไซต์ www.python.org
2. เลือกเมนู Downloads -> Python 3.6.2 หรือใหม่กว่า



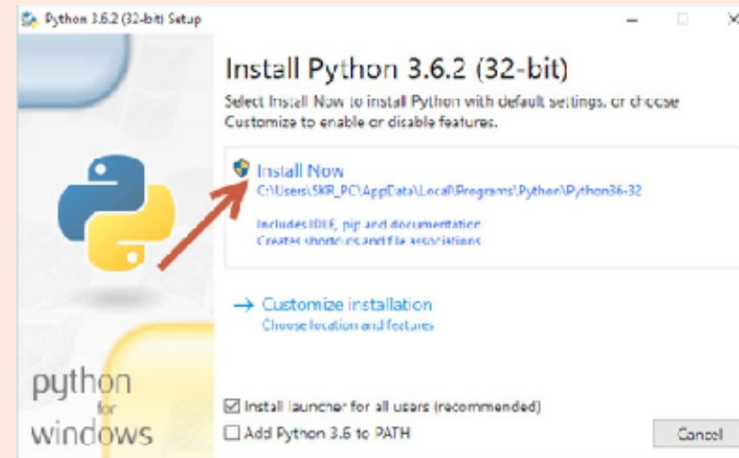
3. เลือก “บันทึกไฟล์”

4. โปรแกรมจะถูกดาวน์โหลดไว้ในโฟลเดอร์ที่กำหนดไว้ หรือในโฟลเดอร์ Downloads

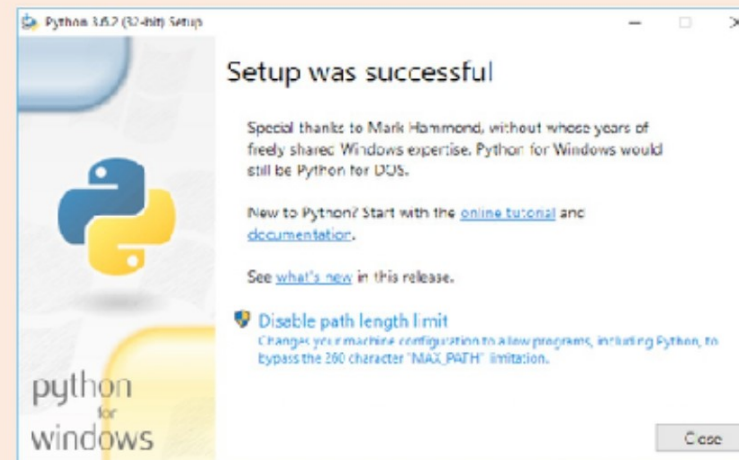


5. ดับเบิลคลิกเพื่อติดตั้งโปรแกรม

6. จะปรากฏหน้าต่างดังรูป ให้คลิกที่ “Install Now”

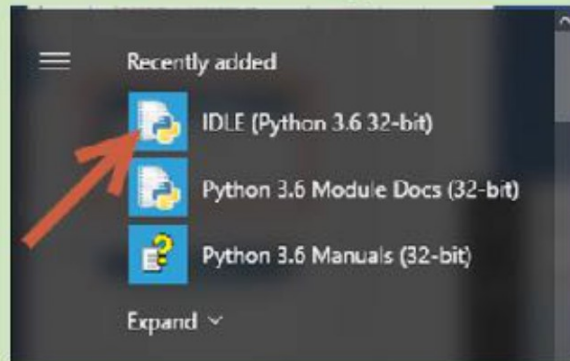


7. โปรแกรมจะทำการติดตั้งตามไฟล์เดอร์ที่กำหนดไว้จนเสร็จสิ้น และสามารถใช้งานได้ทันที



