



วิทยาการคำนวณ ม.1

OCTOBER
Monday
08

บทที่ 3 การโปรแกรมด้วยภาษาไพทอน

คลิกเพื่อเข้าสู่บทเรียน





เมนูหลัก

คำชี้แจง

รู้จักไพทอน

ผู้จัดทำ

ตัวแปร

ชนิดข้อมูลพื้นฐาน

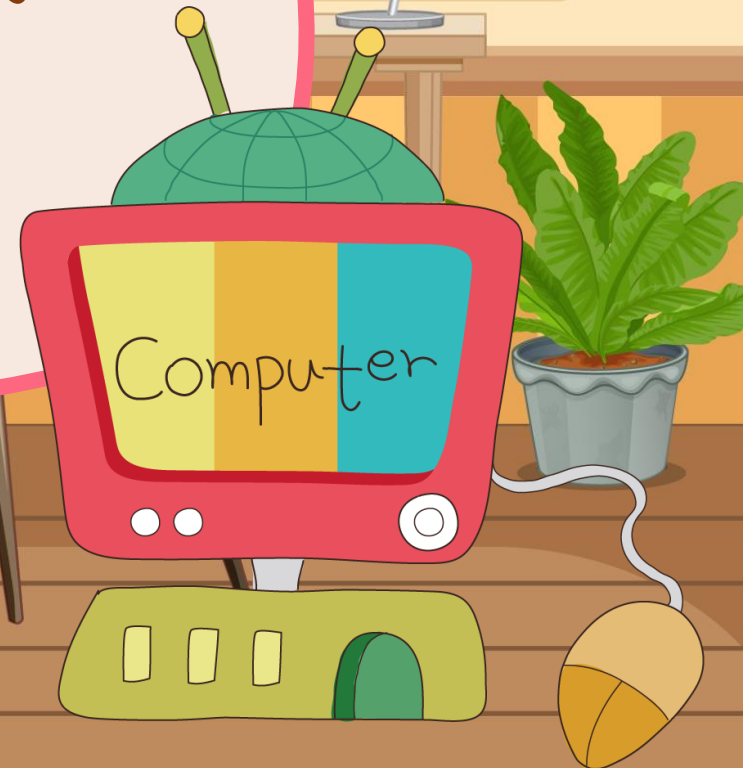
การแปลงชนิดข้อมูล

7	8	9	10	11	12	13
14	15	16	17	18	19	20



คำชี้แจง

1. ให้นักเรียนศึกษาบทเรียน
2. ให้นักเรียนทำแบบทดสอบหลังเรียนทั้งหมด 10 ข้อ 10 คะแนน



1

รู้จักไพทอน



จุดประสงค์การเรียนรู้

1. นักเรียนสามารถออกแบบและเขียนโปรแกรมที่มีการใช้งานตัวแปรได้
2. นักเรียนสามารถออกแบบและเขียนโปรแกรมที่ทำงานแบบมีทางเลือกได้
3. สามารถออกแบบและเขียนโปรแกรมที่มีการทำงานแบบวนซ้ำได้
4. นักเรียนสามารถเขียนโปรแกรมเพื่อแก้ปัญหาทางคณิตศาสตร์หรือวิทยาศาสตร์



ทบทวนความรู้ก่อนเรียน



การวิเคราะห์และกำหนด
รายละเอียดของปัญหาต้อง
ดำเนินการก่อนวางแผนการ
แก้ปัญหา **ถูกต้องหรือไม่?**



ทบทวนความรู้ก่อนเรียน

การออกแบบวิธีการ
แก้ปัญหาโดยใช้รหัสจำลองหรือ
ผังงาน ทำให้นำไปปฏิบัติได้ง่าย
ถูกต้องหรือไม่?



ทบทวนความรู้ก่อนเรียน



การดำเนินการแก้ปัญหา
เป็นการนำวิธีการแก้ปัญหาที่ได้
ออกแบบไว้มาดำเนินการ โดยอาจ
พัฒนาเป็นโปรแกรมเพื่อแก้ปัญหา
ถูกต้องหรือไม่?

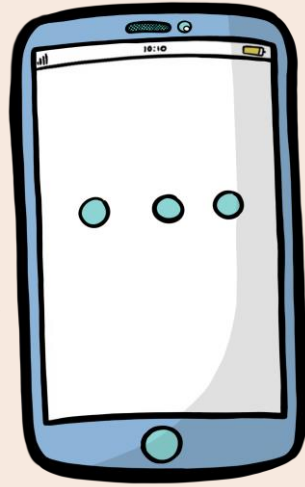


ภาษาไพทอน คือ

เป็นภาษาที่จัดอยู่ในกลุ่มภาษาระดับสูง
วัตถุประสงค์ของการสร้างภาษาไพทอนขึ้นมาก็คือ
เป็นการนำข้อดีของภาษาต่าง ๆ มารวมไว้ด้วยกัน
มีไวยากรณ์ที่ช่วยให้การเขียนโค้ดสั้น
กว่าภาษาอื่น ๆ ทำให้เราจดจำคำสั่งต่าง ๆ



ภาษาไพทอน ใช้ทำอะไรได้บ้าง



สร้างเกมส์



เว็บแอปพลิเคชัน



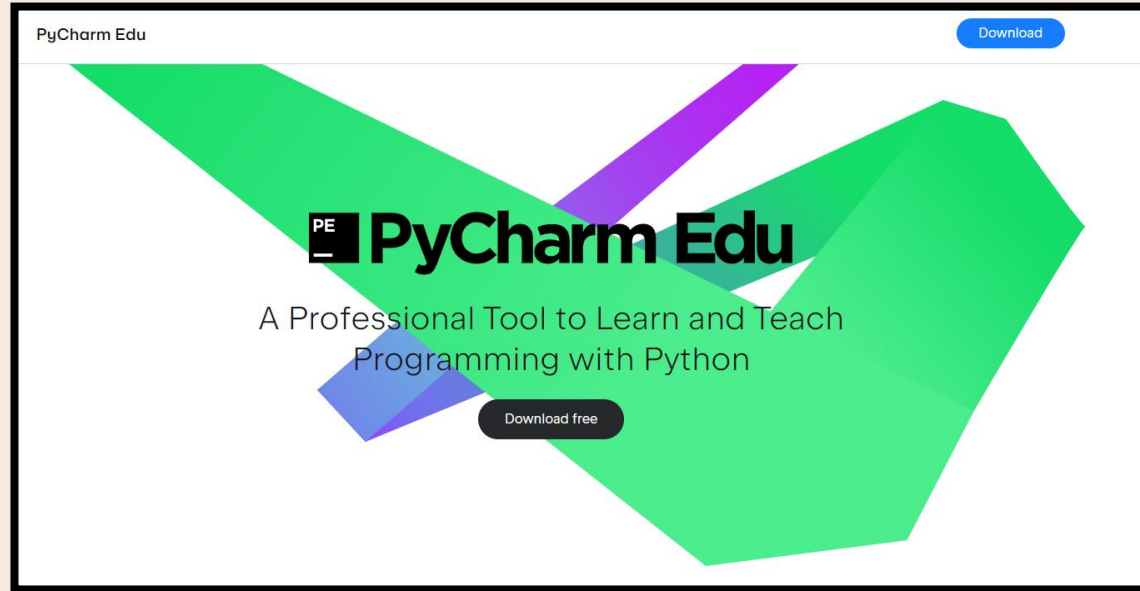
ไพทอน IDE จะทำงานตามคำสั่งได้ 2 โหมด คือ

โหมดอิมมีเดียท เป็นโหมดที่ผู้ใช้พิมพ์คำสั่งลงไป
ทีละคำสั่ง และตัวแปลภาษาไพทอนก็จะแปลคำสั่งนั้น
ในกรณีที่ผู้ใช้เขียนคำสั่งต่าง ๆ ได้ถูกต้อง โปรแกรมก็จะ
แสดงผลลัพธ์ทันที แต่ถ้าหากผู้ใช้เขียนคำสั่งผิด โปรแกรม
ก็จะขึ้นแจ้งเตือนผู้ใช้งาน



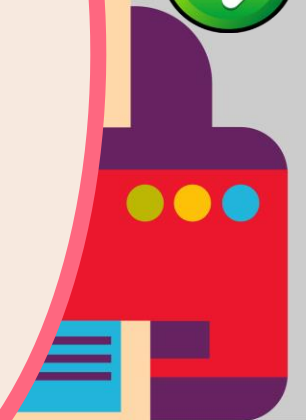
3.1.1 การติดตั้ง ไพทอนไอดีอี (ออฟไลน์)

ในปัจจุบันมีไพทอนไอดีอีให้เลือกใช้จำนวนมาก ในที่นี้เราจะโหลด PyCharm Edu ซึ่งเป็นไพทอนไอดีอีที่สามารถดาวน์โหลดมาใช้ในการเรียนรู้โดยไม่มีค่าใช้จ่าย

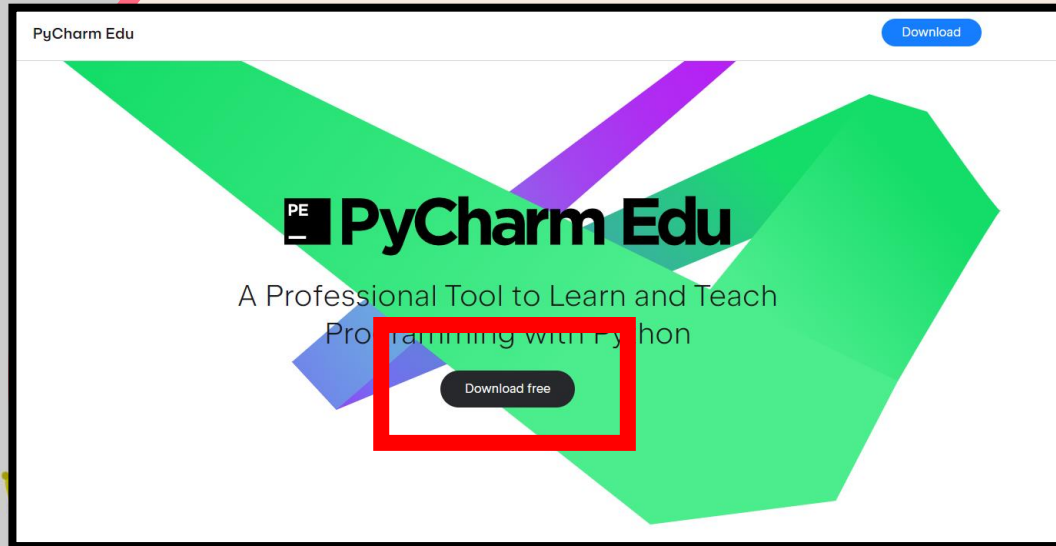


จากเว็บไซต์

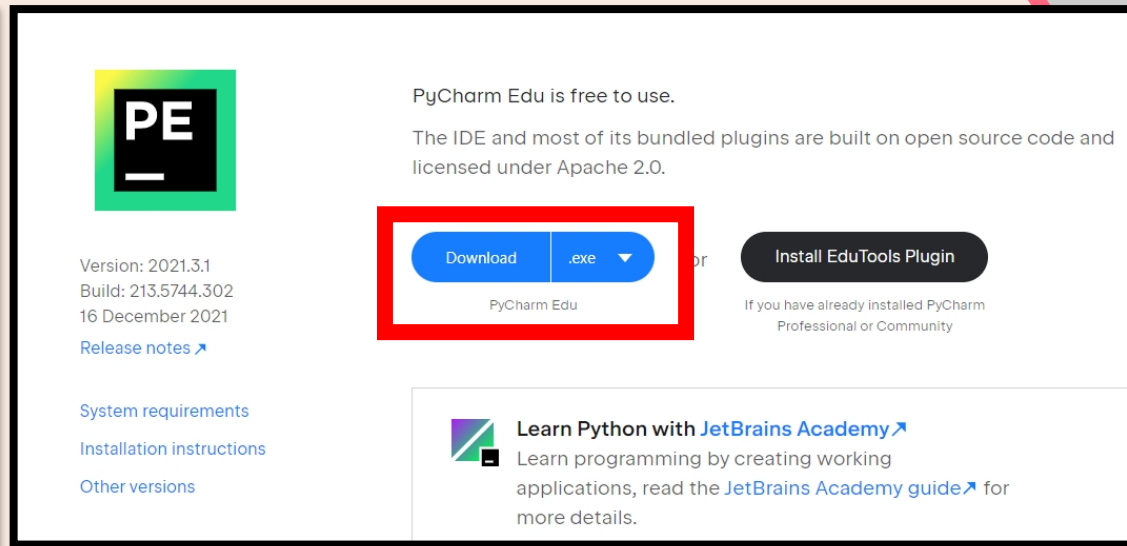
<https://www.jetbrains.com/pycharm-edu/>



การติดตั้งไพทอนไอดีอี (ออฟไลน์)



