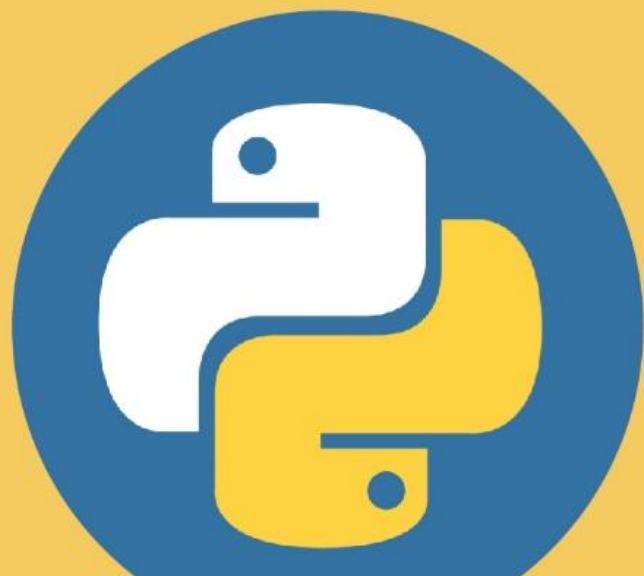


CDG Code Their Dreams X CED KMUTNB



Introduction to Programing with Python



โครงการ Public Training หลักสูตรอบรมการเขียนโปรแกรมคอมพิวเตอร์เพื่อการศึกษา

โครงการความร่วมมือระหว่าง CDG Code Their Dreams กับ ภาควิชาคอมพิวเตอร์ศึกษา คณะครุศาสตร์อุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ช่องทางประชาสัมพันธ์หลัก

Facebook Fanpage :

- Code Their Dreams
- ศูนย์ส่งเสริมการเรียนรู้ สมาคมศิษย์เก่าครุศาสตร์อุตสาหกรรม มจพ.

ช่องทางประชาสัมพันธ์เพิ่มเติม

- หน่วยฝึกอบรมภาควิชาคอมพิวเตอร์ศึกษา มจพ.

Coding

เริ่มเรียนรู้

output

Hello Python

Code

```
print ("Hello Python")
```

Code

```
print ("Hello Python")
```

1. `print ("Name surname")`
2. `print ('Name surname')`
3. `print (Hello Python)`
4. `print (2)`
5. `print ()`

เริ่มเรียนรู้

Code

```
1. print("Titipong Thammavisut")
```

output

```
Titipong Thammavisut
```

เริ่มเรียนรู้

Code

```
2. print('Titipong Thammavisut')
```

output

```
Titipong Thammavisut
```


เริ่มเรียนรู้

Code

```
3. print(Hello Python)
```

output

```
File "<ipython-input-2-9489b56b625c>", line 1  
    print(Hello Python)  
          ^
```

```
SyntaxError: invalid syntax
```

เริ่มเรียนรู้

Code

```
4. print(1)
```

output

```
1
```

ข้อผิดพลาดและความหมายของ
Syntax

ข้อผิดพลาด

Code

```
print (Hello Python)
```

output

```
File "<ipython-input-2-9489b56b625c>", line 1  
  print(Hello Python)  
        ^
```

SyntaxError: invalid syntax

Code

```
print (1)
```

output

1

ข้อผิดพลาด

output

```
File "<ipython-input-2-9489b56b625c>", line 1  
    print(Hello Python)
```

^

```
SyntaxError: invalid syntax
```

ข้อผิดพลาด

output

```
File "<ipython-input-2-9489b56b625c>", line 1  
    print(Hello Python)
```

^

```
SyntaxError: invalid syntax
```

บรรทัดที่มีปัญหา

ข้อผิดพลาด

output

```
File "<ipython-input-2-9489b56b625c>", line 1  
    print(Hello Python)  
          ^
```

```
SyntaxError: invalid syntax
```

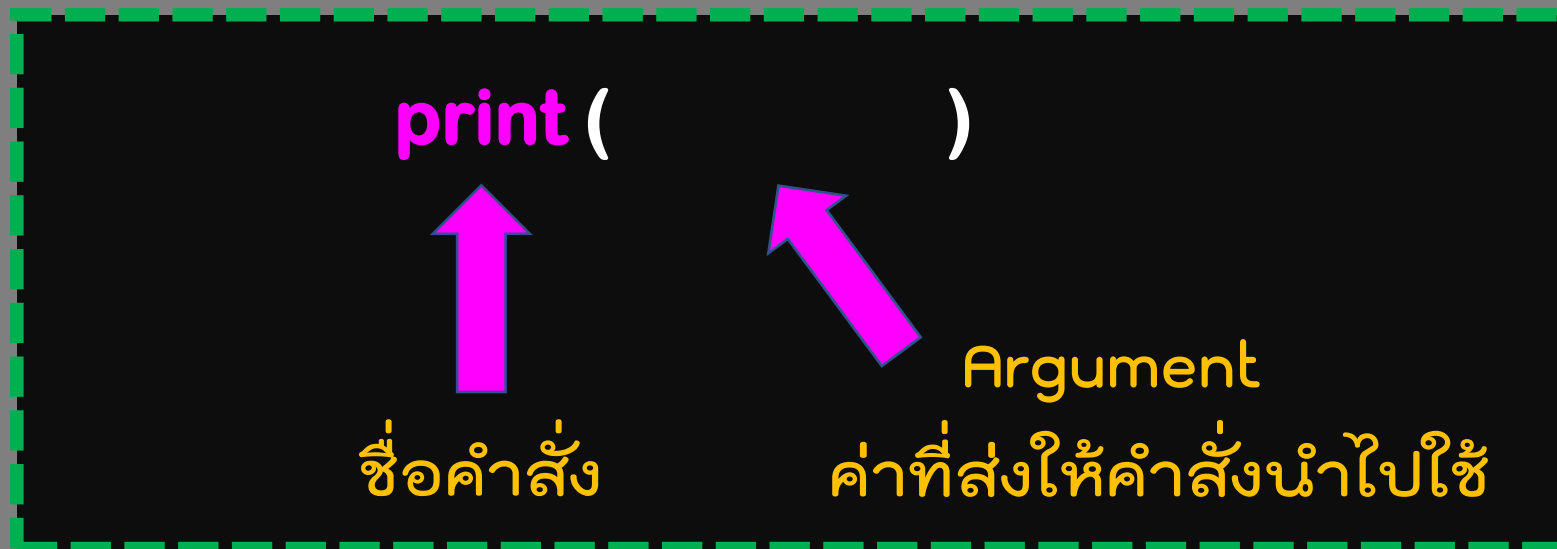
บรรทัดที่มีปัญหา

รูปแบบข้อผิดพลาด

Syntax

โครงสร้างของคำสั่ง กฎการปฏิบัติ

output

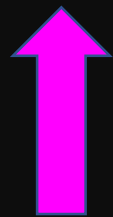


Syntax

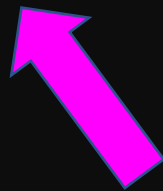
โครงสร้างของคำสั่ง กฎการปฏิบัติ

output

print ("Hello Python")



