

# PYTHON

**Unit 6-1-1 print**

# ภาษา Python คืออะไร



Python เป็นภาษาเขียนโปรแกรมระดับสูงที่ใช้กันอย่างกว้างขวางในการเขียนโปรแกรมสำหรับวัตถุประสงค์ทั่วไป ภาษา Python นั้นสร้างโดย Guido van Rossum และถูกเผยแพร่ครั้งแรกในปี 1991 Python นั้นเป็นภาษาแบบ interprete ที่ถูกออกแบบโดยมีปรัชญาที่จะทำให้โค้ดอ่านได้ง่ายขึ้น และโครงสร้างของภาษานั้นจะทำให้โปรแกรมเมอร์สามารถเข้าใจแนวคิดการเขียนโค้ดโดยใช้บรรทัดที่น้อยลงกว่าภาษาอย่าง C++ และ Java ซึ่งภาษานั้นถูกกำหนดให้มีโครงสร้างที่ตั้งใจให้การเขียนโค้ดเข้าใจง่ายทั้งในโปรแกรมเล็กไปจนถึงโปรแกรมขนาดใหญ่

# ข้อดีของ Python

1. เขียนง่าย
2. มี Community ขนาดใหญ่
3. มี Library ให้ใช้งานหลากหลาย
4. เหมาะกับงาน Data Science
5. เป็นที่ต้องการของหลายๆ บริษัท

# Python ถูกนำไปใช้ที่ไหนบ้าง

Uber

NETFLIX

amazon

YouTube



Spotify

# เริ่มต้นเขียนโปรแกรม

ก่อนเริ่มต้นทำการเขียนโปรแกรม Python มาทบทวนหลักการการทำงานของคอมพิวเตอร์โดยผู้เขียนโปรแกรมต้องออกแบบอัลกอริทึมก่อน โดยต้องวิเคราะห์ว่า มีข้อมูลนำเข้าอะไร ประมวลผลด้วยวิธีการอย่างไร และต้องการให้แสดงผลอย่างไร ซึ่งเป็นหลักการทำงานของคอมพิวเตอร์ที่ต้องการ หน่วยข้อมูลเข้า หน่วยประมวลผล และหน่วยแสดงผลหรือข้อมูลออก



# เริ่มต้นเขียนโปรแกรม



# เริ่มต้นเขียนโปรแกรม

ถ้าแบ่งคำสั่งของโปรแกรมภาษา Python ออกตามหลักการทำงานของคอมพิวเตอร์คือ การใช้แป้นพิมพ์ส่งข้อมูล การประมวลผลด้วยหน่วยประมวลผลกลาง และการแสดงผล โดยใช้จอคอมพิวเตอร์จะทำให้สามารถแบ่งหัวข้อการเรียนรู้คำสั่งต่าง ๆ

คำสั่งนำข้อมูลเข้า (Input)	คำสั่งประมวลผล	คำสั่งแสดงผลข้อมูล (output)
คำสั่งกำหนดค่าตัวแปร ชื่อตัวแปร = .....	ตัวดำเนินการทางคณิตศาสตร์ เปรียบเทียบ	คำสั่งแสดงผลข้อมูลออกทางหน้าจอ Print()
คำสั่งนำข้อมูลเข้าจากแป้นพิมพ์ input()	ตัวดำเนินการ ตรรกะ: กำหนดค่า สมาชิก	การกำหนดรูปแบบ การแปลงข้อมูล การแสดงผลข้อมูล
ชนิดของข้อมูล เบื้องต้น Number, Sequence Type ,Boolean	คำสั่งการทำงานแบบมีเงื่อนไข if ....else...	* การส่งออกไฟล์
ฟังก์ชันและการสร้างฟังก์ชัน def() *Library random datetime turtle	ตัวดำเนินการ ตรวจสอบเงื่อนไข	
*ชนิดของข้อมูล Set , Dictionary * การเรียกใช้ไฟล์	คำสั่งการทำงานแบบวนซ้ำ for .... in .... while	

คำสั่งนำข้อมูลเข้า (Input)	คำสั่งประมวลผล	คำสั่งแสดงผลข้อมูล (output)
คำสั่งกำหนดค่าตัวแปร ชื่อตัวแปร = .....	ตัวดำเนินการทาง คณิตศาสตร์ เปรียบเทียบ	คำสั่งแสดงผลข้อมูลออกทางหน้าจอ <b>Print()</b>
คำสั่งนำข้อมูลเข้าจากแป้นพิมพ์ <b>input()</b>	ตัวดำเนินการ ตรรกะ กำหนดค่า สมาชิก	การกำหนดรูปแบบ การแปลงข้อมูล การแสดงผลข้อมูล
ชนิดของข้อมูล เบื้องต้น Number, Sequence Type ,Boolean	คำสั่งการทำงานแบบมี เงื่อนไข if ....else...	<b>* การส่งออกไฟล์</b>
ฟังก์ชันและการสร้างฟังก์ชัน <b>def()</b> <b>*Library random datetime turtle</b>	ตัวดำเนินการ ตรวจสอบเงื่อนไข	
<b>*ชนิดของข้อมูล Set , Dictionary</b> <b>* การเรียกใช้ไฟล์</b>	คำสั่งการทำงานแบบวนซ้ำ for .... in .... while	