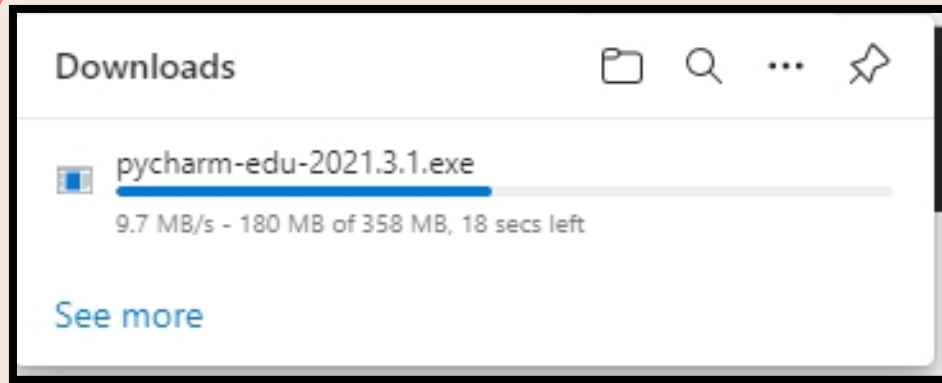
1. คลิก Download Free



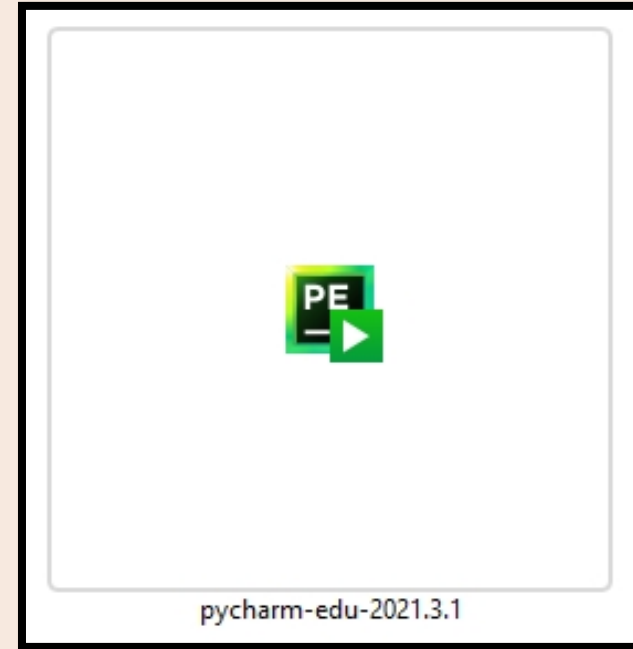
2. คลิก Download
ไฟล์.exe



การติดตั้งไพทอนไอดีอี (ออฟไลน์)



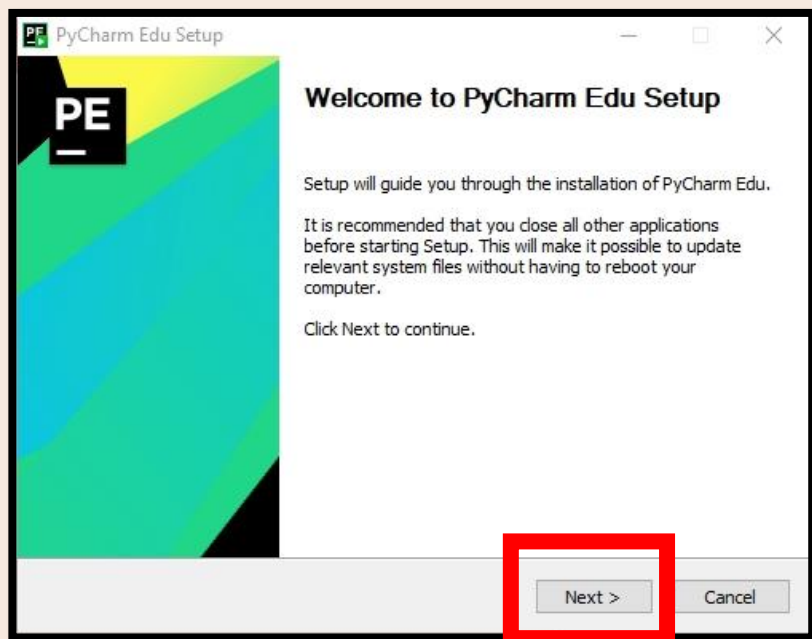
3. รอดาว์นโหลดไฟล์
ลงคอมพิวเตอร์



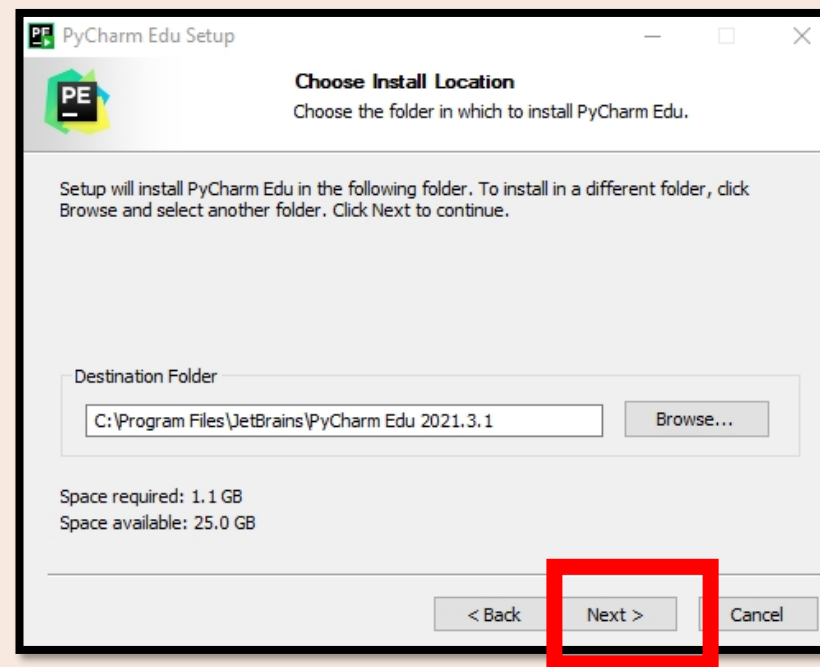
4. เข้า Folder Download
เพื่อดับเบิลคลิกไฟล์
ติดตั้งโปรแกรม



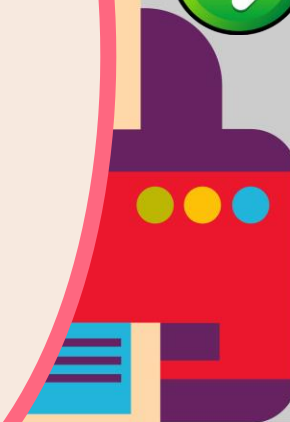
การติดตั้งไพทอนไอดีอี (ออฟไลน์)



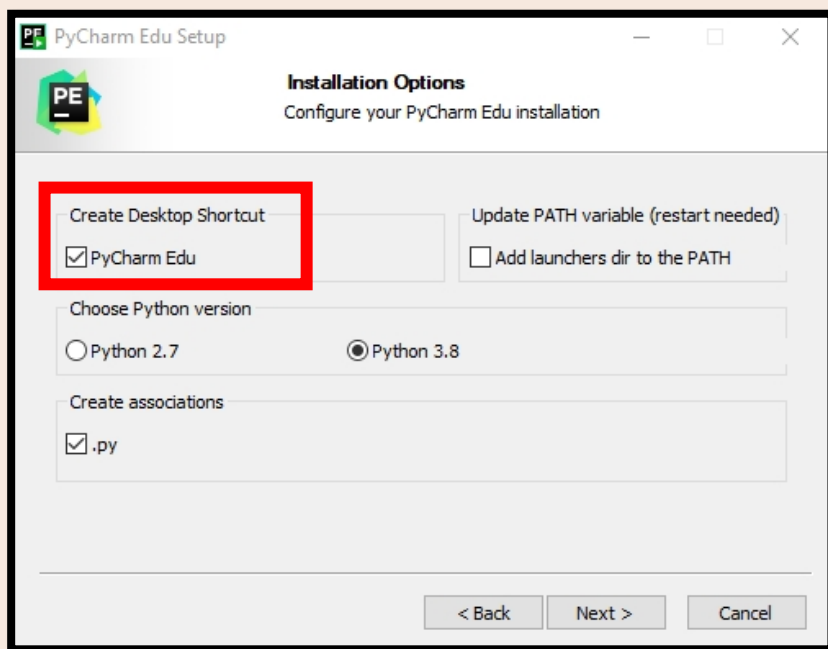
5. กด Next
เพื่อติดตั้งโปรแกรม



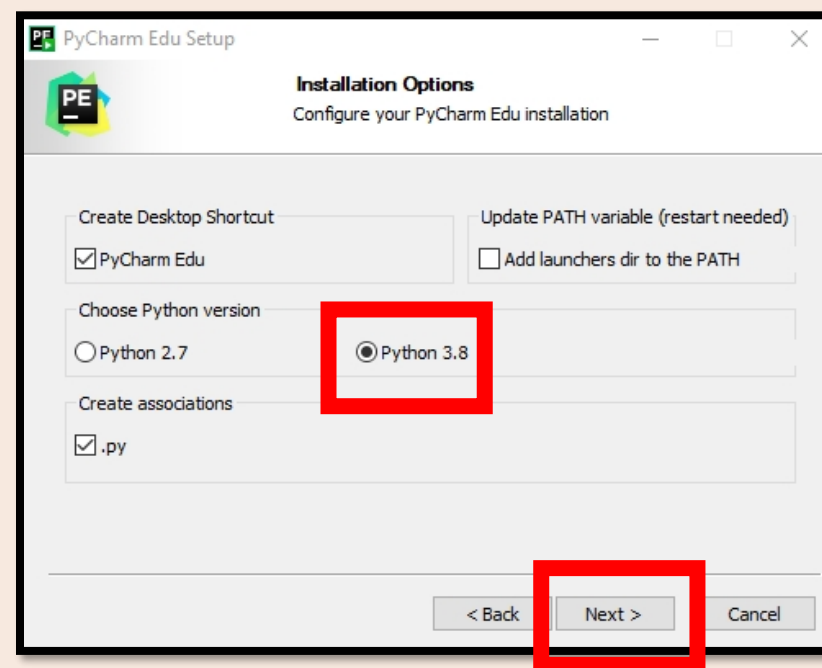
6. เลือกที่จัดเก็บไฟล์
แล้วกด Next



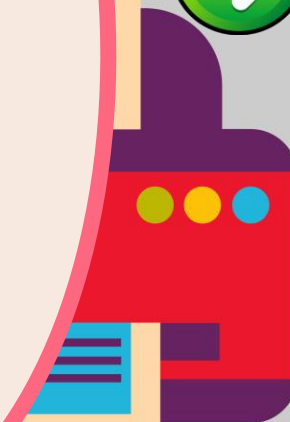
การติดตั้งไพทอนไอดีอี (ออฟไลน์)



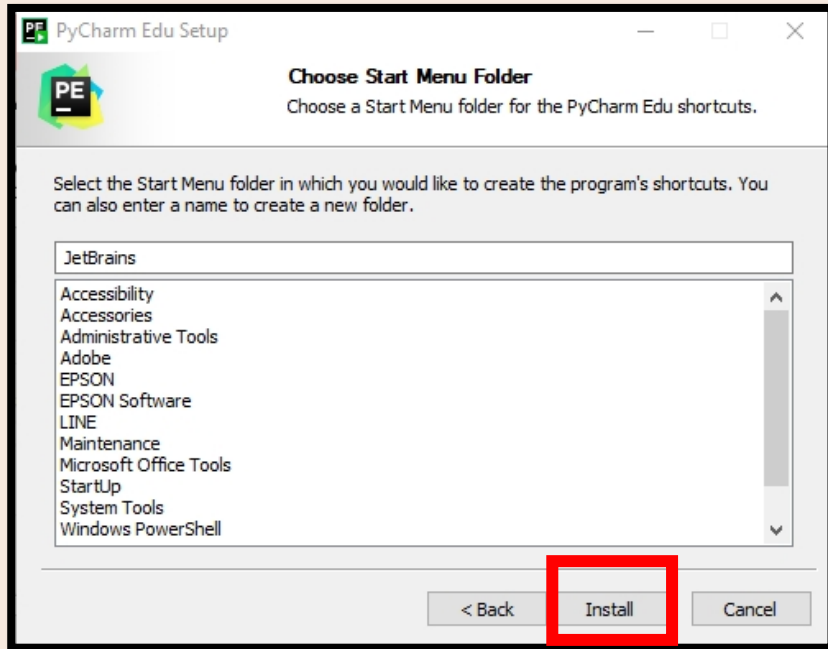
7. เลือกสร้างปุ่มเข้าโปรแกรม
ไว้หน้า Desktop