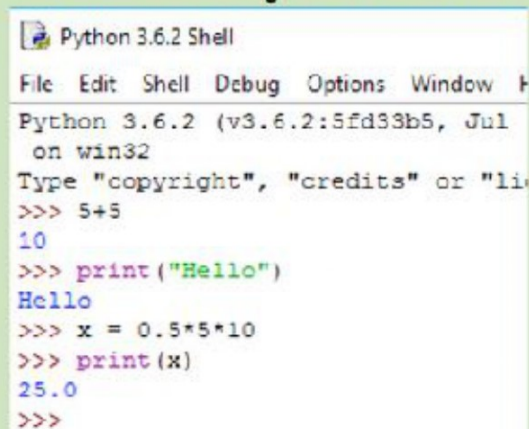
ขั้นตอนการเขียนโปรแกรมไพทอนในโหมดอิมมีเดียท

1) เปิดไอดีอีภาษาไพทอน (Python IDE) จะได้หน้าต่าง IDE หรือ Shell ดังรูป



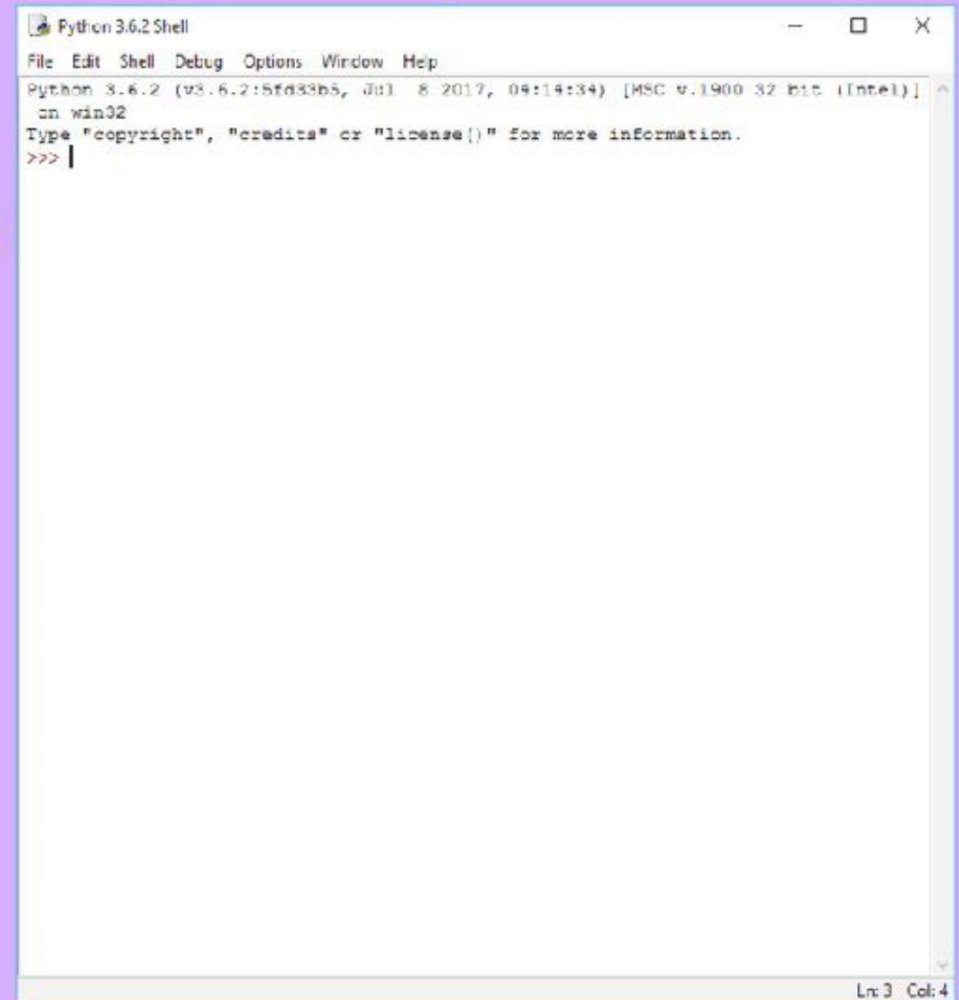
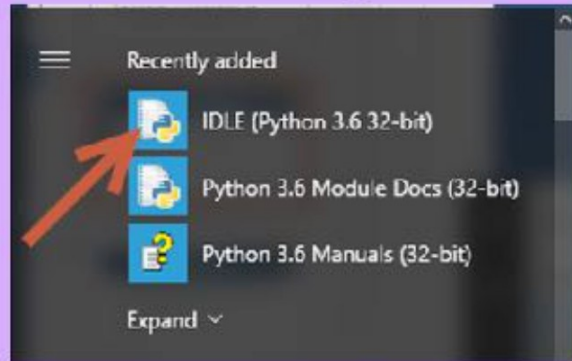
2) สามารถเขียนคำสั่งไพทอนได้ตามต้องการ
โดยคำสั่งที่เขียนจะถูกทำงานทันทีเมื่อกดปุ่ม Enter