ชื่อคำสั่ง



Argument

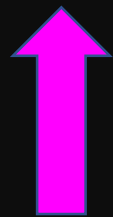
ค่าที่ส่งให้คำสั่งนำไปใช้

Syntax

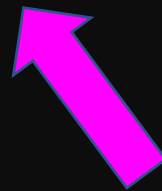
โครงสร้างของคำสั่ง กฎการปฏิบัติ

output

print ('Hello Python')



ชื่อคำสั่ง



Argument

ค่าที่ส่งให้คำสั่งนำไปใช้

Syntax

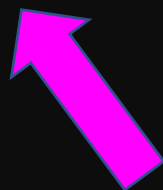
โครงสร้างของคำสั่ง กฎการปฏิบัติ

output

`print (1)`



ชื่อคำสั่ง



Argument

ค่าที่ส่งให้คำสั่งนำไปใช้

Syntax

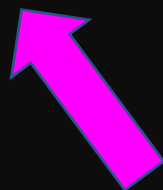
โครงสร้างของคำสั่ง กฎการปฏิบัติ

output

print (variable)



ชื่อคำสั่ง



Argument

ค่าที่ส่งให้คำสั่งนำไปใช้

ตัวดำเป็นการทำงานคณิตศาสตร์

ตัวดำเนินการทางคณิตศาสตร์

Code

```
print (1)
```

output

```
1
```

ตัวดำเนินการทางคณิตศาสตร์

Code

```
print (1+5)
```

output

6

ตัวดำเนินการทางคณิตศาสตร์

+ บวก

$$7 + 3 = 10$$

- ลบ

$$7 - 3 = 4$$

***** คูณ

$$7 * 3 = 21$$

/ หาร

$$7 / 3 = 2.333333333$$

% มอดุโล

$$7 \% 3 = 1$$

****** ยกกำลัง

$$7 ** 3 = 343$$

// หารไม่เอาเศษ

$$7 // 3 = 2$$

Question

ลองหาคำตอบกันดูซิ...ว่าได้อะไรออกมา

```
print (20.5 - 30)
```

```
print (50 / 5)
```

```
print (50 // 5)
```

```
print (1 + 1 + 1 * 0)
```

ลำดับความสำคัญของเครื่องหมาย

ลำดับความสำคัญของเครื่องหมาย

`print (20.5 - 30)` = -9.5

`print (50 / 5)` = 10.0

`print (50 // 5)` = 10

`print (1 + 1 + 1 * 0)` = 2

ลำดับความสำคัญของเครื่องหมาย

`print (1 + 1 + 1 * 0)` = 2

`print (1 + 1 + (1 * 0))` = 2

`print ((1 + 1 + 1) * 0)` = 0

`print (1 + (1 + 1) * 0)` = 1

ลำดับความสำคัญของเครื่องหมาย

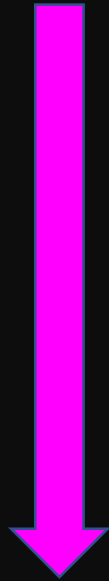
1. ()

2. **

3. *, /, //, %

4. +, -

มากที่สุด



น้อยสุด

ลำดับความสำคัญของเครื่องหมาย

`print (7 // 5 * 3)` = 3

`print (2 * (1 - 3))` = -4

`print ((1 + 1 + 1) * 0)` = 0

`print (1 + (1 + 1) * 0)` = 1

ลำดับความสำคัญของเครื่องหมาย

Code

```
print(7 // 5 * 3)
```

output

$$7 // 5 = 1$$

$$1 * 3 = 3$$

3

Code

```
print(2 * (1 - 3))
```

output

$$1 - 3 = -2$$

$$-2 * 2 = -4$$

-4

Question

ลองหาคำตอบกันดูซิ...ว่าได้อะไรออกมา

Code

```
print ((4 + 7 % 3) - 1 * 3 ** 2)
```

output

$$7 \% 3 = 1$$

$$9 * -1 = -9$$

$$4 + 1 = 5$$

$$5 + -9 = -4$$

$$3 \wedge 3 = 9$$

-4

ตัวแปร

ตัวแปร

โปรแกรมแสดงค่าเส้นรอบวงและพื้นที่ของวงกลม

สูตรการหาเส้นรอบวง = $2\pi r$ และพื้นที่ = πr^2

$\pi = 3.14$ $r = 3$

Code

```
print(2 * 3.14 * 3)  
print(3.14 * (3 ** 2))
```

output

18.84

28.26

ตัวแปร

โปรแกรมแสดงค่าเส้นรอบวงและพื้นที่ของวงกลม

สูตรการหาเส้นรอบวง = $2\pi r$ และพื้นที่ = πr^2

$\pi = 3.14$ $r = 45$

Code

```
print(2 * 3.14 * 3)  
print(3.14 * (3 ** 2))
```

output

18.84

28.26

ตัวแปร

โปรแกรมแสดงค่าเส้นรอบวงและพื้นที่ของวงกลม

สูตรการหาเส้นรอบวง = $2\pi r$ และพื้นที่ = πr^2

$\pi = 3.14$ $r = 45$

Code

```
print(2 * 3.14 * 45)
```

```
print(3.14 * (45 ** 2))
```

output

282.6

6358.5

ตัวแปร

โปรแกรมแสดงค่าเส้นรอบวงและพื้นที่ของวงกลม

สูตรการหาเส้นรอบวง = $2\pi r$ และพื้นที่ = πr^2

$\pi = 3.14159$ $r = 45$

Code

output

```
print(2 * 3.14159 * 45)
```

```
print(3.14159 * (45 ** 2))
```

```
282.74309999999997
```

```
6361.71975
```