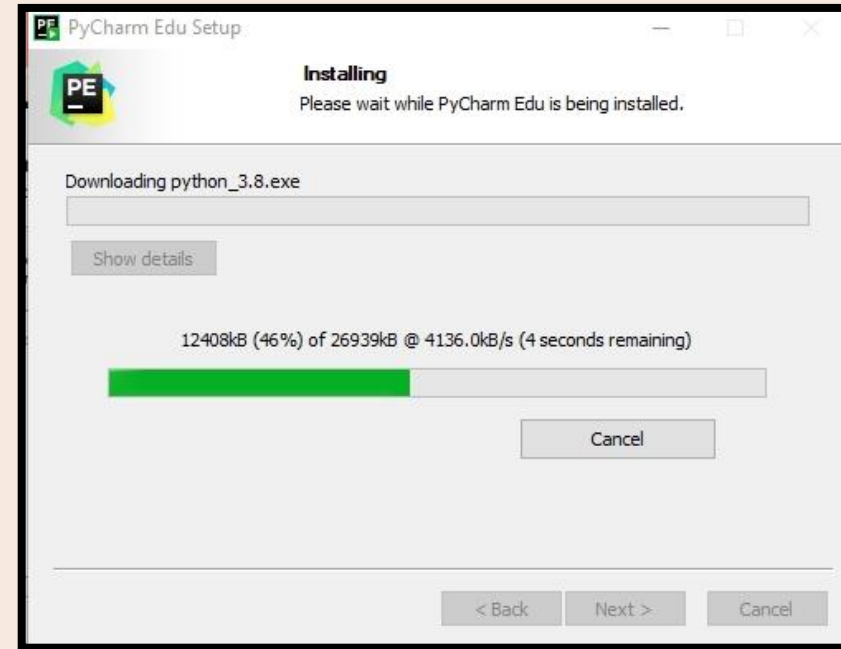
8. เลือก Version ล่าสุด
แล้วกด Next



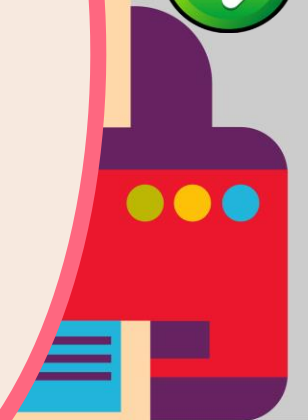
การติดตั้งไพทอนไอดีอี (ออฟไลน์)



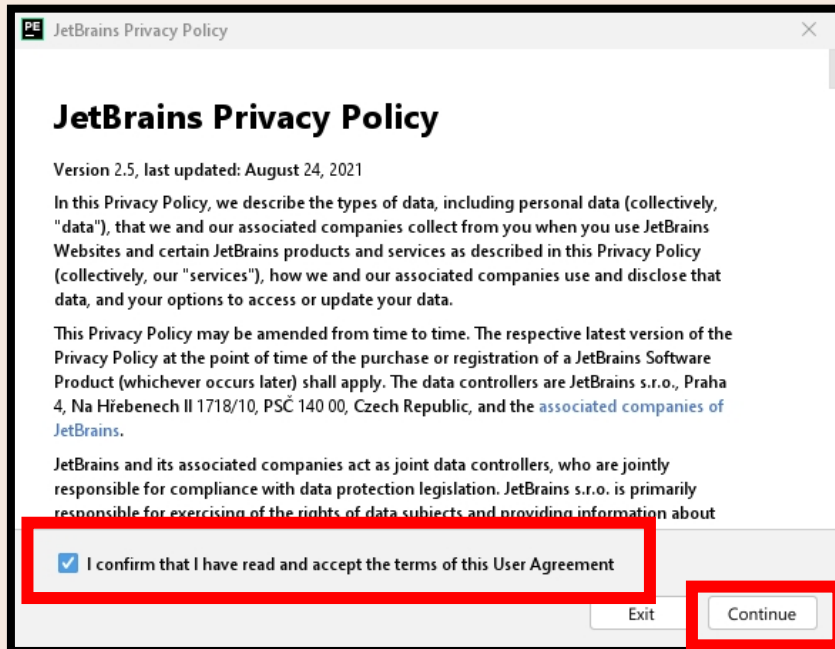
9. กด Install
เพื่อติดตั้งโปรแกรม



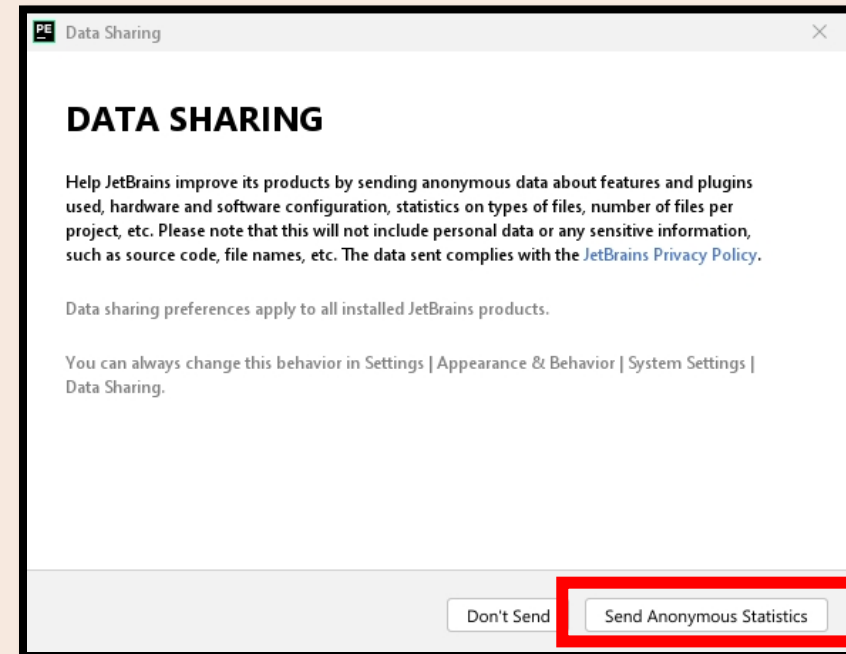
10. รอการติดตั้งจนกว่า
จะเสร็จสมบูรณ์



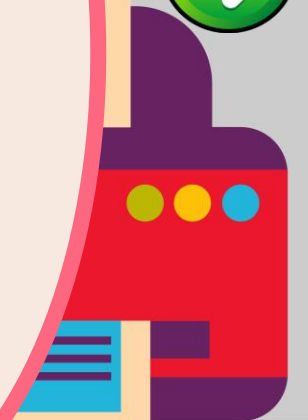
การติดตั้งไพทอนไอดีอี (ออฟไลน์)



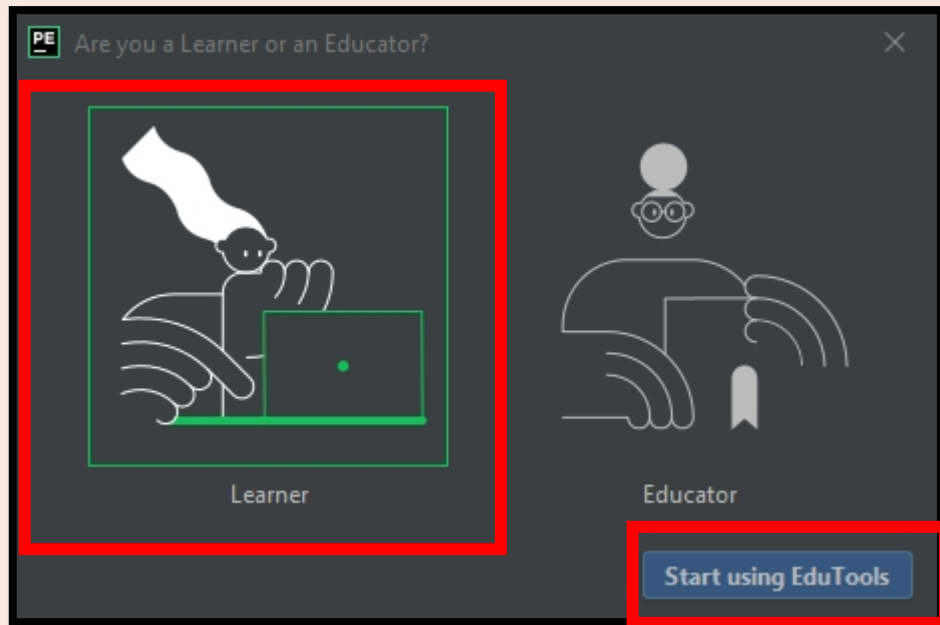
11. เมื่อติดตั้งเสร็จ เปิดโปรแกรมขึ้นมา
จะพบหน้าต่างตกลง ให้เลือก I Confirm
และ Continue



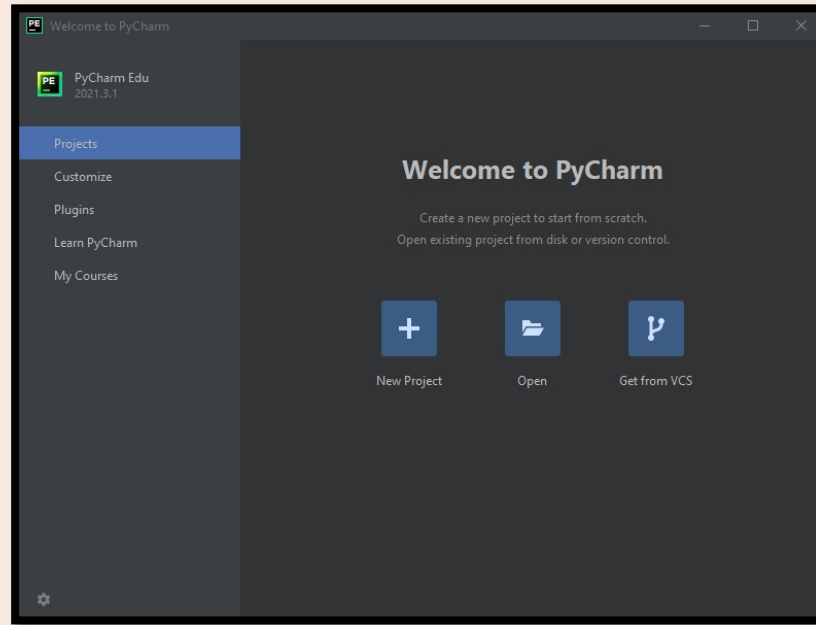
12. คลิก Send Anonymous
Statistics



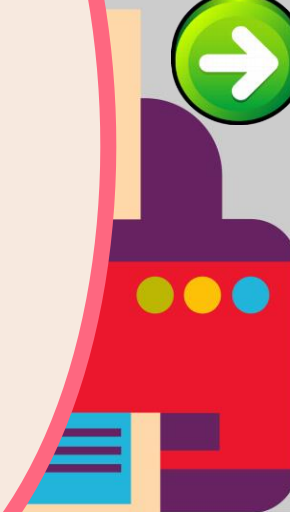
การติดตั้งไพทอนไอดีอี (ออฟไลน์)



13. เลือก Learner สำหรับการเรียนรู้
แล้วคลิก Start using EduTools

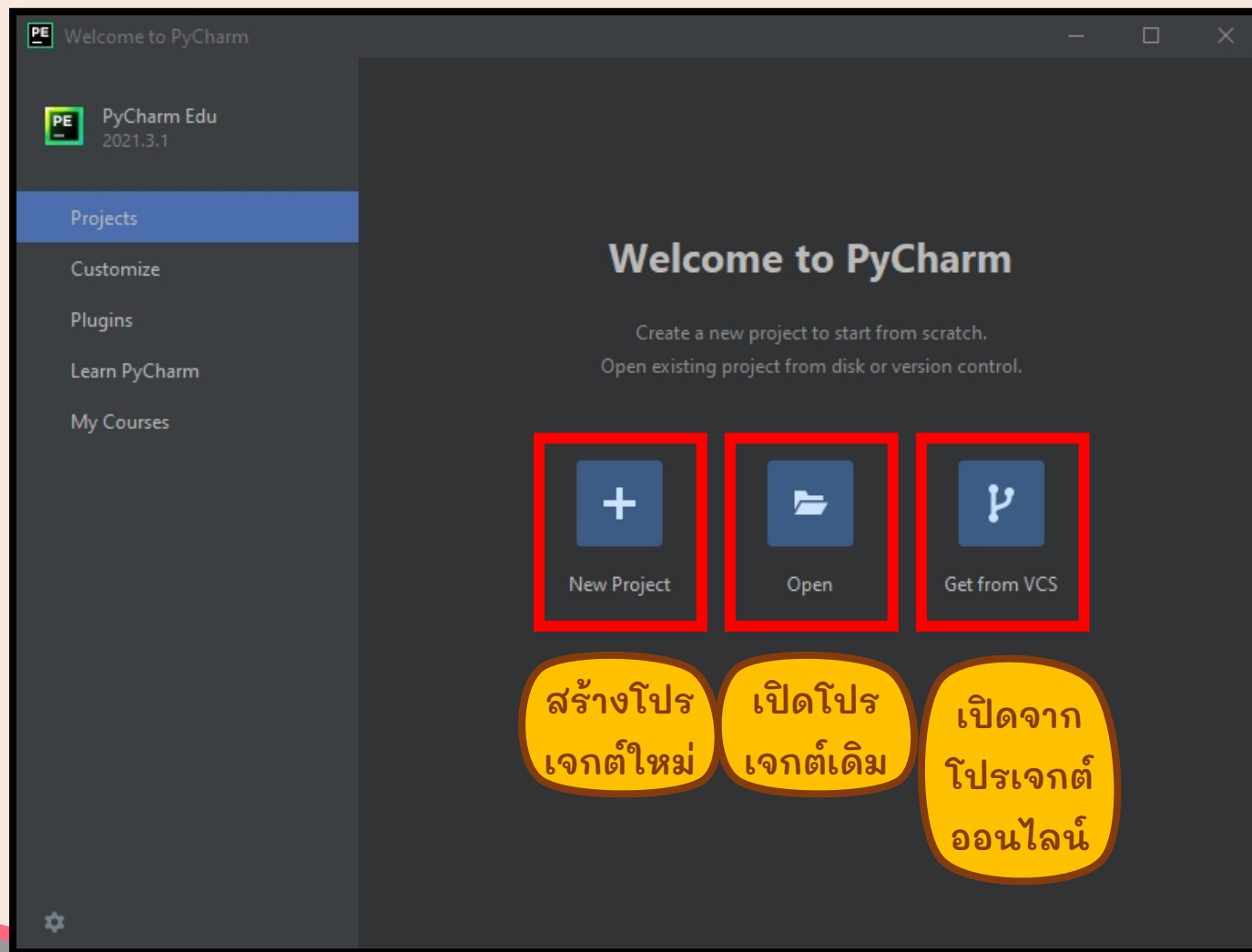


14. หน้าตาของโปรแกรม
เขียนภาษาไพทอน
แบบออฟไลน์



17

ส่วนประกอบโปรแกรม ไพทอนไอดีอี (ออฟไลน์)