# PRINT COMMAND

คำสั่งแสดงผลทางหน้าจอ

Print Command

# PRINT (ข้อมูล)

ออกทางหน้าจอแสดงผล

## คำสั่งการแสดงผลค่าบนหน้าจอคอมพิวเตอร์



```
print(ค่าที่ 1,ค่าที่2, ..., sep = ' ', end = '\n')
```

เป็นคำสั่งแสดงผลค่าที่ต้องการบนหน้าจอโดยสามารถแสดงผลค่าได้ทั้งตัวอักษร ตัวเลข และอื่น ๆ (ขึ้นอยู่กับชนิดของข้อมูล) หรือ ตัวแปร ได้หลายค่าโดยใช้เครื่องหมาย , คั่น โดยหากค่าที่ใส่เป็นจำนวนไม่ต้องใส่ " " double quotes หรือ ' ' single quotes และ ถ้าเป็นตัวอักษรต้องใส่หากไม่ใส่จะนับเป็นค่าตัวแปร ซึ่งจะจบด้วยขึ้นบรรทัดใหม่ \n เสมอ โดยมีตัวอย่างดังต่อไปนี้

# ตัวอย่างการใช้งานคำสั่ง print ที่ 1

ตัวอย่างคำสั่ง	รายละเอียดคำสั่ง	ผลลัพธ์ของโปรแกรมที่เขียน
<code>Print ("Lipda","chan")</code>	ให้แสดงอักษรคำว่า Lipda และ: chan ออกทางหน้าจอ	Lipda chan
<code>Print (2.3,1.8)</code>	ให้แสดงค่าจำนวน 2.3 และ: 1.8	2.3 1.8

Live

Upload



CyberPi

[Help Docs for Device](#)

⇒ Devices

Connect

Project Files



main.py



main.py ×

1

## กิจกรรมที่ 6-1-1.1

ให้เขียนโปรแกรม Python

โดยให้แสดงผล ชื่อ นามสกุล อายุของตนเอง

โดยให้เขียนรหัสล้าลองด้วย

*คำถามท้ายกิจกรรม : ข้อมูลที่พิมพ์ลงไปนักเรียนคิดว่าเป็นข้อมูลชนิดใดบ้าง*



Run



Upload



Clear

Log



```
Python 3.6.5 (default, Sep 28 2019, 10:46:48)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> |
```

# ตัวอย่างการใช้งานคำสั่ง print ที่ 2

ตัวอย่างคำสั่ง	รายละเอียดคำสั่ง	ผลลัพธ์ของโปรแกรมที่เขียน
<code>print ("Lipda\tPola","\tLipda\tPola")</code>	ให้แสดงผลคำว่า Lipda Pola 2 ครั้งโดยใส่ \t หรือ <b>Tab แนวนอน</b>	Lipda    Pola    Lipda    Pola
<code>Print (2.3+1.8)</code>	ให้แสดงค่าผลบวก 2.3+1.8 ออกทางหน้าจอ	4.1

หมายเหตุ : 1. การแบ่งข้อมูลที่จะแสดงหรือคำสั่งต้องมีเครื่องหมาย , คั่น

# คำสั่ง backslash(\) การจัดการแสดงผลบนหน้าจอ Escape character

escape character	ความหมาย	escape character	ความหมาย
\' หรือ \"	การใส่ single หรือ double quotes	\\	การใส่ Back-slash
\n	new line - การขึ้นบรรทัดใหม่	\r	Carriage Return ย้ายตำแหน่ง Cursor ไปซ้ายสุดของบรรทัด
\t	ย้ายตำแหน่ง Cursor ไป 1 Tab แนวนอน	\v	ย้ายตำแหน่ง Cursor ไป 1 Tab แนวตั้ง
\b	ย้ายตำแหน่ง Cursor ไป 1 Backspace	\f	form feed - ขึ้นหน้าใหม่

# ตัวอย่างการใช้งานคำสั่ง print ที่ 3

ตัวอย่างคำสั่ง	รายละเอียดคำสั่ง	ผลลัพธ์ของโปรแกรมที่เขียน
<code>Print (2.3,1.8,sep=",")</code>	ให้แสดงค่าจำนวน 2.3 และ 1.8 และมี , คั่นระหว่างค่าทั้งสอง	2.3,1.8
<code>print (2.3,1.8,sep="\n")</code>	ให้แสดงค่าจำนวน 2.3 และ 1.8 โดยให้ขึ้นบรรทัดใหม่	2.3 1.8