เช่น

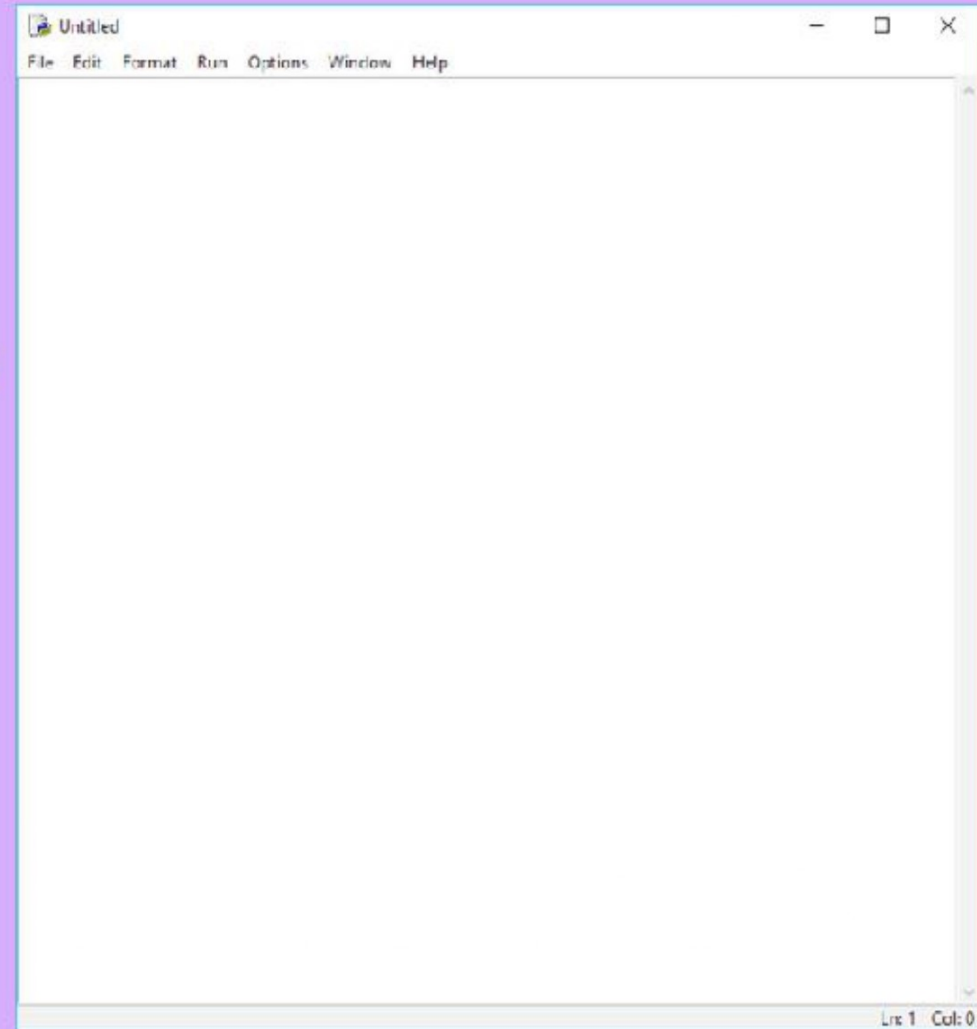
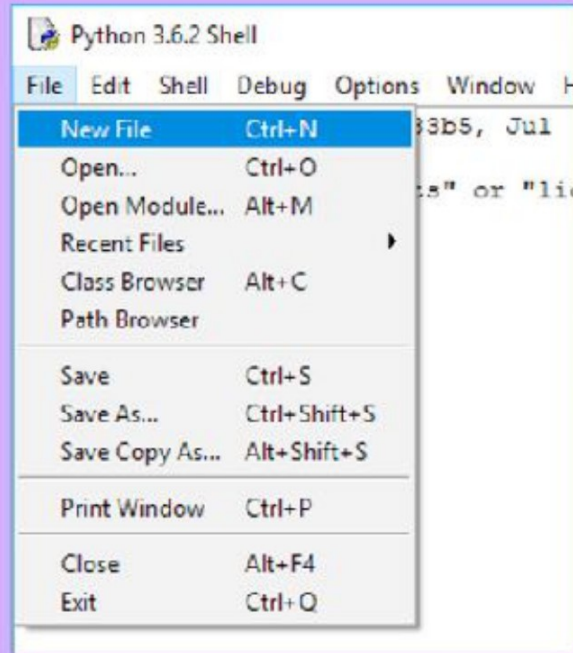