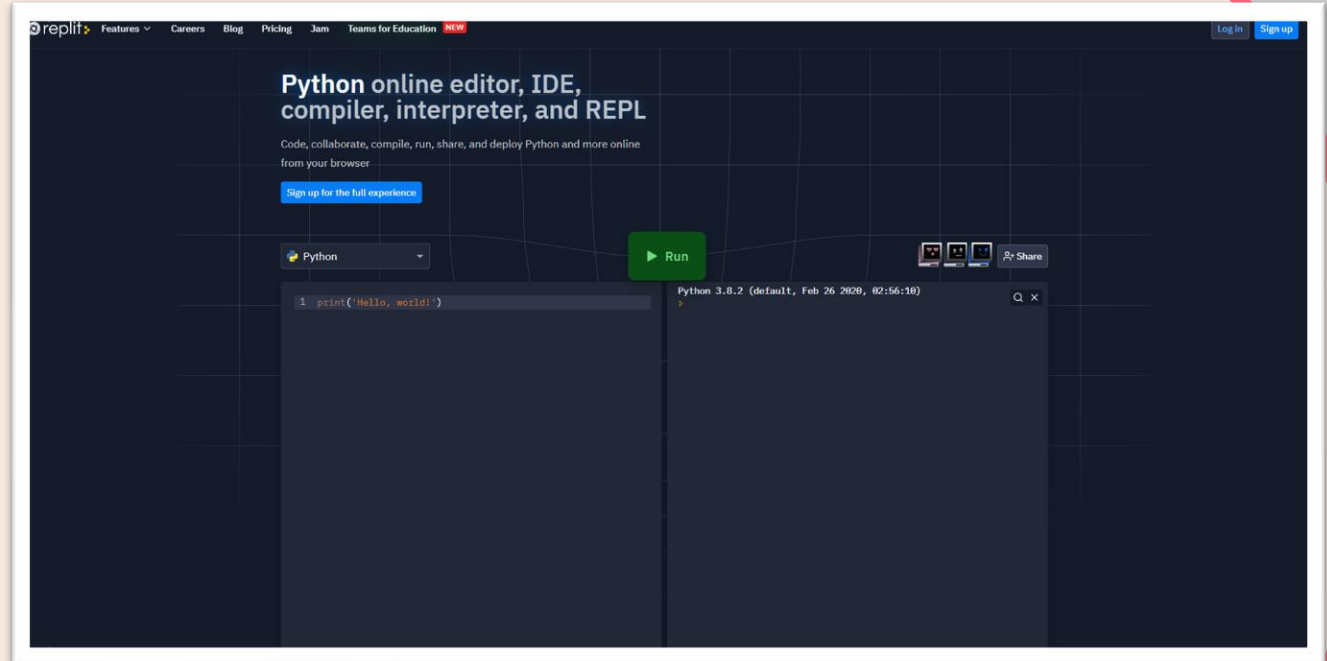
สร้างโปร
เจกต์ใหม่

เปิดโปร
เจกต์เดิม

เปิดจาก
โปรเจกต์
ออนไลน์

โปรแกรมไพทอนออนไลน์

หากคอมพิวเตอร์ที่ใช้
งานอยู่เชื่อมต่อ สามารถเข้า
เขียนโปรแกรมภาษาไพทอน
แบบออนไลน์ได้จากเว็บไซต์
ที่ให้บริการตัวแปลภาษา
ไพทอน



จากเว็บไซต์

<https://replit.com/languages/python3>

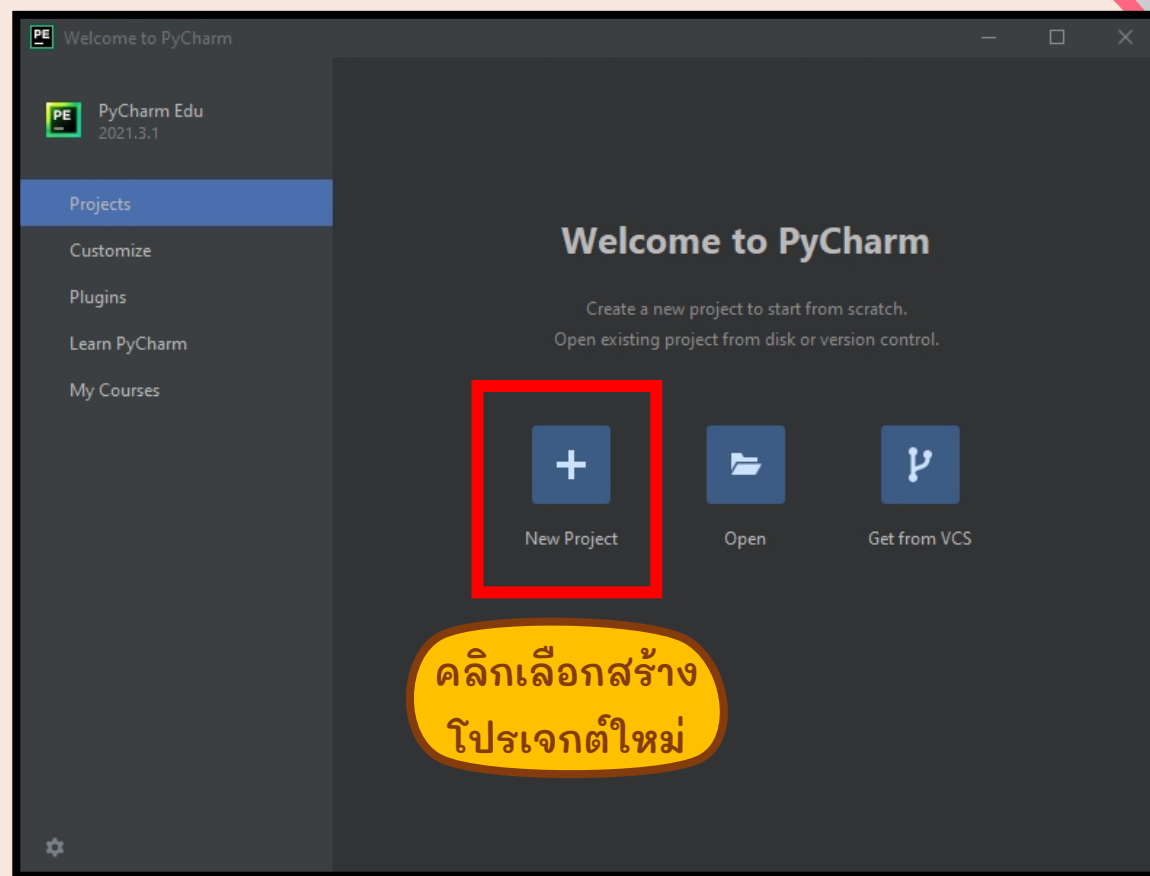
3.1.2 เริ่มต้นเขียน โปรแกรมภาษาไพทอน

คำสั่งเบื้องต้นของภาษา python ที่จะ
เรียนรู้คือคำสั่งที่ใช้แสดงผลทางจอภาพ
และคำสั่งที่ใช้ในการรับข้อมูลเข้าจากผู้ใช้
ผ่านทางคีย์บอร์ด ในขั้นเริ่มต้นนี้จะใช้คำสั่ง
python ในโหมดอิมมีเดียท ดังตัวอย่าง
ต่อไปนี้

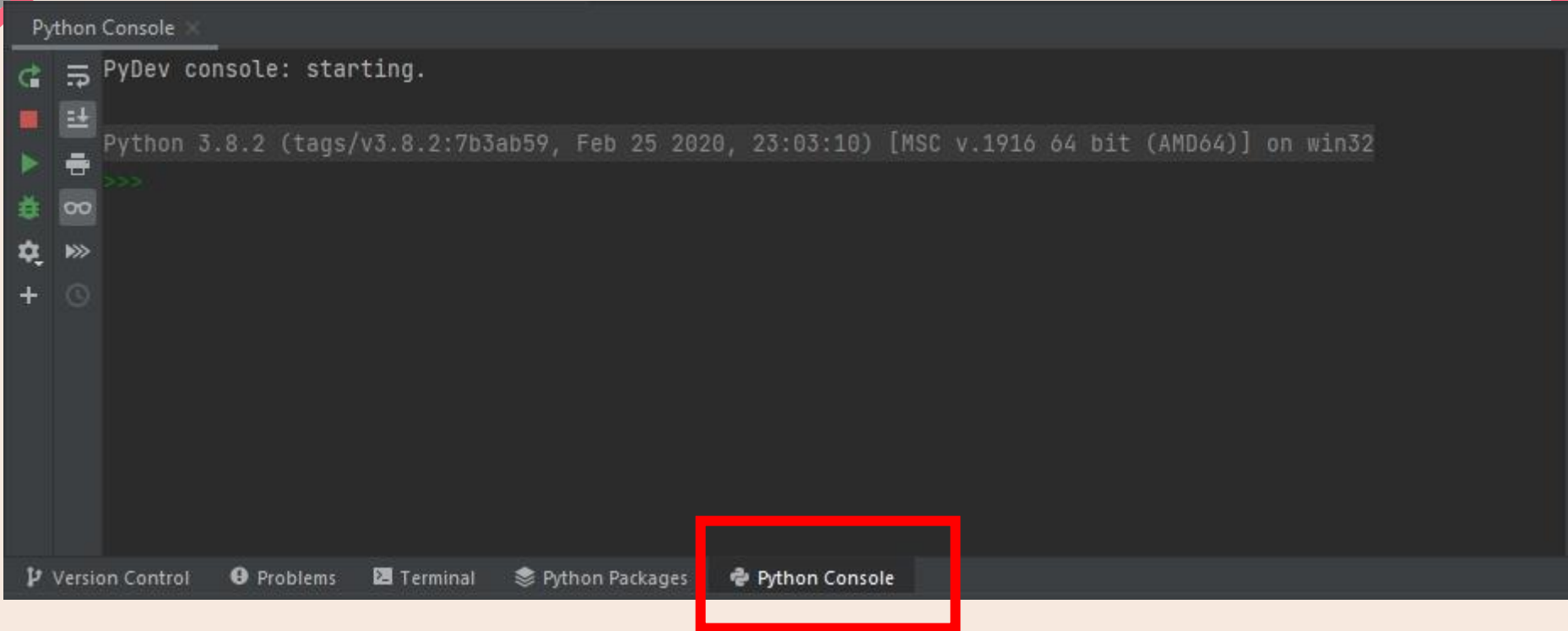


3.1.2 เริ่มต้นเขียน โปรแกรมภาษาไพทอน

เปิดโปรแกรม
PyCharm หรือ โปรแกรม
เขียนภาษาไพทอน
ออนไลน์ เพื่อทำการเขียน
คำสั่งแสดงผล