- หมายเหตุ :
2. หากต้องการแสดงเครื่องหมายระหว่างค่าที่แสดงให้ใช้คำสั่ง `sep = '....'` (Seperate)
  3. หากต้องการขึ้นบรรทัดใดระหว่างข้อมูล ให้เพิ่ม `sep = '\n'` (Escape character)
  4. โดยปกติคำสั่งแต่ละคำสั่งจะขึ้นบรรทัดใหม่ หากไม่ต้องการขึ้นบรรทัดใหม่ให้ใส่ `end`



# ตัวอย่างการใช้งานคำสั่ง print ที่ 4

ตัวอย่างคำสั่ง	รายละเอียดคำสั่ง	ผลลัพธ์ของโปรแกรมที่เขียน
<code>Print ("a"*10)</code>	ให้แสดงอักษรคำว่า a จำนวน 10 ครั้งออกทางหน้าจอ	aaaaaaaaaa

Live

Upload



CyberPi

[Help Docs for Device](#)

Devices

Connect

Project Files



main.py

main.py x

1

## กิจกรรมที่ 6-1-1.2

ให้นักเรียนใช้คำสั่ง Print

เพื่อสร้างภาพ เรขาคณิต สีเหลี่ยม

หรือภาพที่ต้องการ 1 ภาพ

โดยให้เขียนรหัสล้าลองด้วย

Run Upload Clear

Log

```
Python 3.6.5 (default, Sep 28 2019, 10:46:48)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> |
```



Live

Upload



CyberPi

[Help Docs for Device](#)

⇒ Devices

Connect

Project Files



main.py

main.py ×

1

## กิจกรรมที่ 6-1-1.3

ให้แสดง ชื่อ นามสกุล สถานะในครอบครัว หรืออื่น ๆ ที่นักเรียนต้องการโดยให้มีการค้นด้วย , โดยให้แสดงในบรรทัดเดียว โดยให้เขียนรหัสล้าลองด้วย

Run Upload Clear

Log

```
Python 3.6.5 (default, Sep 28 2019, 10:46:48)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> |
```

## 2. Print Numeric

### Numbers

ภาษาไพธอนจะมีชนิดข้อมูลอยู่ 9 ชนิดด้วยกัน ได้แก่ ตัวเลข ข้อมูล เรียงลำดับ การจับคู่ callable มอดูล คลาส อินสแตนซ์คลาส ไฟล์ และ internal ในแต่ละชนิดจะมีชนิดข้อมูลย่อยอีก แต่ที่จะนำมาศึกษาในเล่มนี้จะ นำเอาชนิดของข้อมูลที่เป็นพื้นฐานที่ใช้กันบ่อย ๆ ซึ่งอยู่ด้วยกัน 4 ชนิด ได้แก่ integer, float, string และ dictionary ซึ่งขึ้นอยู่กับตัวแปรที่จะใช้ว่าต้องการ กำหนดให้เป็นข้อมูลชนิดใด

คำสั่งนำข้อมูลเข้า (Input)	คำสั่งประมวลผล	คำสั่งแสดงผลข้อมูล (output)
คำสั่งกำหนดค่าตัวแปร ชื่อตัวแปร = .....	ตัวดำเนินการทาง คณิตศาสตร์ เปรียบเทียบ	คำสั่งแสดงผลข้อมูลออกทางหน้าจอ Print()
คำสั่งนำข้อมูลเข้าจากแป้นพิมพ์ input()	ตัวดำเนินการ ตรรกะ: กำหนดค่า สมาชิก	การกำหนดรูปแบบ การแปลงข้อมูล การแสดงผลข้อมูล
ชนิดของข้อมูล เบื้องต้น Number, Sequence Type , Boolean	คำสั่งการทำงานแบบมี เงื่อนไข if ....else...	* การส่งออกไฟล์
ฟังก์ชันและการสร้างฟังก์ชัน def() *Library random datetime turtle	ตัวดำเนินการ ตรวจสอบเงื่อนไข	
*ชนิดของข้อมูล Set , Dictionary * การเรียกใช้ไฟล์	คำสั่งการทำงานแบบวนซ้ำ for .... in .... while	

## 3.3 การแสดงผลข้อมูล กับการใช้สัญลักษณ์ %d %f %s

สัญลักษณ์พิเศษหรือรหัสในการแสดงค่า Print ตามชนิดของข้อมูล (%s %d %f)