ตัวแปร

โปรแกรมแสดงค่าเส้นรอบวงและพื้นที่ของวงกลม

สูตรการหาเส้นรอบวง = $2\pi r$ และพื้นที่ = πr^2

$\pi = 3.14$ $r = 3$

Code

output

```
pi = 3.14
```

```
r = 3
```

```
print(2 * pi * r)
```

```
print(pi * (r ** 2))
```

18.84

28.26

ตัวแปร

โปรแกรมแสดงค่าเส้นรอบวงและพื้นที่ของวงกลม

สูตรการหาเส้นรอบวง = $2\pi r$ และพื้นที่ = πr^2

$\pi = 3.14159$ $r = 45$

Code

output

```
pi = 3.14159
```

```
r = 45
```

```
print (2 * pi * r)
```

```
print (pi * (r ** 2))
```

```
282.74309999999997
```

```
6361.71975
```

การตั้งชื่อตัวแปร

การตั้งชื่อตัวแปร

1. ขึ้นต้นด้วยตัวอักษร A-Z หรือ a-z หรือ
2. ตามด้วยตัวอักษร ตัวเลข (0-9) หรือ `_` ก็ตัวก็ได้
3. ห้ามตั้งตรงกับคำที่สงวนไว้

```
r  
pi  
radius  
word_length  
text1  
text2
```

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

การกำหนดค่าให้กับตัวแปร

การกำหนดค่าโดยใช้เครื่องหมาย =

```
pi = 3.14  
radius = 3  
area = pi * (radius ** 2)
```

การเรียกใช้ค่าตัวแปร

```
print (pi * (radius ** 2))  
print (area)
```

ตัวแปรสามารถเปลี่ยนค่าได้

Code

```
radius = 3  
print (radius)
```

output

3

ตัวแปรสามารถเปลี่ยนค่าได้

Code

```
radius = 3  
radius = 5  
print (radius)
```

output

5

ตัวแปรสามารถเปลี่ยนค่าได้

Code

```
radius = 3  
radius = 5  
radius = radius + 1  
print (radius)
```

output

6

ตัวแปรสามารถเปลี่ยนค่าได้และอ้างอิงค่าเก่า

Code

```
radius = 3  
radius = 5  
radius = radius + 1  
print (radius)
```

output

7

ตัวแปรสามารถเปลี่ยนค่าได้และอ้างอิงค่าเก่า

Code

```
radius = 3  
radius = 5  
radius = radius + 1  
radius += 1  
radius -= 5  
print (radius)
```

output

2

Question

ลองหาคำตอบกันดูซิ...ว่าได้อะไรออกมา

Code

```
year = 2  
month = year * 12  
print (month)  
year = year + 1  
print (month)
```

output

24

24

ชนิดของข้อมูล

สตริง หรือ ข้อความ

ใช้เก็บข้อมูลประเภทตัวอักษร จะอยู่ในเครื่องหมาย “ ” หรือ ‘ ’

```
Print (“Hello Python”)
```

Hello Python

```
Print (“3”)
```

3

```
Print (“3.14 + 2.5”)
```

3.14 + 2.5

สตริง หรือ ข้อความ

ความแตกต่างของการมีกับไม่มี “ ”

Print (“3.14 + 2.5”)

Print (3.14 + 2.5)

3.14 + 2.5

5.64

สตริง หรือ ข้อความ

ความแตกต่างของการมีกับไม่มี “ ”

```
Print (“Hello”)
```

```
Hello
```

```
Print (Hello)
```

```
NameError: name 'Hello'  
is not defined
```

สตริง หรือ ข้อความ

ความแตกต่างของการมีกับไม่มี “ ”

```
Print (“Hello”)
```

```
Hello
```

```
Hello = “Python”  
Print (Hello)
```

```
Python
```


สตริง หรือ ข้อความ

สตริง ตัวเลข และการกระทำ

```
a = 3  
b = 5  
Print (a + b)
```

8

```
a = "3"  
b = "5"  
print (a + b)
```

35

สตริง หรือ ข้อความ

สตริง ตัวเลข และการกระทำ

```
a = 3  
b = 5  
Print (a - b)
```

- 2

```
a = "3"  
b = "5"  
print (a - b)
```

```
TypeError: unsupported  
operand type(s) for -:  
'str' and 'str'
```

สตริง หรือ ข้อความ

สตริง ตัวเลข และการกระทำ

```
a = "3"  
b = 5  
print (a + b)
```

```
TypeError: can only  
concatenate str (not "int")  
to str
```

สตริง หรือ ข้อความ

สตริง ตัวเลข และการกระทำ

```
a = "3"  
b = 5  
print (a + str(b))
```

35

สตริง หรือ ข้อความ

สตริง ตัวเลข และการกระทำ

```
a = "3"  
b = 5  
print(type( a ))  
print (int(a) + b)  
print(type( a ))
```

8

สตริง หรือ ข้อความ

สตริง ตัวเลข และการกระทำ + ต่อข้อความ

```
a = "3"  
b = 5  
print ("This is " + a + " and " + str(b))
```

```
This is 3 and 5
```

สตริง หรือ ข้อความ

สตริง ตัวเลข และการกระทำ + ต่อข้อความ

```
a = "3"
```

```
b = 5
```

```
print ("This is %s and %d" % (a, b) )
```

สตริง หรือ ข้อความ

สตริง ตัวเลข และการกระทำ + ต่อข้อความ

```
a = "3"  
b = 5  
print ("This is %s and %d" % (a, b) )
```

```
This is 3 and 5
```


คุณสมบัติของสตรึง

อินเดกซ์ (Index) ของสตริง

สตริง ตัวเลข และการกระทำ + ต่อข้อความ

```
s = "Hello, world"  
print (s[0])
```

H

0	1	2	3	4	5	6	7	8	9	10	11
H	e	l	l	o	,		w	o	r	l	d

อินเดกซ์ (Index) ของสตริง

สตริง ตัวเลข และการกระทำ + ต่อข้อความ

```
s = "Hello, world"  
print (s[7])
```

w

0	1	2	3	4	5	6	7	8	9	10	11
H	e	l	l	o	,		w	o	r	l	d

Question

ลองหาคำตอบกันดูซิ...ว่าได้อะไรออกมา

```
s = "Hello, world"  
print (s[12])
```

```
IndexError: string  
index out of range
```