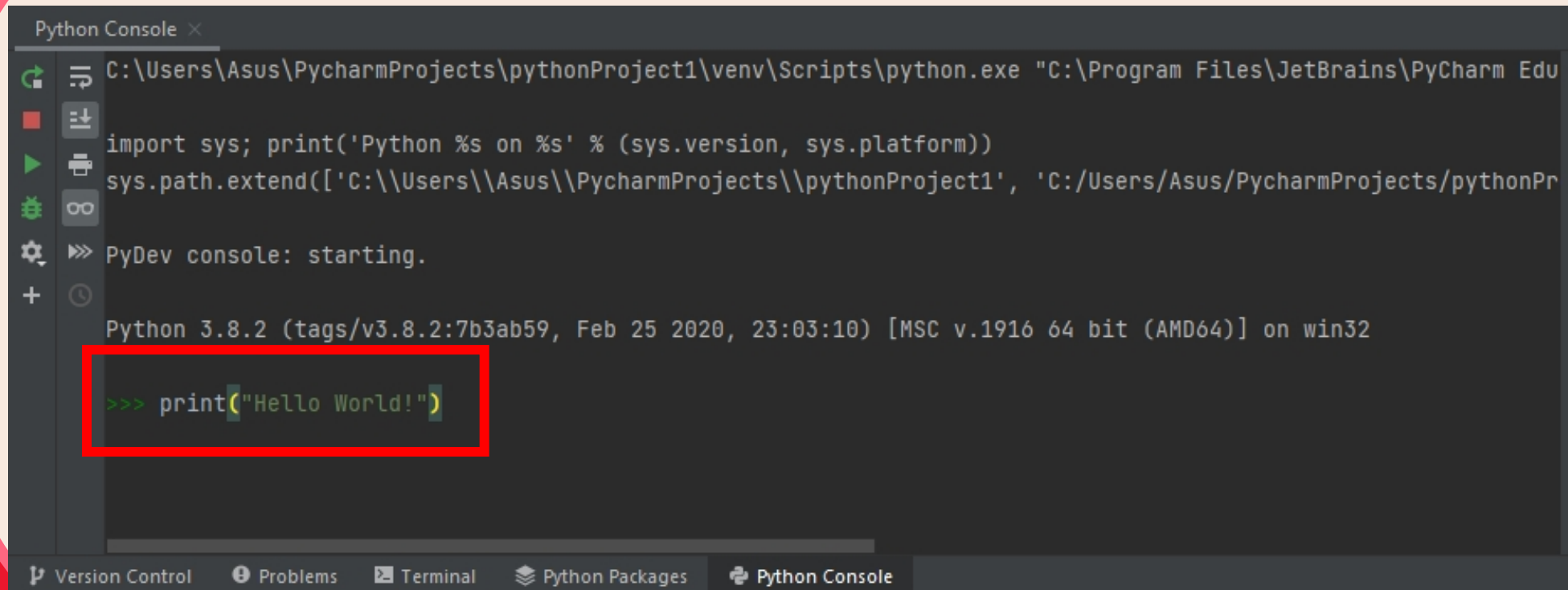
ตัวอย่างที่ 3.1 คำสั่งแสดงผล ในโปรแกรมภาษาไพทอน



1.คลิกเลือก Python Console

ตัวอย่างที่ 3.1 คำสั่งแสดงผล ในโปรแกรมภาษาไพทอน

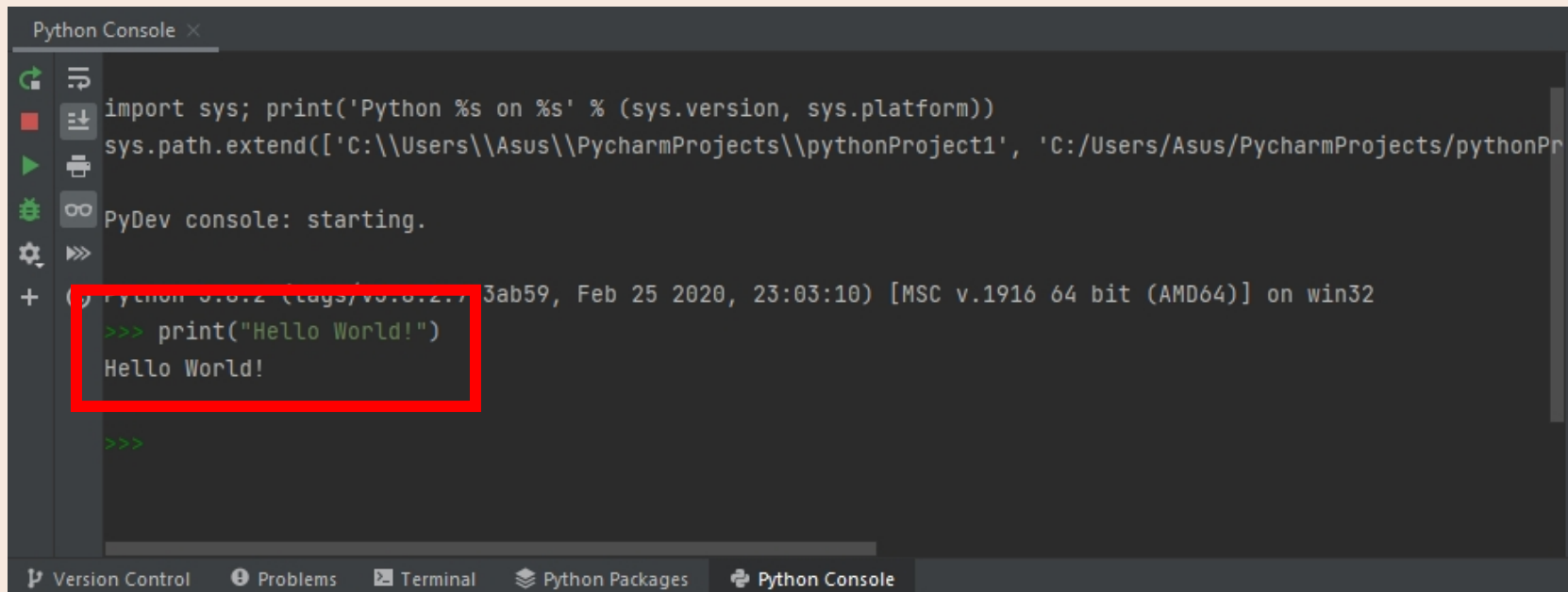
2. ทดลองพิมพ์คำสั่งต่อไปนี้ในคอนโซล แล้วกดแป้น Enter



```
Python Console x
C:\Users\Asus\PycharmProjects\pythonProject1\venv\Scripts\python.exe "C:\Program Files\JetBrains\PyCharm Edu
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['C:\\Users\\Asus\\PycharmProjects\\pythonProject1', 'C:/Users/Asus/PycharmProjects/pythonPr
PyDev console: starting.
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
>>> print("Hello World!")
```

ตัวอย่างที่ 3.1 คำสั่งแสดงผล ในโปรแกรมภาษาไพทอน

3. ผลลัพธ์ที่ได้คือ



The screenshot shows a Python Console window with the following content:

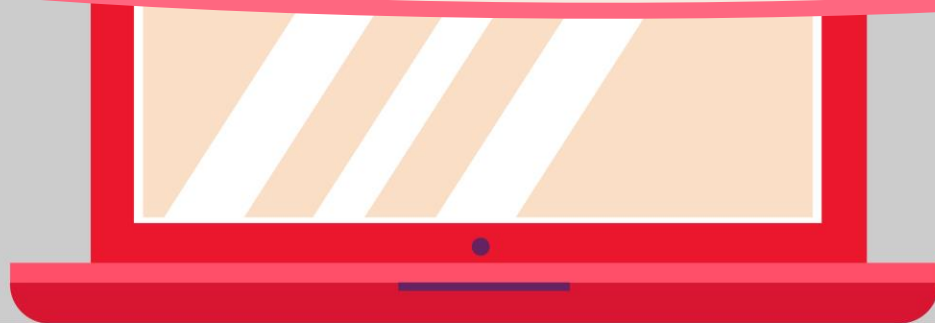
```
Python Console X
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['C:\\Users\\Asus\\PycharmProjects\\pythonProject1', 'C:/Users/Asus/PycharmProjects/pythonPr
PyDev console: starting.
Python 3.8.2 (tags/v3.8.2:73ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
>>> print("Hello World!")
Hello World!
>>>
```

The output `Hello World!` is highlighted with a red box.

จากตัวอย่างที่ 3.1 อธิบายได้ว่า `print()` เป็นคำสั่งชนิดฟังก์ชัน (function) ทำหน้าที่แสดงสิ่งที่อยู่ภายในเครื่องหมายวงเล็บ `()` ออกทางจอภาพ ให้สังเกตผลลัพธ์ที่ได้ว่า ไม่มีเครื่องหมาย “”

เกร็ดน่ารู้

รูปแบบการเขียนฟังก์ชัน คำสั่งชนิดฟังก์ชันใน python เช่น `print ()` และ `input ()` จะมีเครื่องหมายวงเล็บต่อท้ายเสมอ โดยทั่วไปฟังก์ชันจะนำข้อมูลที่ผู้ใช้กำหนดในวงเล็บเข้าไปประมวลผล ตามที่ function ได้ถูกออกแบบมา ชนิดและจำนวนของข้อมูลที่ผู้ใช้ต้องกำหนดในแต่ละฟังก์ชันจะแตกต่างกันออกไป



กิจกรรม

ให้นักเรียนพิมพ์คำสั่งต่อไปนี้ลงในคอลโซล แล้ว
สังเกตผลลัพธ์ที่ได้ว่าเหมือนหรือแตกต่างกันอย่างไร และ
เพราะเหตุใด

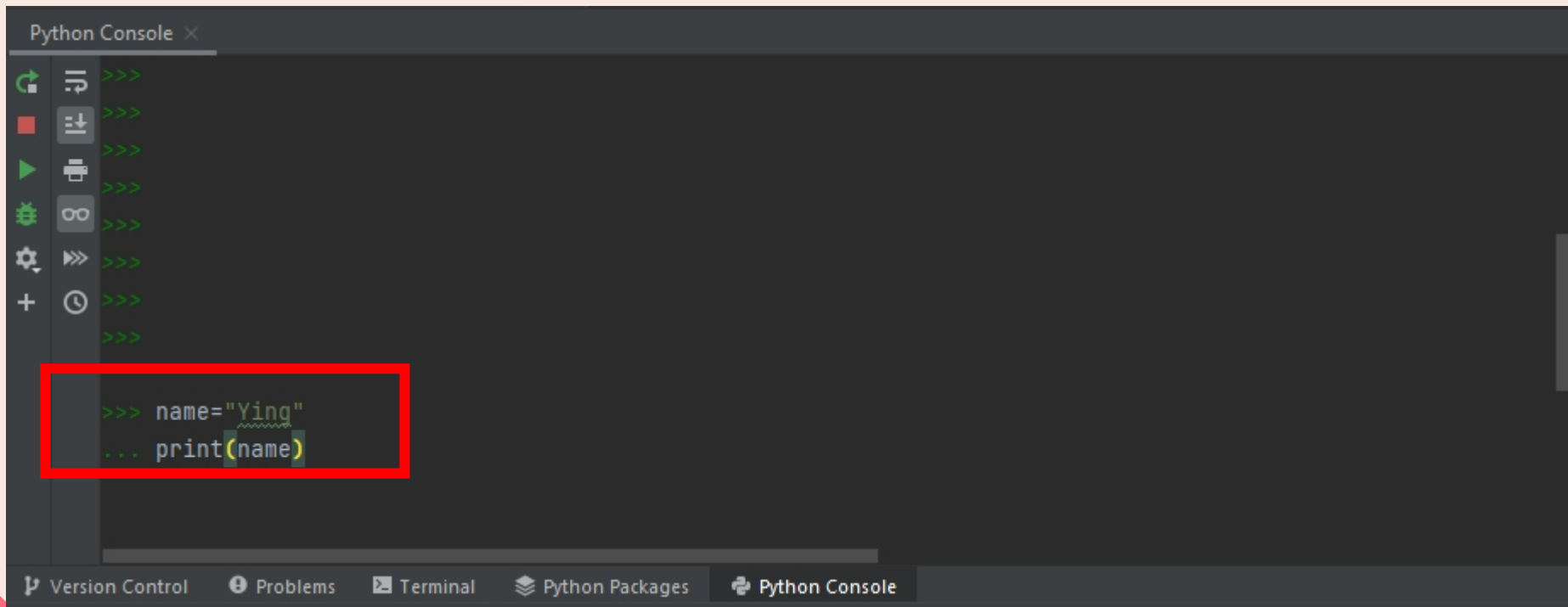
```
Print ("3+5")
```

```
Print (3+5)
```



ตัวอย่างที่ 3.2 คำสั่งรับข้อมูลเข้า

ทดลองพิมพ์คำสั่งต่อไปนี้ในคอนโซล แล้วกดแป้น Enter

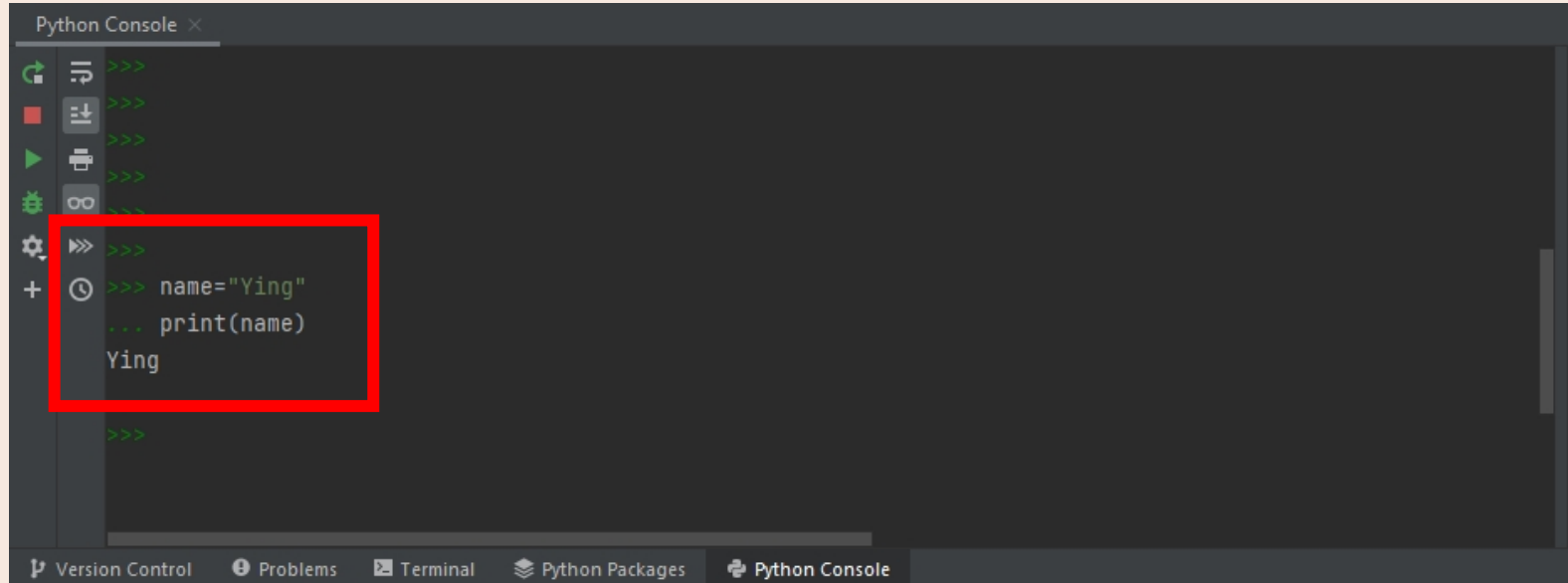
A screenshot of a Python Console window. The window has a dark background with a sidebar on the left containing various icons. The main area shows a Python prompt '>>>' followed by two lines of code: 'name="Ying"' and '... print(name)'. These two lines are enclosed in a red rectangular box. The bottom of the window has a tab bar with 'Python Console' selected.

```
>>> name="Ying"
... print(name)
```