ในการแสดงค่าข้อมูลหน้าจอสสามารถกำหนดให้แสดงได้โดยใช้รหัสแทนการใช้คำสั่งแปลงค่า

%s จัดรูปแบบเป็นตัวอักษร (String)

%d จัดรูปแบบเป็นตัวเลขฐานสิบ (Decimal Number)

%f จัดรูปแบบเป็นจำนวนจริง (Floating Number)

```
print ("ข้อความ %(c,s,d,f)" %ตัวแปร)
```

```
print ("ข้อความ %s, %d, %f" %(ตัวแปร, %ตัวแปร, %ตัวแปร))
```

## ตัวอย่างการใช้คำสั่ง

## ผลลัพธ์

```
x,y,z = "Lipda",99.99,13.14
```

```
print ("แสดงค่า x= ให้เป็นค่าตัวอักษร %s" %x)
```

แสดงค่า x= ให้เป็นค่าตัวอักษร Lipda

**print( แสดงค่า x = ให้เป็นค่าตัวอักษร %s %x )**



เป็นการนำค่าในตัวแปร x คือ Lipda ไปใส่ใน %ที่อยู่ด้านหน้า

โดยหลังจากหมด ' single หรือ double quotes ไม่ต้องมี , comma คั่น

## ตัวอย่างการใช้คำสั่ง

## ผลลัพธ์

```
x,y,z = "Lipda",99.99,13.14
```

```
print ("แสดงค่า y= ให้เป็นค่าตัวเลขจำนวนเต็มฐานสิบ %d" %y)
```

แสดงค่า y= ให้เป็นค่าตัวเลขจำนวนเต็มฐานสิบ 99

`print( แสดงค่า y = ให้เป็นค่าจำนวนเต็มฐานสิบ %d %y )`



เป็นการนำค่าในตัวแปร y คือ 99.99 ที่เป็นจำนวนเต็มไปใส่ใน %ที่อยู่ด้านหน้า

**MAKE** โดยหลังจากหมด ' single หรือ double quotes ไม่ต้องมี , comma คั่น



## ตัวอย่างการใช้คำสั่ง

## ผลลัพธ์

```
x,y,z = "Lipda",99.99,13.14
```

```
print ("แสดงค่า z= ให้เป็นค่าจำนวนจริง =%f" %z)
```

```
แสดงค่า z= ให้เป็นค่าจำนวนจริง =13.140000
```

`print( แสดงค่า z = ให้เป็นค่าจำนวนจริง %f %z )`

เป็นการนำค่าในตัวแปร z คือ 13.14 ที่เป็นจำนวนเต็มไปใส่ใน %ที่อยู่ด้านหน้า โดยหลังจากหมด ' single หรือ double quotes ไม่ต้องมี , comma คั่น

## ตัวอย่างการใช้คำสั่ง

## ผลลัพธ์

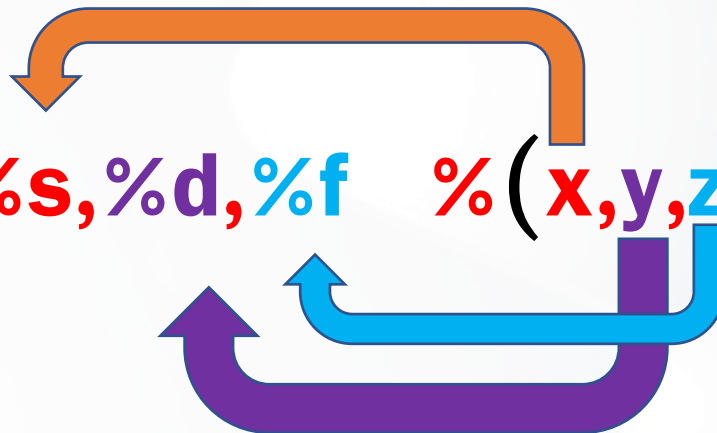
```
x,y,z = "Lipda",99.99,13.14
```

```
print ("แสดงค่า x y z= %s,%d,%f" %(x,y,z))
```

```
แสดงค่า x y z= Lipda,99,13.140000
```



`print( แสดงค่า x y z = %s,%d,%f %(x,y,z) )`

A diagram with three colored arrows: an orange arrow at the top points from the variable 'x' in the code to the string 'Lipda' in the output; a blue arrow at the bottom points from the variable 'y' in the code to the integer '99' in the output; and a purple arrow at the bottom points from the variable 'z' in the code to the float '13.140000' in the output.

เป็นการนำค่าในตัวแปร x y z คือ Lipda,99.99,13.14 ไปใส่ใน %ที่อยู่ด้านหน้า โดยเรียงลำดับการใส่ก่อนหลัง %ตัวแรกเป็น ค่า x ตัวที่ 2 เป็นค่า y ค่าตัวที่ 3 เป็นค่า z