0	1	2	3	4	5	6	7	8	9	10	11
H	e	l	l	o	,		w	o	r	l	d

ลองหาคำตอบกันดูซิ...ว่าได้อะไรออกมา

```
s = "Hello, world"  
print (s[-1])
```

d

0	1	2	3	4	5	6	7	8	9	10	11
H	e	l	l	o	,		w	o	r	l	d

ลองหาคำตอบกันดูซิ...ว่าได้อะไรออกมา

```
s = "Hello, world"  
print (s[-7])
```

,

0	1	2	3	4	5	6	7	8	9	10	11
H	e	l	l	o	,		w	o	r	l	d

การเข้าถึงบางส่วนของสตริง

การเข้าถึงบางส่วนของสตริง

```
s = "Hello, world"  
print (s[2:5])
```

llo

0	1	2	3	4	5	6	7	8	9	10	11
H	e	l	l	o	,		w	o	r	l	d

การเข้าถึงบางส่วนของสตริง

```
s = "Hello, world"
```

```
print (s[2:5])
```

```
print (s[0:6])
```

llo

0	1	2	3	4	5	6	7	8	9	10	11
H	e	l	l	o	,		w	o	r	l	d

การเข้าถึงบางส่วนของสตริง

```
s = "Hello, world"
```

```
print (s[2:5])
```

```
print (s[0:6])
```

llo

Hello,

0	1	2	3	4	5	6	7	8	9	10	11
H	e	l	l	o	,		w	o	r	l	d

การเข้าถึงบางส่วนของสตริง

```
s = "Hello, world"
```

```
print (s[2:5])
```

```
print (s[0:6])
```

```
print (s[:6])
```

llo

Hello,

0	1	2	3	4	5	6	7	8	9	10	11
H	e	l	l	o	,		w	o	r	l	d

การเข้าถึงบางส่วนของสตริง

```
s = "Hello, world"
```

```
print (s[2:5])
```

```
print (s[0:6])
```

```
print (s[:6])
```

```
llo
```

```
Hello,
```

```
Hello,
```

0	1	2	3	4	5	6	7	8	9	10	11
H	e	l	l	o	,		w	o	r	l	d

การเข้าถึงบางส่วนของสตริง

```
s = "Hello, world"  
print (s[1:11])
```

ello, worl

0	1	2	3	4	5	6	7	8	9	10	11
H	e	l	l	o	,		w	o	r	l	d

การเข้าถึงบางส่วนของสตริง

```
s = "Hello, world"
```

```
print (s[1:11])
```

```
print (s[1:])
```

```
ello, worl
```

0	1	2	3	4	5	6	7	8	9	10	11
H	e	l	l	o	,		w	o	r	l	d

การเข้าถึงบางส่วนของสตริง

```
s = "Hello, world"
```

```
print (s[1:11])
```

```
print (s[1:])
```

```
ello, worl
```

```
ello, world
```

0	1	2	3	4	5	6	7	8	9	10	11
H	e	l	l	o	,		w	o	r	l	d

การเข้าถึงบางส่วนของสตริง

```
s = "Hello, world"
```

```
print (s[1:11])
```

```
print (s[1:])
```

```
print (s[7:-2])
```

```
ello, worl
```

```
ello, world
```

0	1	2	3	4	5	6	7	8	9	10	11
H	e	l	l	o	,		w	o	r	l	d

การเข้าถึงบางส่วนของสตริง

```
s = "Hello, world"
```

```
print (s[1:11])
```

```
print (s[1:])
```

```
print (s[7:-2])
```

```
ello, worl
```

```
ello, world
```

```
wor
```

0	1	2	3	4	5	6	7	8	9	10	11
H	e	l	l	o	,		w	o	r	l	d

Question

ลองหาคำตอบกันดูซิ...ว่าได้อะไรออกมา

```
s = "Hello, world"  
print (.....)
```

world

0	1	2	3	4	5	6	7	8	9	10	11
H	e	l	l	o	,		w	o	r	l	d

ลองหาคำตอบกันดูซิ...ว่าได้อะไรออกมา

```
s = "Hello, world"  
print (s[7:])
```

world

0	1	2	3	4	5	6	7	8	9	10	11
H	e	l	l	o	,		w	o	r	l	d

ลองหาคำตอบกันดูซิ...ว่าได้อะไรออกมา

```
s = "Hello, world"
```

```
print (s[7:])
```

```
print (s[7:12])
```

```
world  
world
```

0	1	2	3	4	5	6	7	8	9	10	11
H	e	l	l	o	,		w	o	r	l	d

การหาความยาวของสตริง

การหาความยาวของสตริง (length)

```
s = "Hello, world"  
print (len(s))
```

12

การค้นหาลำหัดจริง

การค้นหาสตริง (find)

ค้นหาอินเด็กซ์เริ่มต้นของสตริง

```
text = "Hello, world"  
word = "word"  
print (text.find(word))
```

7

การค้นหาสตริง (find)

ค้นหาอินเด็กซ์เริ่มต้นของสตริง

```
text = "Hello, world"  
word = "man"  
print (text.find(word))
```

-1

การค้นหาสตริง (find)

ค้นหาอินเด็กซ์เริ่มต้นและจุดสิ้นสุด

```
text = "Hello, world. This is my world."  
word = "world"  
print (text.find(word, 14))  
print (text.find(word, 14, len(text)))
```

25

การค้นหาคำที่อยู่ที่ติดกัน

การค้นหาสตริงที่อยู่ติดกันในคำถัดไป (find)

```
text = "Today is Sunday. I stay at home and keep coding. "  
word = "keep"  
text = text [ text.find(word) + len (word) + 1 : ]  
next_word = text [ : text.find(" ") ]  
print (next_word)
```

coding.