ตัวอย่างที่ 3.2 คำสั่งรับข้อมูลเข้า

ผลลัพธ์ที่ได้คือ



```
Python Console x
>>>
>>>
>>>
>>>
>>>
>>> name="Ying"
... print(name)
Ying
>>>
```

The screenshot shows a Python Console window with a dark background. The code entered is `name="Ying"` followed by `print(name)`. The output `Ying` is displayed below the code. A red rectangle highlights the code and its output. The console window has a sidebar on the left with various icons and a bottom bar with tabs for Version Control, Problems, Terminal, Python Packages, and Python Console.

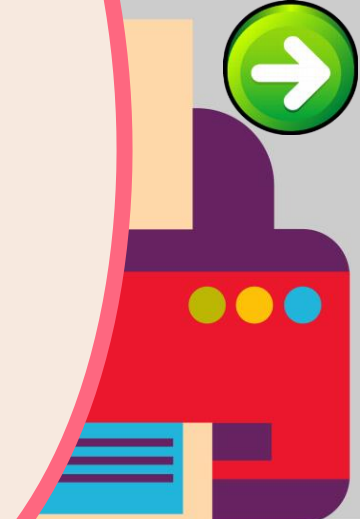
จากตัวอย่างที่ 3.2 อธิบายได้ว่า บรรทัดแรกเป็นการกำหนดค่าให้ตัวแปร `name` มีค่าเป็น “Ying” หลังจากนั้นบรรทัดที่ 2 จะแสดงค่าในตัวแปร `name` ออกมาทางจอภาพ ซึ่งก็คือคำว่า Ying

ตัวอย่างที่ 3.2 คำสั่งรับข้อมูลเข้า

ทดลองพิมพ์คำสั่งต่อไปนี้ในคอนโซล แล้วกดแป้น Enter

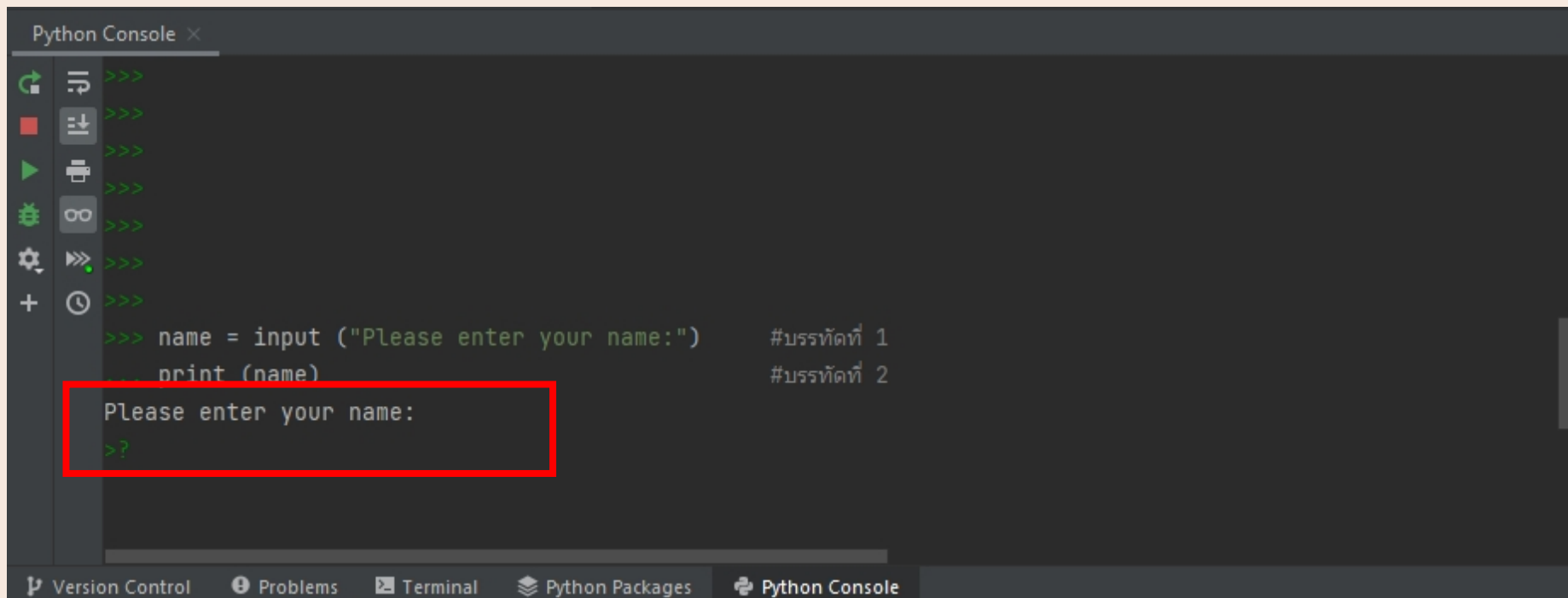
[illegible]

หมายเหตุ : ไพทอนจะใช้สัญลักษณ์ # แสดงจุดเริ่มต้นของคอมเมนต์ (comment) ในแต่ละบรรทัด



ตัวอย่างที่ 3.2 คำสั่งรับข้อมูลเข้า

ผลลัพธ์ที่ได้คือ



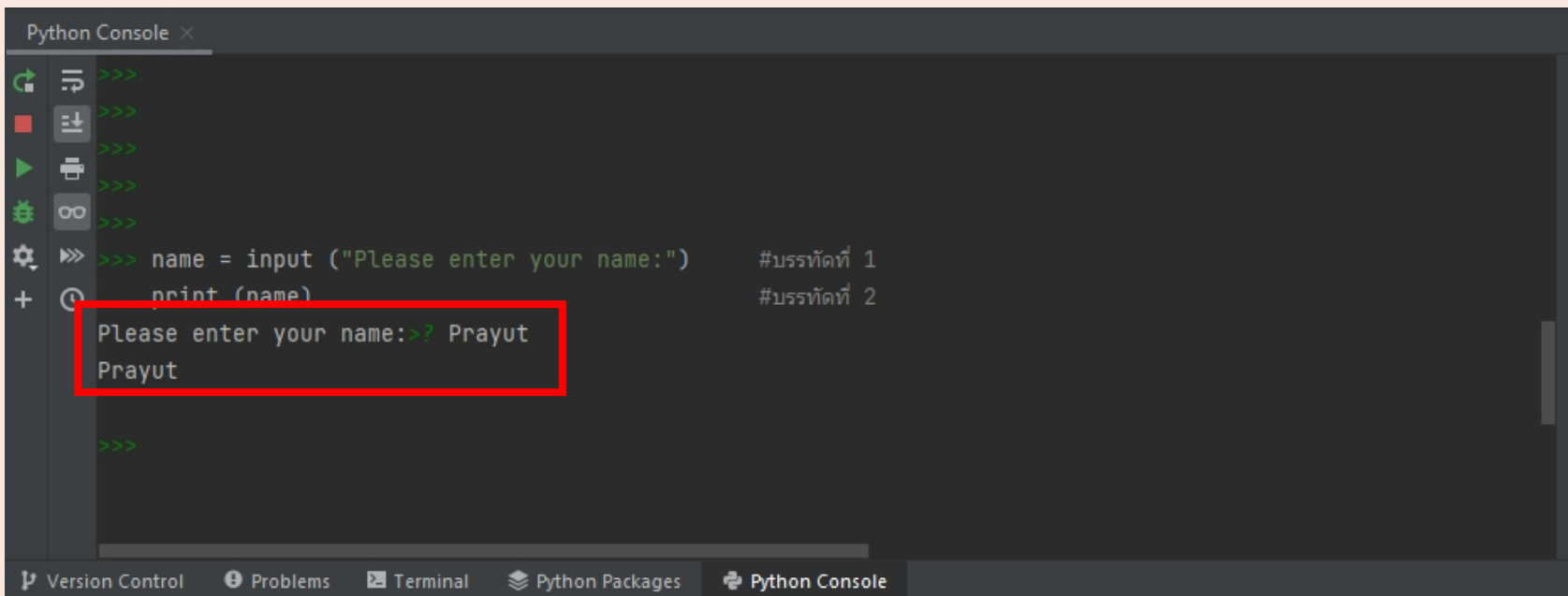
```
Python Console x
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>> name = input("Please enter your name:") #บรรทัดที่ 1
>>> print(name) #บรรทัดที่ 2
Please enter your name:
>?
```

>?_# เครื่องหมาย >? เป็นเครื่องหมายการรอให้ผู้ใช้ป้อน

ข้อมูลของ PyCharm Edu

ตัวอย่างที่ 3.2 คำสั่งรับข้อมูลเข้า

จากนั้นให้ผู้ใช้ป้อนชื่อตนเอง เช่น ป้อนชื่อว่า Prayut



```
Python Console x
>>>
>>>
>>>
>>>
>>> name = input("Please enter your name:") #บรรทัดที่ 1
>>> print(name) #บรรทัดที่ 2
Please enter your name:>? Prayut
Prayut
>>>
```

The screenshot shows a Python IDE with a console window. The code being executed is a simple input and print statement. The user has entered 'Prayut' in response to the prompt 'Please enter your name:'. The input and the resulting output are highlighted with a red rectangular box.



ตัวอย่างที่ 3.2 คำสั่งรับข้อมูลเข้า

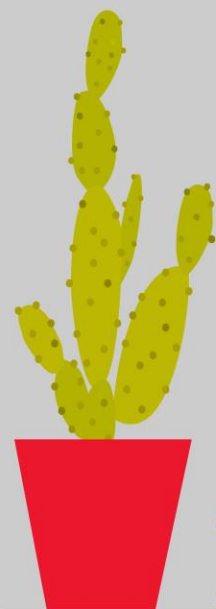
จากตัวอย่างที่ 3.2 `input()` เป็นคำสั่งชนิดฟังก์ชัน (function) ทำหน้าที่รับข้อมูลเข้าที่ผู้ใช้ป้อนผ่านคีย์บอร์ด แล้วส่งคืนสิ่งที่ผู้ใช้ป้อนเข้ามาเป็นข้อมูลชนิดสตริง ให้กับตัวแปรที่กำหนดไว้หน้าเครื่องหมาย = ในที่นี้คือตัวแปร `name`

หลังจากนั้นบรรทัดที่ 2 จะแสดงค่าในตัวแปร `name` ออกมาทางจอภาพ ซึ่งก็คือคำว่า `Prayut`



1

ตัวแปร



ตัวแปร (Variable)

ใช้ในการอ้างอิงค่าข้อมูล โดยตัวแปรจะถูกกำหนดค่าด้วยเครื่องหมาย = เช่น

`C = 16`

`name = "somchai"`

บรรทัดแรกเป็นการกำหนดให้ตัวแปรชื่อ c ชี้ไปยังจำนวนเต็ม 16 และบรรทัดที่ 2 ตัวแปรชื่อ name ชี้ไปยังสตริง somchai



เกร็ดน่ารู้

การตรวจสอบชนิดของข้อมูล

ตัวเลขจำนวนเต็ม 16 และสตริง "somchai" เป็นข้อมูลคนละชนิดกัน ในไพทอนตรวจสอบชนิดของข้อมูลด้วยฟังก์ชัน `type()` ทดลองพิมพ์คำสั่งต่อไปนี้ เพื่อตรวจสอบชนิดของข้อมูลในไพทอน

Type (16)

Type ('somchai')

Type ("somchai")

Type (3+5)

Type ("3+5")



ตัวแปร (Variable)