ขั้นตอนการเขียนโปรแกรมไพทอนในโหมดสคริปต์

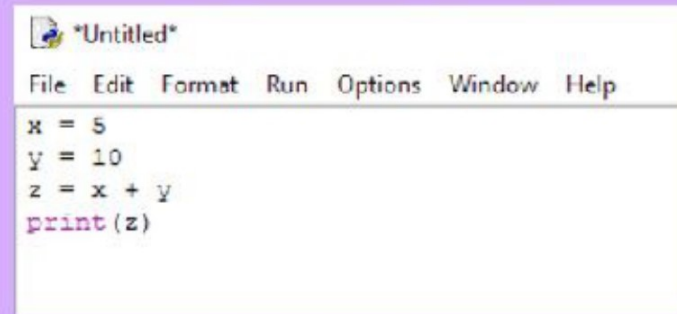
1) เปิดไอทีอีภาษาไพทอน (Python IDE) จะได้หน้าต่าง IDE หรือ Shell ดังรูป



2. เลือกเมนู “File” -> “New File” จะได้หน้าต่างสำหรับเขียนโปรแกรมในโหมดสคริปต์ ดังรูป

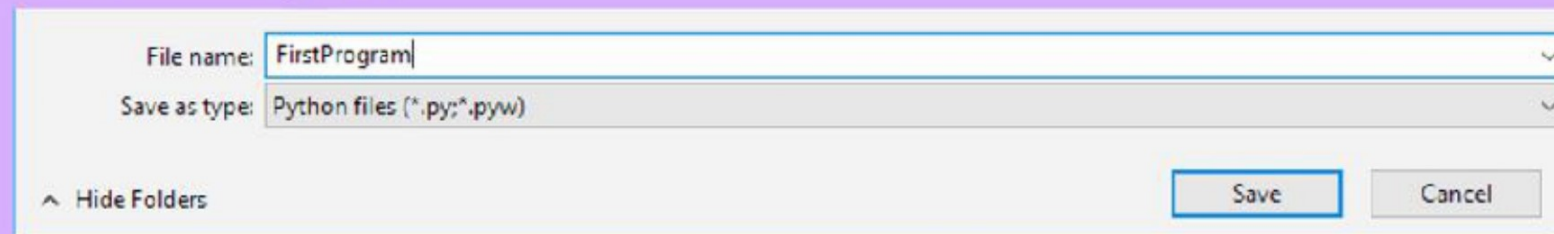


3. สามารถเขียนโปรแกรมได้ทันทีและเขียนให้เสร็จสิ้นก่อนแล้วจึงรันโปรแกรมได้

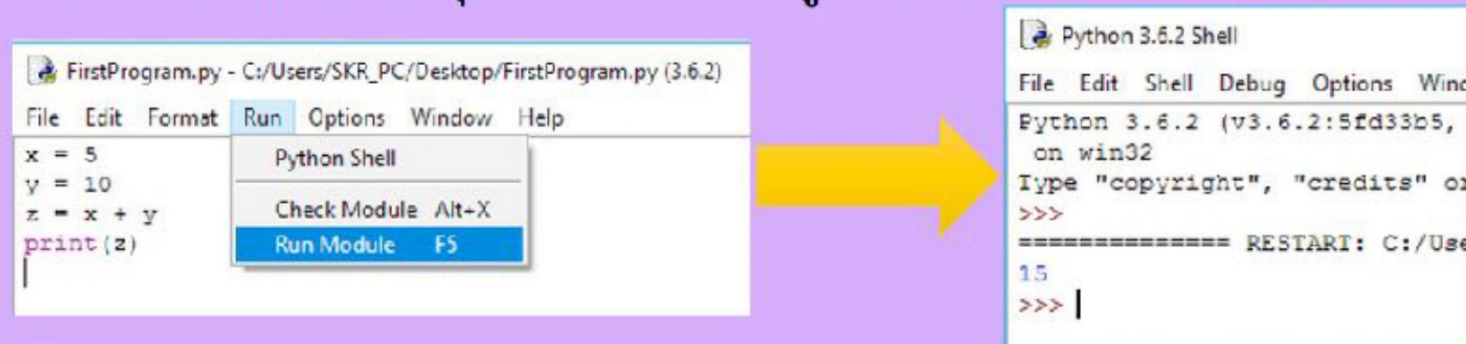


```
*Untitled*
File Edit Format Run Options Window Help
x = 5
y = 10
z = x + y
print(z)
```

4. บันทึกไฟล์เป็นนามสกุล .py



5. รันโปรแกรมโดยการกดปุ่ม F5 หรือเลือกเมนู Run -> Run Module จะได้ผลการรันโปรแกรม ดังรูป



ลองคิด...ลองเขียนโปรแกรม