การค้นหาสตริงที่อยู่ติดกันในคำถัดไป (find)

```
text = "Today is Sunday. I stay at home and keep coding."  
word = "at"  
text = text [ text.find(word) + len (word) + 1 : ]  
next_word = text [ : text.find(" ") ]  
print (next_word)
```

home

การค้นหาสตริงที่อยู่ติดกันในคำถัดไปหลายคำ (find)

```
text = "Today is Sunday. I stay at home and keep coding. Today is  
Monday. I stay at office and keep cleaning but still keep coding. "
```

```
word = "keep"
```

```
start_pos = 0
```

```
text = text [ text.find ( word, start_pos) + len (word) + 1 : ]
```

```
next_word = text [ : text.find(" ") ]
```

```
print (next_word)
```

```
start_pos = text.find (" ") + 1
```

```
coding.
```


การค้นหาสตริงที่อยู่ติดกันในคำถัดไปหลายคำ (find)

```
text = "Today is Sunday. I stay at home and keep coding. Today is  
Monday. I stay at office and keep cleaning but still keep coding. "
```

```
word = "keep"
```

```
text = text [ text.find ( word, start_pos) + len (word) + 1 : ]
```

```
next_word = text [ : text.find(" ") ]
```

```
print (next_word)
```

```
start_pos = text.find (" ") + 1
```

```
cleaning
```

การค้นหาสตริงที่อยู่ติดกันในคำถัดไปหลายคำ (find)

```
text = "Today is Sunday. I stay at home and keep coding. Today is  
Monday. I stay at office and keep cleaning but still keep coding. "
```

```
word = "keep"
```

```
text = text [ text.find ( word, start_pos) + len (word) + 1 : ]
```

```
next_word = text [ : text.find(" ") ]
```

```
print (next_word)
```

```
coding.
```

ฟังก์ชัน

ฟังก์ชัน

```
word = "keep"  
print ("world")  
Word_length = len (word)
```

ฟังก์ชัน

```
pi = 3.14
radius = 3
area = pi * (radius ** 2)
print (area)
new_pi = 3.1416
new_radius = 5
new_area = new_pi * (new_radius ** 2 )
print (new_area)
```

ฟังก์ชัน

```
pi = 3.14
```

```
radius = 3
```

```
area = pi * (radius ** 2)
```

```
print (area)
```

```
new_pi = 3.1416
```

```
new_radius = 5
```

```
new_area = new_pi * (new_radius ** 2)
```

```
print (new_area)
```

ฟังก์ชัน

การประกาศฟังก์ชัน

```
def function_name ( < parameters > ) :  
    < code >  
    return result
```

การเรียกใช้ฟังก์ชัน

```
function_name ( < parameters > ) :
```

ฟังก์ชัน

```
def get_area ( pi, radius ) :  
    return pi * (radius ** 2)
```

```
pi = 3.14
```

```
radius = 3
```

```
area = get_area ( pi, radius)
```

```
print (area)
```

```
new_pi = 3.1416
```

```
new_radius = 5
```

```
new_area = get_area (new_pi, new_radius)
```

```
print (new_area)
```


Question

ฟังก์ชัน

ให้เขียนฟังก์ชันคำนวณพื้นที่สามเหลี่ยม

กำหนดชื่อฟังก์ชันเป็น `get_triangle_area`

รับพารามิเตอร์ 2 ตัว คือ `height`, `base_width`

ค่าที่ต้องส่งคืน คือ ค่าของพื้นที่สามเหลี่ยมตามสูตร $0.5 * \text{ฐาน} * \text{สูง}$

แสดงค่าผลลัพธ์ที่หาได้

ฟังก์ชัน

```
def get_tringle_area ( height, base_width ) :  
    return 0.5 * base_width * height
```

```
height = 10
```

```
base_width = 5
```

```
print(get_tringle_area(height, base_width))
```

25.0

การค้นหาสตริง (find)

ค้นหาอินเด็กซ์เริ่มต้นของสตริง

```
text = "Hello, world"  
word = "world"  
print (text.find(word))
```

7

การค้นหาสตริง (find)

ค้นหาอินเด็กซ์เริ่มต้นของสตริง

```
text = "Hello, world"  
word = "man"  
print (text.find(word))
```

-1

การค้นหาสตริง (find)

ค้นหาอินเด็กซ์เริ่มต้นและจุดสิ้นสุด

```
text = "Hello, world. This is my world."  
word = "world"  
print (text.find(word, 14))  
print (text.find(word, 14, len(text)))
```

```
25  
25
```

การค้นหาคำที่อยู่ที่ติดกัน

การค้นหาสตริงที่อยู่ติดกันในคำถัดไป (find)

```
text = "Today is Sunday. I stay at home and keep coding."  
word = "keep"  
text = text [ text.find(word) + len (word) + 1 : ]  
next_word = text [ 0: text.find(" ") ]  
print (next_word)
```

coding.

การค้นหาสตริงที่อยู่ติดกันในคำถัดไป (find)

```
text = "Today is Sunday. I stay at home and keep coding."  
word = "at"  
text = text [ text.find(word) + len (word) + 1 : ]  
next_word = text [ : text.find(" ") ]  
print (next_word)
```

home

การค้นหาสตริงที่อยู่ติดกันในคำถัดไปหลายคำ (find)

```
text = "Today is Sunday. I stay at home and keep coding. Today is  
Monday. I stay at office and keep cleaning but still keep coding. "
```

```
word = "keep"
```

```
start_pos = 0
```

```
text = text [ text.find ( word, start_pos) + len (word) + 1 : ]
```

```
next_word = text [ : text.find(" ") ]
```

```
print (next_word)
```

```
start_pos = text.find (" ") + 1
```

```
coding.
```

การค้นหาสตริงที่อยู่ติดกันในคำถัดไปหลายคำ (find)

```
text = "Today is Sunday. I stay at home and keep coding. Today is  
Monday. I stay at office and keep cleaning but still keep coding. "
```

```
word = "keep"
```

```
text = text [ text.find ( word, start_pos) + len (word) + 1 : ]
```

```
next_word = text [ : text.find(" ") ]
```

```
print (next_word)
```

```
start_pos = text.find (" ") + 1
```

```
cleaning
```

การค้นหาสตริงที่อยู่ติดกันในคำถัดไปหลายคำ (find)

```
text = "Today is Sunday. I stay at home and keep coding. Today is  
Monday. I stay at office and keep cleaning but still keep coding. "
```

```
word = "keep"
```

```
text = text [ text.find ( word, start_pos) + len (word) + 1 : ]
```

```
next_word = text [ : text.find(" ") ]
```

```
print (next_word)
```

```
coding.
```

ฟังก์ชัน

ฟังก์ชัน

```
word = "keep"  
print ("world")  
Word_length = len (word)
```

ฟังก์ชัน

```
pi = 3.14
radius = 3
area = pi * (radius ** 2)
print (area)
new_pi = 3.1416
new_radius = 5
new_area = new_pi * (new_radius ** 2 )
print (new_area)
```

ฟังก์ชัน

```
pi = 3.14
```

```
radius = 3
```

```
area = pi * (radius ** 2)
```

```
print (area)
```

```
new_pi = 3.1416
```

```
new_radius = 5
```

```
new_area = new_pi * (new_radius ** 2 )
```

```
print (new_area)
```