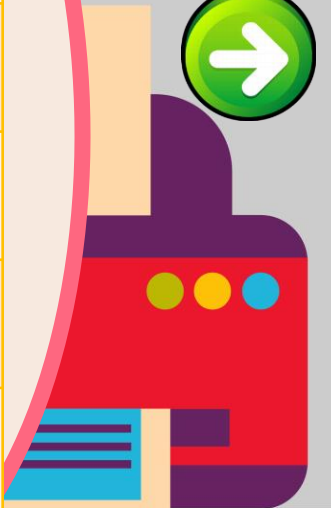
สำหรับการตั้งชื่อตัวแปรในไพทอน ชื่อตัวแปรประกอบไปด้วยตัวอักษร ตัวเลข หรือ เครื่องหมายขีดเส้นใต้เท่านั้น โดยต้องขึ้นต้นด้วยตัวอักษรหรือเครื่องหมายขีดเส้นใต้ นอกจากนี้ตัวอักษรภาษาอังกฤษตัวพิมพ์เล็กและตัวพิมพ์ใหญ่จะถือว่าแตกต่างกัน ดังนั้น ตัวแปรชื่อ count และ Count จึงเป็นตัวแปรคนละตัวกัน



ตัวแปร (Variable)

นอกจากนี้ชื่อตัวแปรที่ตั้งขึ้นจะต้องไม่ซ้ำกับคำหลัก (keyword) ที่ไพทอนใช้เป็นคำสั่ง โดยคำหลักมีดังต่อไปนี้

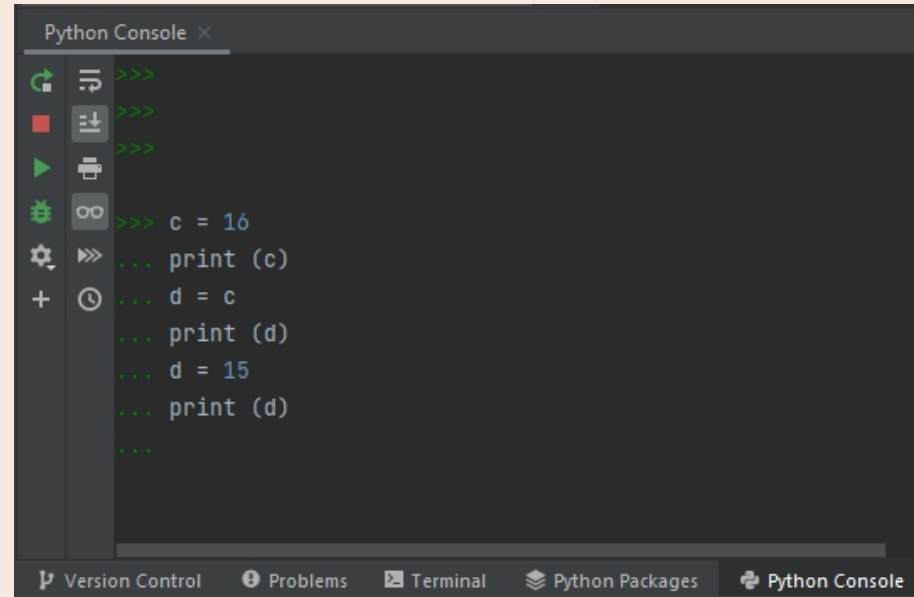
and	as	assert	break	class	continue	def
del	elif	else	except	exec	finally	for
from	global	if	import	in	is	lambda
nonlocal	not	or	pass	raise	return	try
while	with	yield	True	False	None	



ตัวอย่างที่ 3.3 การเปลี่ยนค่าของตัวแปร

พิมพ์คำสั่งต่อไปนี้ในคอนโซล

```
c = 16  
print (c)  
d = c  
print (d)  
d = 15  
print (d)
```



```
Python Console x  
>>>  
>>>  
>>>  
>>> c = 16  
>>> print (c)  
16  
>>> d = c  
>>> print (d)  
16  
>>> d = 15  
>>> print (d)  
15  
>>>
```

ตัวอย่างที่ 3.3 การเปลี่ยนค่าของตัวแปร

จะได้ผลลัพธ์ดังนี้

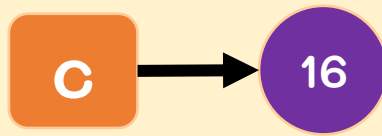
```
Python Console x
... print (c)
... d = c
... print (d)
... d = 15
... print (d)
...
16
16
15
>>>
```

จากตัวอย่างที่ 3.3 อธิบายได้ดังนี้

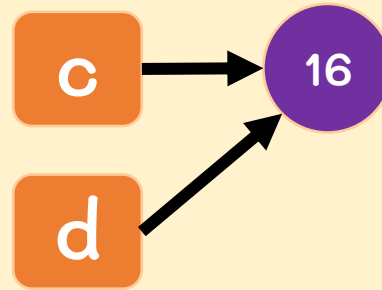
1. $c = 16$ เป็นคำสั่งกำหนดให้ตัวแปร c ชี้ไปจำนวนเต็ม 16 เมื่อแสดงผลด้วยคำสั่ง `print ()` จึงได้ผลลัพธ์เป็น 16
2. $d = c$ เป็นคำสั่งสร้างตัวแปร d แล้วชี้ไปที่เดียวกับที่ตัวแปร c ชี้อยู่ จึงทำให้ตัวแปร d ชี้ไปยังจำนวนเต็ม 16 เมื่อพิมพ์ด้วยคำสั่ง `print (d)` จึงได้ผลลัพธ์เป็น 16
3. $d = 15$ เป็นคำสั่งกำหนดให้ตัวแปร d ชี้ไปที่จำนวนเต็ม 15 เมื่อแสดงผลด้วยคำสั่ง `print (d)` จึงได้ผลลัพธ์เป็นค่า 15

ตัวอย่างที่ 3.3 การเปลี่ยนค่าของตัวแปร

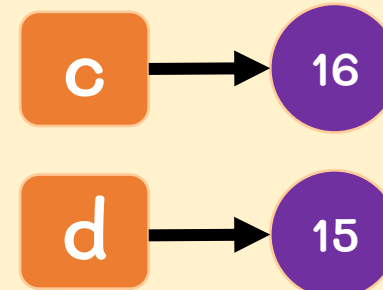
การเปลี่ยนแปลงค่าของตัวแปร c และ d ที่เกิดขึ้นตามคำสั่งใน
ตัวอย่างที่ 3.3 แสดงรูปดังนี้



$c = 16$



$d = c$



$d = 15$



1

ชนิดข้อมูลพื้นฐาน



ชนิดข้อมูลพื้นฐาน

โปรแกรมภาษาไพทอนมีการแบ่งประเภทของข้อมูลออกเป็นหลายประเภท โดยมีประเภทข้อมูลพื้นฐาน คือ

- ข้อมูลประเภทข้อความ (String data type)
- ข้อมูลประเภทจำนวน (numerical data type)

ข้อมูลประเภทข้อความ (String data type)

การกำหนดเครื่องหมายอัญประกาศรอบข้อความที่ต้องการกำหนด ข้อความที่เป็นข้อความหรือสตริง ให้ใช้โดยเลือกใช้ได้ทั้งอัญประกาศเดี่ยว (') หรือคู่ (") ตามความเหมาะสม หรือกรณีที่มีข้อความยาวหลายบรรทัด ต้องใช้เครื่องหมาย “ หรือ ‘ ติดต่อกัน 3 ตัว รอบหน้าและหลังข้อความ ดังตัวอย่างต่อไปนี้



ตัวอย่างที่ 3.4

การกำหนดข้อมูลชนิดข้อความ

พิมพ์คำสั่งต่อไปนี้ในคอนโซล

```
>>> organization = "สวท"
>>> address = "924 ถนนสุขุมวิท เขตคลองเตย กรุงเทพฯ 10110"
>>> print (organization)
>>> print (address)
>>> address = "สสวท"
...921 ถนนสุขุมวิท
... เขตคลองเตย กรุงเทพฯ
...10110"
>>> print (address)
```

บรรทัดที่ 1
#บรรทัดที่ 2
#บรรทัดที่ 3
#บรรทัดที่ 3
#บรรทัดที่ 5

#บรรทัดที่ 6

```
>>> organization = “ส่วท”
```

บรรทัดที่ 1

```
>>> address = “924 ถนนสุขุมวิท เขตคลองเตย กรุงเทพฯ 10110”
```

#บรรทัดที่ 2

```
>>> print (organization)
```

#บรรทัดที่ 3

```
“ส่วท”
```

```
>>> print (address)
```

#บรรทัดที่ 4

```
924 ถนนสุขุมวิท เขตคลองเตย กรุงเทพฯ 10110
```

```
>>> address = “““ส่วท”
```

#บรรทัดที่ 5

```
...921 ถนนสุขุมวิท
```

```
... เขตคลองเตย กรุงเทพฯ
```

```
...10110””
```

```
>>> print (address)
```

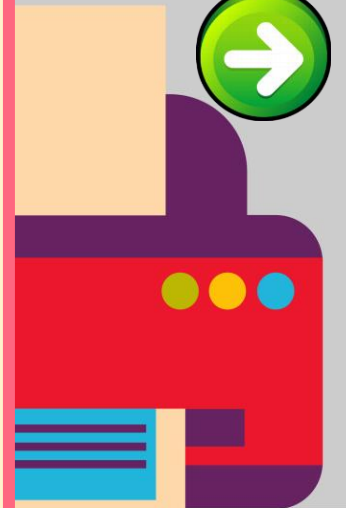
#บรรทัดที่ 5

```
“ส่วท”
```

```
921 ถนนสุขุมวิท
```

```
เขตคลองเตย กรุงเทพฯ
```

```
10110
```



```
Python Console x
>>> organization = "สสวท"
>>> address = "924 ถนนสุขุมวิท เขตคลองเตย กรุงเทพฯ 10110"
>>> print(organization)
"สสวท"
>>> print(address)
924 ถนนสุขุมวิท เขตคลองเตย กรุงเทพฯ 10110
>>> address = '''สสวท
... 921 ถนนสุขุมวิท
... เขตคลองเตย กรุงเทพฯ
... 10110'''
>>>
```

```
Python Console x
>>> address = '''สสวท
... 921 ถนนสุขุมวิท
... เขตคลองเตย กรุงเทพฯ
... 10110'''
>>> print(address)
"สสวท"
921 ถนนสุขุมวิท
เขตคลองเตย กรุงเทพฯ
10110
>>>
```

ตัวอย่างที่ 3.4

การกำหนดข้อมูลชนิดข้อความ

จากตัวอย่างที่ 3.4 อธิบายดังนี้

1. คำสั่งในบรรทัดที่ 1 เป็นการใช้เครื่องหมายอัญประกาศเดี่ยวเพื่อกำหนดตัวแปร organization เป็น “สสวท” ทำให้มีความยาว 6 ตัวอักษร (รวมเครื่องหมายอัญประกาศคู่ด้วย)
2. คำสั่งในบรรทัดที่ 2 ใช้เครื่องหมายอัญประกาศคู่เพื่อกำหนดสตริงให้กับตัวแปร address
3. คำสั่ง Print() ในบรรทัดที่ 3 และ 4 นำค่าในตัวแปร organization และ address ออกมาแสดงผล
4. คำสั่งในบรรทัดที่ 5 ใช้เครื่องหมาย ‘ จำนวน 3 ตัว กำหนด String จำนวน 4 บรรทัด ให้กับตัวแปร address
5. คำสั่งในบรรทัดที่ 6 นำค่าที่เก็บในตัวแปร address ออกมาแสดงผล

เกร็ดน่ารู้

ตัวดำเนินการ + และ * กับสตริง

เมื่อใช้ตัวดำเนินการ + ระหว่างสตริง 2 ตัว จะเป็นการนำสตริง 2 ตัว มาต่อกัน เช่น 'Hello' + 'World' จะเกิดสตริงใหม่ คือ 'HelloWorld' และถ้า ใช้ตัวดำเนินการ * ระหว่างสตริงกับค่าจำนวนเต็ม จะเป็นการนำสตริงมาต่อกันตามจำนวนครั้งของจำนวนเต็ม

หลังเครื่องหมาย * เช่น 'Go' * 3

จะเกิดสตริงใหม่ คือ 'GoGoGo'



ข้อมูลประเภทจำนวน (numerical data type)

ภาษา python มีข้อมูลจำนวนที่สามารถนำไปคำนวณทางคณิตศาสตร์ได้หลายชนิด ในที่นี่จะแนะนำ 2 ชนิดคือ

- จำนวนเต็มแบบมีเครื่องหมาย (signed integer) หรือเรียกว่าจำนวนเต็ม (integer หรือ int) สามารถเก็บค่าจำนวนเต็มบวกและจำนวนเต็มลบ
- จำนวนจริง (floating point number หรือ float) สามารถเก็บค่าทั้งจำนวนจริงบวกและจำนวนจริงลบ ที่อยู่ในรูปทศนิยมได้

ตัวอย่างที่ 3.5

การกำหนดข้อมูลจำนวนและการคำนวณพื้นฐาน

พิมพ์คำสั่งต่อไปนี้ในคอนโซล

```
base = 10                                #บรรทัดที่ 1
height = 15                              #บรรทัดที่ 2
area = (1/2) * base * height              #บรรทัดที่ 3
print ("base =", base)                   #บรรทัดที่ 4
print ("height =", height)                #บรรทัดที่ 5
print ("area =", area)                    #บรรทัดที่ 6
```