จงเขียนโปรแกรม โดยมีข้อกำหนด ดังนี้

สร้างตัวแปรชื่อ “name” สำหรับเก็บชื่อของตนเอง

สร้างตัวแปรชื่อ “surname” สำหรับเก็บนามสกุลของตนเอง

สร้างตัวแปรชื่อ “age” สำหรับเก็บอายุของตนเอง

สร้างตัวแปรชื่อ “weight” สำหรับเก็บน้ำหนักของตนเอง (kg)

สร้างตัวแปรชื่อ “height” สำหรับเก็บความสูงของตนเอง (cm)

แล้วใช้คำสั่ง `print()` แสดงข้อมูลทั้งหมดออกทางหน้าจอ

เริ่มเขียนโปรแกรมภาษา Python



ฟังก์ชันแสดงผล

ฟังก์ชันแสดงผล (print function) เป็นฟังก์ชันที่สำคัญฟังก์ชันหนึ่งที่มีมักจะถูกเรียกใช้งานอยู่บ่อยครั้ง เพราะเป็นฟังก์ชันที่ทำหน้าที่ในการแสดงผลลัพธ์หรือข้อมูลที่ต้องการออกสู่หน้าจอแสดงผล ในไพทอนมีฟังก์ชันที่ใช้แสดงผล คือฟังก์ชัน `print()`