ฟังก์ชัน

การประกาศฟังก์ชัน

```
def function_name ( < parameters > ) :  
    < code >  
    return result
```

การเรียกใช้ฟังก์ชัน

```
function_name ( < parameters > ) :
```

ฟังก์ชัน

```
def get_area ( pi, radius ) :  
    return pi * (radius ** 2)
```

```
pi = 3.14
```

```
radius = 3
```

```
area = get_area ( pi, radius)
```

```
print (area)
```

```
new_pi = 3.1416
```

```
new_radius = 5
```

```
new_area = get_area (new_pi, new_radius)
```

```
print (new_area)
```

Question

ฟังก์ชัน

ให้เขียนฟังก์ชันคำนวณพื้นที่สามเหลี่ยม

กำหนดชื่อฟังก์ชันเป็น `get_triangle_area`

รับพารามิเตอร์ 2 ตัว คือ `height`, `base_width`

ค่าที่ต้องส่งคืน คือ ค่าของพื้นที่สามเหลี่ยมตามสูตร $0.5 * \text{ฐาน} * \text{สูง}$

แสดงค่าผลลัพธ์ที่หาได้

ฟังก์ชัน

```
def get_tringle_area ( height, base_width ) :  
    return 0.5 * base_width * height
```

```
height = 10
```

```
base_width = 5
```

```
print(get_tringle_area(height, base_width))
```

```
25.0
```

ขอบเขตตัวแปร

ขอบเขตของตัวแปร (การโยนค่าพารามิเตอร์)

```
def get_area(pi, radius):  
    return pi * (radius ** 2)
```

```
pi = 3.14
```

```
radius = 3
```

```
area = get_area(3.1416, 5)
```

```
print(area)
```

ขอบเขตของตัวแปร (การโยนค่าพารามิเตอร์)

```
def get_area(pi, radius):  
    return pi * (radius ** 2)
```

```
pi = 3.14
```

```
radius = 3
```

```
area = get_area(3.1416, 5)
```

```
print(area)
```



ขอบเขตของตัวแปร (การโยนค่าพารามิเตอร์)

```
def get_area(pi, radius):  
    print("in function")  
    print("pi = %0.4f, radius = %d" % (pi, radius))  
    return pi * (radius ** 2)
```

```
pi = 3.14  
radius = 3  
area = get_area(3.1416, 5)
```

ขอบเขตของตัวแปร (การโยนค่าพารามิเตอร์)

```
def get_area(pi, radius):  
    print("in function")  
    print("pi = %0.4f, radius = %d" % (pi, radius))  
    return pi * (radius ** 2)
```



```
pi = 3.14
```

```
radius = 3
```

```
area = get_area(3.1416, 5)
```

```
print("after function")
```

```
print("pi = %0.4f, radius = %d" % (pi, radius))
```

ขอบเขตของตัวแปร (การโยนค่าพารามิเตอร์)

```
def get_area(pi, radius):  
    print("in function")  
    print("pi = %0.4f, radius = %d" % (pi, radius))  
    return pi * (radius ** 2)
```

pi = 3.14

radius = 3

area = get_area(3.1416, 5)

print("after function")

print("pi = %0.4f, radius = %d" % (pi, radius))

in function

pi = 3.1416, radius = 5

after function

pi = 3.1400, radius = 3

ขอบเขตของตัวแปร (การโยนค่าพารามิเตอร์)

```
def get_area(pi, radius):  
    radius += 1  
    print("in function")  
    print("pi = %0.4f, radius = %d" % (pi, radius))  
    return pi * (radius ** 2)
```

```
pi = 3.14  
radius = 3  
area = get_area(3.1416, 5)  
print("after function")  
print("pi = %0.4f, radius = %d" % (pi, radius))
```

```
in function  
pi = 3.1416, radius = 6  
after function  
pi = 3.1400, radius = 3
```

ขอบเขตของตัวแปร (การโยนค่าพารามิเตอร์)

```
def get_area(pi):  
    print("in function")  
    print("radius = %d" % (radius))  
    return pi * (radius ** 2)  
pi = 3.14  
radius = 3  
area = get_area(3.1416)  
print("after function")  
print("radius = %d" % (radius))
```

```
in function  
radius = 3  
after function  
radius = 3
```

ขอบเขตของตัวแปร (การโยนค่าพารามิเตอร์)

```
def get_area(pi):          Local variable
    print("in function")
    print("radius = %d" % (radius))
    return pi * (radius ** 2)
```

```
pi = 3.14                  Global variable
```

```
radius = 3
```

```
area = get_area(3.1416)
```

```
print("after function")
```

```
print("radius = %d" % (radius))
```

in function

radius = 3

after function

radius = 3

การอ้างอิงตัวแปร

1. มองหาค่าจาก Local variable ก่อน

2. ถ้าไม่เจอ ไปมองหาค่าจาก Global variable

สร้างฟังก์ชันหาคำถัดไป

```
def find_next_word(text, word, start_pos):  
    text = text[text.find(word, start_pos) + len(word) + 1:]  
    next_word = text[: text.find(" ")]  
    print(next_word)  
    start_pos = text.find(" ") + 1  
    return text, start_pos
```

```
text = "Today is Sunday. I stay at home and keep coding. Today is  
Monday. I stay, at office and keep cleaning but still keep coding. "
```

```
word = "keep"
```

```
start_pos = 0
```

```
text, start_pos = find_next_word(text, word, start_pos)
```

```
text, start_pos = find_next_word(text, word, start_pos)
```

```
text, start_pos = find_next_word(text, word, start_pos)
```

เงื่อนไข

เงื่อนไข ความเป็นจริงไม่มีเว้นวรรค

```
text = "I keep coding "  
word = "keep"  
text = text[text.find(word) + len(word) + 1:]  
next_word = text[:text.find(" ")]  
print(next_word)
```

```
text = "coding "  
text.find(" ") = 6
```

เจอ " " จะส่ง -1 กลับมา

coding

เงื่อนไข ความเป็นจริงไม่มีเว้นวรรค

```
text = "I keep coding"  
word = "keep"  
text = text[text.find(word) + len(word) + 1:]  
next_word = text[:text.find(" ")]  
print(next_word)
```

```
text = "coding"  
text.find(" ") = -1    ไม่เจอ " " จะส่ง -1 กลับมา
```

codin

เงื่อนไข if

```
if (<Condition>):  
    <Block 1>
```

เงื่อนไข if

```
if (<Condition>):  
    <Block 1>  
else:  
    <Block 2>
```

เงื่อนไข if

```
if (<Condition 1>):  
    <Block 1>  
elif (<Condition 2>) :  
    <Block 2>  
else:  
    <Block 3>
```