ตัวอย่างที่ 3.5

การกำหนดข้อมูลจำนวนและการคำนวณพื้นฐาน

ผลลัพธ์ที่ได้คือ

<code>base = 10</code>	<code>#บรรทัดที่ 1</code>
<code>height = 15</code>	<code>#บรรทัดที่ 2</code>
<code>area = (1/2) * base * height</code>	<code>#บรรทัดที่ 3</code>
<code>print ("base =", base)</code>	<code>#บรรทัดที่ 4</code>
<code>base = 10</code>	
<code>print ("height =", height)</code>	<code>#บรรทัดที่ 5</code>
<code>height = 15</code>	
<code>print ("area =", area)</code>	<code>#บรรทัดที่ 6</code>
<code>area = 75.0</code>	

ตัวอย่างที่ 3.5

การกำหนดข้อมูลจำนวนและการคำนวณพื้นฐาน

จากตัวอย่างที่ 3.5 อธิบายได้ว่า

1. บรรทัดที่ 1 และ 2 กำหนดให้ตัวแปร base และ height ขึ้นไปที่ 10 และ 15 ตามลำดับ
2. บรรทัดที่ 3 กำหนดให้ตัวแปร area ขึ้นที่ผลจากการคำนวณ $(1/2) * \text{base} * \text{height}$ ซึ่งเป็นการหาพื้นที่สามเหลี่ยมนั่นเอง (เมื่อใช้เครื่องหมาย * และ / กับข้อมูลจำนวน จะหมายถึง การคูณ และการหาร ทางคณิตศาสตร์ตามลำดับ)
3. บรรทัดที่ 4-6 เป็นคำสั่งแสดงผลของตัวแปรที่มีข้อสังเกตว่าคำสั่ง `print ()` สามารถรับค่าที่ต้องการให้แสดงผล ได้มากกว่า 1 ค่า โดยเขียนแต่ละค่าหรือแต่ละตัวแปรเรียงกันไป คั่นด้วยเครื่องหมายจุลภาค (,) เช่น บรรทัดที่ 4-6 สามารถเขียนรวมกันได้ดังนี้

```
print("base = ",base, "height = ",height, "area = ", area)
```

4. สังเกตอีกว่า คำสั่งในบรรทัดที่ 6 ได้ผลลัพธ์ของตัวแปร area แสดงออกมาในรูปแบบจำนวนจริง



เกร็ดน่ารู้

ตัวดำเนินการทางคณิตศาสตร์

ตัวดำเนินการสำหรับการบวก ลบ คูณและหารในไพทอนจะใช้เครื่องหมาย + , - , * และ / ตามลำดับ นอกจากนี้ยังมีเครื่องหมายเพิ่มเติมเช่น

**** หมายถึง ยกกำลัง** ตัวอย่างคือ $4^{**}3$ หมายถึง 4^3

// หมายถึง หารปัดเศษทิ้ง ตัวอย่างคือ $7//3$ ได้ผลลัพธ์เป็นจำนวนเต็ม 2

% หมายถึง เศษที่ได้จากการหาร ตัวอย่างคือ $7\%3$ ได้ผลลัพธ์เป็นจำนวนเต็ม 1

โดยสามารถใช้เครื่องหมาย () ล้อมรอบ นิพจน์ที่ต้องการให้ดำเนินการก่อน เช่นเดียวกับ การเขียนนิพจน์ทางคณิตศาสตร์



ตัวอย่างที่ 3.6 ทอนเท่าไร

นักเรียนซื้อแฟลชไดรฟ์ขนาด 32 GB ราคา 117 บาท ที่ร้านสะดวกซื้อ โดยจ่ายเงินด้วยธนบัตร 100 บาท 2 ใบ ในลิ้นชักเก็บเงินของร้าน มีแต่ธนบัตร 20 บาท และเหรียญ 1 บาท พนักงานจะต้องทอนเงินเป็นธนบัตร 20 บาท จำนวนกี่ใบ และเหรียญ 1 บาท จำนวนกี่เหรียญ
ให้ใช้คำสั่ง `python` แสดงวิธีการหาคำตอบที่ละเอียด



ตัวอย่างที่ 3.6 ทอนเท่าไร

แนวคิดการแก้ปัญหา

1. คำนวณจำนวนเงินทอน
2. คำนวณจำนวนธนบัตรที่ยีสิบบาทที่ได้รับ จากจำนวนเงินทอนหารด้วย 20 โดย**ปัดเศษทิ้ง**
3. คำนวณจำนวนเหรียญบาทที่ได้รับ จากจำนวนเงินทอนหารด้วย 20 โดย**นำมาเฉพาะเศษที่ได้จากการหาร**



ตัวอย่างที่ 3.6 ทอนเท่าไร

พิมพ์คำสั่งต่อไปนี้ในคอนโซล

```
change = 200-117
note20 = change // 20
coin1 = change % 20
print("เงินทอนทั้งหมด" , change , "บาท")
print("เป็นธนบัตรยี่สิบบาท จำนวน" , note20 , "ใบ")
print("เป็นเหรียญหนึ่งบาท จำนวน" , coin1 , "เหรียญ")
```



ตัวอย่างที่ 3.6 ทอนเท่าไร

ผลลัพธ์ที่ได้คือ

```
change = 200-117
```

```
note20 = change // 20
```

```
coin1 = change % 20
```

```
print("เงินทอนทั้งหมด", change, "บาท")
```

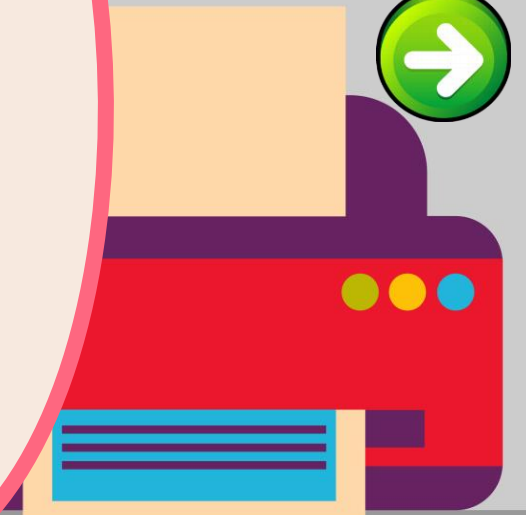
เงินทอนทั้งหมด 83 บาท

```
print("เป็นธนบัตรยี่สิบบาท จำนวน", note20, "ใบ")
```

เป็นธนบัตรยี่สิบบาท จำนวน 4 ใบ

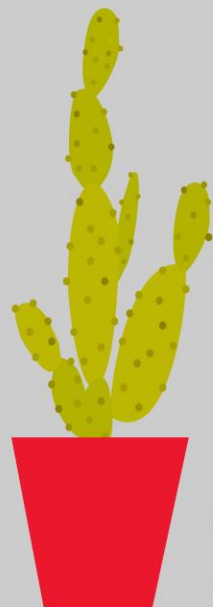
```
print("เป็นเหรียญหนึ่งบาท จำนวน", coin1, "เหรียญ")
```

เป็นเหรียญหนึ่งบาท จำนวน 3 เหรียญ



1

การแปลงชนิดข้อมูล



การแปลงชนิดข้อมูล

ฟังก์ชัน `input ()` จะรับข้อมูลเข้าจากผู้ใช้งาน
คีย์บอร์ด และจะเป็นข้อมูลชนิดสตริงเท่านั้น ซึ่งไม่
สามารถนำไปคำนวณทางคณิตศาสตร์ได้



การแปลงชนิดข้อมูล

นักเรียนคิดว่าคำสั่งต่อไปนี้ ทำงานได้หรือไม่

```
width = input ("ป้อนความกว้างของสี่เหลี่ยม")  
length = input ("ป้อนความยาวของสี่เหลี่ยม")  
area = width * length  
print("พื้นที่สี่เหลี่ยม คือ" , area, "ตารางหน่วย")
```



การแปลงชนิดข้อมูล

ไม่ได้ เพราะ หากพิมพ์คำสั่งในคอนโซลจะมีข้อผิดพลาดจากการทำงาน ในคำสั่ง $\text{area} = \text{width} * \text{length}$ เพราะตัวแปร `width` และ `length` จะเก็บข้อมูลสตริง แม้ว่าผู้ใช้จะป้อนข้อมูลเป็นตัวเลขก็ตาม ดังนั้นหากต้องการนำค่าที่ได้รับจากฟังก์ชัน `input()` ไปใช้ในการคำนวณทางคณิตศาสตร์ จะต้องแปลงให้เป็นข้อมูลชนิดจำนวนก่อน โดยใช้ฟังก์ชัน `int()` เพื่อแปลงสตริงเป็นจำนวนเต็มหรือฟังก์ชัน `float()` เพื่อแปลงสตริงเป็นจำนวนจริง

ตัวอย่างที่ 3.7

การแปลงข้อมูลสตริงให้เป็นข้อมูลชนิดจำนวน

พิมพ์คำสั่งต่อไปนี้ในคอนโซล

```
val = input("ป้อนค่าจำนวนเต็มค่าหนึ่ง")    #บรรทัดที่ 1
type(val)                                    #บรรทัดที่ 2
intVal = int(val)                             #บรรทัดที่ 3
type(intVal)                                 #บรรทัดที่ 4
```

ตัวอย่างที่ 3.7

การแปลงข้อมูลสตริงให้เป็นข้อมูลชนิดจำนวน

ผลลัพธ์ที่ได้คือ

```
val = input("ป้อนค่าจำนวนเต็มค่าหนึ่ง")      #บรรทัดที่ 1
ป้อนค่าจำนวนเต็มค่าหนึ่ง >? 4
type(val)                                       #บรรทัดที่ 2
str
intVal = int(val)                              #บรรทัดที่ 3
type(intVal)                                   #บรรทัดที่ 4
int
```



ตัวอย่างที่ 3.7

การแปลงข้อมูลสตริงให้เป็นข้อมูลชนิดจำนวน

จากตัวอย่างที่ 3.7 อธิบายได้ดังนี้

1. หลังจากไพทอนรันคำสั่งในบรรทัดที่ 1 และผู้ใช้ป้อนตัวอักขระ "4" แล้วตัวแปร `val` จะเก็บข้อมูลสตริง ของอักขระ "4" ซึ่งตรวจสอบได้จากคำสั่ง `type(val)` ในบรรทัดที่ 2
2. บรรทัดที่ 3 ใช้ฟังก์ชัน `int()` ในการแปลงค่าสตริงของตัวแปรให้เป็นจำนวนเต็ม แล้วกำหนดค่าให้กับตัวแปร `intVal` ซึ่งตรวจสอบชนิดได้จากคำสั่งในบรรทัดที่ 4



ตัวอย่างที่ 3.8 ร่วมด้วยช่วยแชร์

ถ้านักเรียนไปรับประทานอาหารฉลองวันปิดเทอมกับเพื่อนๆ และตกลงกันว่าจะจ่ายค่าอาหารคนละเท่าๆกัน นักเรียนแต่ละคนจะต้องจ่ายค่าอาหารคนละเท่าใด ให้ใช้คำสั่งไพทอนแสดงวิธีการหาคำตอบที่ละเอียด

แนวคิดการแก้ปัญหา

1. `totalPrice` ← รับค่าอาหารทั้งหมด
2. `number` ← รับจำนวนผู้รับประทานอาหาร
3. `avg` ← ค่าอาหารทั้งหมด/จำนวนผู้รับประทานอาหาร
4. แสดงผล `avg`

ตัวอย่างที่ 3.7

การแปลงข้อมูลสตริงให้เป็นข้อมูลชนิดจำนวน

พิมพ์คำสั่งต่อไปนี้ในคอนโซล

```
totalPrice = int (input ('ค่าอาหารทั้งหมด'))           #บรรทัดที่ 1
number = int   (input ('จำนวนผู้รับประทานอาหาร'))    #บรรทัดที่ 2
avg = totalPrice / number                               #บรรทัดที่ 3
print("จ่ายค่าอาหารคนละ", avg, 'บาท')                  #บรรทัดที่ 4
```



ตัวอย่างที่ 3.7

การแปลงข้อมูลสตริงให้เป็นข้อมูลชนิดจำนวน

ผลลัพธ์ที่ได้คือ

```
totalPrice = int (input ('ค่าอาหารทั้งหมด')) #บรรทัดที่ 1
```

ราคาอาหารทั้งหมด >? 1289

```
number = int (input ('จำนวนผู้รับประทานอาหาร')) #บรรทัดที่ 2
```

จำนวนผู้รับประทานอาหาร >? 15

```
avg = totalPrice / number #บรรทัดที่ 3
```

```
print("จ่ายค่าอาหารคนละ", avg, 'บาท') #บรรทัดที่ 4
```

จ่ายค่าอาหารคนละ 85.93333333333334 บาท



ชวนคิด

1. จากตัวอย่างที่ 3.8 ถ้านักเรียนต้องการให้การแสดงผลเป็นทศนิยมสองตำแหน่ง จะใช้คำสั่งใด
2. เขียนโปรแกรมคำนวณหาพื้นที่วงกลม โดยรับข้อมูลเข้าเป็นความยาวของเส้นผ่านศูนย์กลาง





ผู้จัดทำ

นายกานตพงศ์ สุวรรณทา
ตำแหน่ง ครู