เงื่อนไข if

```
if (<Condition 1>):  
    <Block 1>  
elif (<Condition 2>):  
    <Block 2>  
elif (<Condition 3>):  
    <Block 3>  
else:  
    <Block 4>
```

เงื่อนไข if

```
if (<Condition 1>):  
    <Block 1>  
elif (<Condition 2>):  
    <Block 2>  
elif (<Condition 3>):  
    <Block 3>  
else:  
    <Block 4>
```

เงื่อนไข if การหาคำสุดท้าย -1

```
next_word = text[:text.find(" ")]
```

```
if (text.find(" ") != -1):  
    next_word = text[:text.find(" ")]  
else:  
    next_word = text
```

==, !=, <, >, <=, >=

การเชื่อมเงื่อนไข

```
if (<Condition 1> and <Condition 2> and <Condition 3>):  
    <Block 1>
```

```
if (<Condition 1> or <Condition 2> or <Condition 3>):  
    <Block 1>
```

การเชื่อมเงื่อนไข

if (a >= 5 and a <= 10) :

if (a == 5 or a == 10) :

if (a == 5 or a == 10 or a == 15) :

if ((a == 5) or (a >= 10 and a <= 15)) :

เงื่อนไขซ้อนเงื่อนไข

```
if (<Condition 1>):  
    if (<Condition 2>):  
        <Block 1>  
    else (<Condition 3>):  
        <Block 2>  
  
else:  
    <Block 3>
```

Question

ลองเขียนดูซิ.....

กำหนดตัวแปร Score เก็บตัวเลขคะแนนสอบไว้ ให้เขียนโปรแกรมคำนวณเกรด โดยโปรแกรมจะ พิมพ์เกรดที่ได้ออกทางหน้าจอ

วิธีการคำนวณเป็นดังนี้

ถ้าได้ตั้งแต่ 80 คะแนนขึ้นไป ให้เกรด "A"

ถ้าได้ตั้งแต่ 70 คะแนนขึ้นไป แต่น้อยกว่า 80 ให้เกรด "B"

ถ้าได้ตั้งแต่ 60 คะแนนขึ้นไป แต่น้อยกว่า 70 ให้เกรด "C"

ถ้าได้ตั้งแต่ 50 คะแนนขึ้นไป แต่น้อยกว่า 60 ให้เกรด "D"

ถ้าได้ต่ำกว่า 50 คะแนน ให้เกรด "F"

ลองเขียนดูซิ.....

```
score = 80
```

```
if (score >= 80):  
    print("A")
```

```
elif (score >= 70 and score < 80 ):  
    print("B")
```

```
elif (score >= 60 and score < 70 ):  
    print ("C")
```

```
elif (score >= 50 and score < 60 ):  
    print("D")
```

```
else:  
    print("F")
```

ลองเขียนดูซิ.....

```
score = 80
```

```
if (score >= 80):
```

```
    print("A")
```

```
elif (70 <= score < 80 ):
```

```
    print("B")
```

```
elif (60 <= score < 70 ):
```

```
    print("C")
```

```
elif (50 <= score < 60 ):
```

```
    print("D")
```

```
else:
```

```
    print("F")
```

ลองเขียนดูซิ.....

```
score = 80
```

```
if (score >= 80):
```

```
    print("A")
```

```
elif (score >= 70):
```

```
    print ("B"),
```

```
elif (score >= 60):
```

```
    print("C")
```

```
elif (score >= 50):
```

```
    print("D")
```

```
else:
```

```
    print("F")
```


Question

ลองเขียนดูซิ.....

```
def find_next_word(text, word, start_pos):
```

1. แสดงค่าคำที่อยู่ถัดจาก word ใน text โดยเริ่มค้นหาจากตำแหน่ง start_pos และ return ค่าตำแหน่งของจุดเริ่มหาถัดไป แต่ถ้าสิ้นสุด text แล้วให้ return -1
2. ถ้าไม่มี word ปรากฏใน text ให้แสดงสตริงว่าง “ ” แล้ว return -1

ลองเขียนดูซิ..... สร้างตัวแปรใหม่

```
def find_next_word(text, word, start_pos):  
    text = text[text.find(word, start_pos) + len(word) + 1:]  
    next_word = text[: text.find(' ')]  
    print(next_word)  
    start_pos = text.find(' ') + 1  
    return text, start_pos
```

ลองเขียนดูซิ..... สร้างตัวแปรใหม่

```
def find_next_word(text, word, start_pos):  
    word_pos = text.find(word, start_pos)  
    text = text[word_pos + len(word) + 1:]  
    next_word = text[: text.find(' ')]  
    print(next_word)  
    start_pos = text.find(' ') + 1  
    return text, start_pos
```

ลองเขียนดูซิ.....เช็คเงื่อนไขว่าตำแหน่งที่หาไม่เท่ากับ -1

```
def find_next_word(text, word, start_pos):  
    word_pos = text.find(word, start_pos)  
    if (word_pos != -1):          ถ้าเจอ  
        text = text[word_pos + len(word) + 1:]  
        next_word = text[: text.find(' ')]  
        print(next_word)  
        start_pos = text.find(' ') + 1  
        return text, start_pos  
    else:                          ถ้าไม่เจอ  
        print(" ")  
        return " ", -1
```

ลองเขียนดูซิ.....ยังมีช่องว่างอยู่หรือไม่

```
def find_next_word(text, word, start_pos):  
    word_pos = text.find(word, start_pos)  
    if (word_pos != -1):  
        text = text[word_pos + len(word) + 1:]  
        if (text.find(' ') != -1):    ถ้ายังมีอยู่ยังไม่ใช่คำสุดท้าย  
            next_word = text[: text.find(' ')]  
            print(next_word)  
            start_pos = text.find(' ') + 1  
            return text, start_pos  
    else:  
        print(" ")  
        return -1
```

ลองเขียนดูซิ.....ยังมีช่องว่างอยู่หรือไม่

```
def find_next_word(text, word, start_pos):
    word_pos = text.find(word, start_pos)
    if (word_pos != -1):
        text = text[word_pos + len(word) + 1:]
        if (text.find(' ') != -1):
            # ถ้ายังมีอยู่ยังไม่ใช่คำสุดท้าย
            next_word = text[: text.find(' ')]
            print(next_word)
            start_pos = text.find(' ') + 1
            return text, start_pos
        else:
            # ถ้าเท่ากับ -1
            next_word = text
            print(next_word)
            start_pos = -1
            return text, start_pos
    else:
        print(" ")
        return -1
```

while Loop

การวน loop while ตราบที่เงื่อนไขเป็นจริง

```
while (<Condition>):  
    <Block>  
next_command
```

การวน loop while

```
a = 2
```

```
b = 1
```

```
while (a <= 12):
```

```
    print(str(a) + "X" + str(b) + "=" + str(a * b))
```

```
    b = b + 1
```

```
    if (b > 12):
```

```
        a = a + 1
```

```
        b = 1
```

การ break while

การ break while

```
a = 2
b = 1
while (True) :
    print(str(a) + "X" + str(b) + "=" + str(a * b))
    b = b + 1
    if (b > 12) :
        a = a + 1
        b = 1
    if (a > 12) :
        break
```

ข้อมูล List

โครงสร้างข้อมูล List

สตริง – เก็บลำดับของตัวอักษร (sequence of characters)

```
s = "Hello World"
```

ลิสต์ - เก็บลำดับของอะไรก็ได้

```
l = [1, 2.5, "a" , "hello"]
```

โครงสร้างข้อมูล List

สตริง – เก็บลำดับของตัวอักษร (sequence of characters)

```
s = "Hello World"  
print(s[2])
```

l

ลิสต์ - เก็บลำดับของอะไรก็ได้

```
l = [1, 2.5, "a", "hello"]  
print(l[2])
```

a

โครงสร้างข้อมูล List

สตริง – เก็บลำดับของตัวอักษร (sequence of characters)

```
s = "Hello World"  
print(s[2])  
print(s[1:3])
```

```
l  
el
```

ลิสต์ - เก็บลำดับของอะไรก็ได้

```
l = [1, 2.5, "a", "hello"]  
print(l[2])  
print(l[1:3])
```

```
a  
[2.5, "a"]
```


Question

โครงสร้างข้อมูล List

ให้สร้างตัวแปรประเภทลิสต์ชื่อ month เพื่อเก็บชื่อย่อเดือน
ทั้ง 12 เดือน

'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep',
'Oct', 'Nov', 'Dec'

```
month = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug',  
, 'Sep', 'Oct', 'Nov', 'Dec']
```

แก้ไขข้อมูลใน List

แก้ไขข้อมูลใน List

การแก้ไขข้อมูลในสตริง

```
s = "Hello World"  
s[0] = 'M'  
print(s)
```

```
Traceback (most recent call last):  
File ".....", line 2, in <module>  
s[0] = 'M'  
TypeError: 'str' object does not support  
item assignment
```

แก้ไขข้อมูลใน List

การแก้ไขข้อมูลใน List

```
l = [1, 2.5]  
l[0] = 'M'  
print (l)
```

```
['M', 2.5]
```

การต่อ List

การต่อ List

`<name list>.append(<element list>)`

```
l = [1, 2]  
l.append(3)  
print(l)
```

```
[1,2,3]
```

ความยาว List

ความยาว List

len(<name list>)

```
l = [1, 2]  
print(len(l))
```

2

ความยาว List

len(<name list>)

```
l = []  
print(len(l))
```

0

For loop

For loop

การวนลูป for ในลิสต์

```
weekdays = ['Monday', 'Tuesday', 'Wednesday',  
            'Thursday', 'Friday']
```

ชื่อตัวแปร ชื่อ list

```
for day in weekdays:  
    print("Today is " + day)
```

```
Today is Monday  
Today is Tuesday  
.....
```

For loop

การวนลูปคำใน String

```
text = "Today is Sunday. I stay at home and keep  
coding. Today is Monday. I stay at office and keep  
cleaning."
```

For loop

การวนลูปคำใน String

```
text = "Today is Sunday. I stay at home and keep  
coding. Today is Monday. I stay at office and keep  
cleaning."  
splitted_text = text.split()  
print(splitted_text)
```

```
['Today', 'is', 'Sunday.', 'I', 'stay', 'at', 'home',  
'and', 'keep', 'coding.', 'Today', 'is', 'Monday.', 'I', 'stay',  
'at', 'office', 'and', 'keep', 'cleaning.']
```

Question

For loop

การวนลูปคำใน String

```
text = "Today is Sunday. I stay at home and keep  
coding. Today is Monday. I stay at office and keep  
cleaning."  
splitted_text = text.split()  
print(splitted_text)  
print(splitted_text[2])  
print(splitted_text[300])
```


For loop

การวนลูปคำใน String

```
['Today', 'is', 'Sunday.', 'I', 'stay', 'at', 'home', 'and', 'keep',  
'coding.', 'Today', 'is', 'Monday.',  
'I', 'stay', 'at', 'office', 'and', 'keep', 'cleaning.']
```

Sunday

Traceback (most recent call last):

File ..., line 5, in <module>

For loop

การวนลูปคำใน String

```
text = "Today is Sunday. I stay at home and keep  
coding. Today is Monday. I stay at office and keep  
cleaning."  
splitted_text = text.split()  
for word in splitted_text:  
    print(word)
```

For loop

การวนลูปคำใน String

```
Today  
is  
Sunday.  
...  
..  
keep  
cleaning.
```

For loop

การวนลูป for กับ range()

```
for i in range(5) :  
    print(i)
```

```
0  
1  
2  
3  
4
```

Question

ลองดูซิ.....

จงแสดงเลขคู่ในช่วง 0 ถึง 10 (รวม 10 ด้วย)

ลองดูซิ.....

การวนลูป for กับ range()

```
for i in range(11):  
    if (i % 2 == 0):  
        print(i)
```

```
0  
2  
...  
...  
10
```

input

Input String

```
name = input("Please enter your name: ")  
print("Your name is " + name)
```

Please enter your name:

Input Integer

```
num = input ("Please enter a number: ")  
print(num + 5)
```

```
Please enter a number: 10<enter>  
Traceback (most recent call last):  
File ..., line 2, in <module>  
print(num + 5)  
TypeError: Can't convert 'int' object to str implicitly
```

Input Integer

```
num = input("Please enter a number: ")  
print(int(num) + 5)
```

```
Please enter a number: 10<enter>  
15
```

Input Integer

```
num = int(input("Please enter a number: "))  
print(num + 5)
```

```
Please enter a number: 10<enter>  
15
```