



Mahidol University  
*Wisdom of the Land*



# Lecture 02: Python Basics

Lecturers:

Aj. Jidapa Kraisangka

Aj. Akara Supratak

Aj. Tipajin Thaipsisutikul



- RECAP: Intro to Python
- Expression
  - Arithmetic operation
  - Precedence
- Common Data Type in Python
- Variable and Assignment
- Write a Simple Python Program



Mahidol University  
*Wisdom of the Land*



# RECAP: Intro to Python



- Python เป็นภาษาระดับสูงที่ใช้ได้ทั่วไป เช่น
  - **Computation (การคำนวณ)**
  - Data Science (การใช้วิทยาศาสตร์ข้อมูล)
  - Web programming (การเขียนโปรแกรมบนเว็บ)etc.
- Python is Interpreted – โปรแกรมจะถูกแปลและ run ทีละคำสั่ง
- Python is Interactive – สามารถใช้ในลักษณะ prompt เพื่อเขียนและ run คำสั่งได้ทันที

Ref: [https://www.tutorialspoint.com/python/python\\_overview.htm](https://www.tutorialspoint.com/python/python_overview.htm)



helloworld.py

Source code

```
1. print('Hello, ')\n2. print('World')\n3. print('This is my ')\n4. print('first program')
```

Python จะถูกแปลและ run ทีละ  
คำสั่ง

**Statement:** คำสั่งซึ่งอาจจะเป็น 1 คำสั่งหรือ  
ประกอบด้วยชุดคำสั่งหลาย ๆ คำสั่ง

print() เป็นคำสั่งที่ใช้แสดงผลลัพธ์

Output (Console)



## helloworld2.py

```
1. print('Hello, ')
2. print('World')
3. # print('This is my ')
4. print('first program')
5. ''' This is how I comment
6.     several lines of code
7.     ๔•๕•๖•๗
8. '''
```

**Comment** เป็นบอก Interpreter ไม่ให้ run คำสั่งในบรรทัดนั้น ๆ

นักเขียนโปรแกรม สามารถลืมนิดของตัวเองได้ จึงจำเป็นต้องมีการเขียน Comment

- Single line comment: การ Comment บรรทัดเดียวใช้สัญลักษณ์ **#**
- Block comments: การ Comment หลายบรรทัด เริ่มจากการพิมพ์ **'''** ตามด้วยสิ่งที่จะ Comment และปิดด้วย **'''**



## 1 Write a Source Code

helloworld2.py

```
1. print('Hello, ')  
2. print('World')  
3. # print('This is my ')  
4. print('first program')  
5. ''' This is how I comment  
6.     several lines of code  
7.     ๕•๕•?๓♡  
8. '''
```

## 2 Run a Program

## 3 See the output

```
Hello,  
World  
first program
```

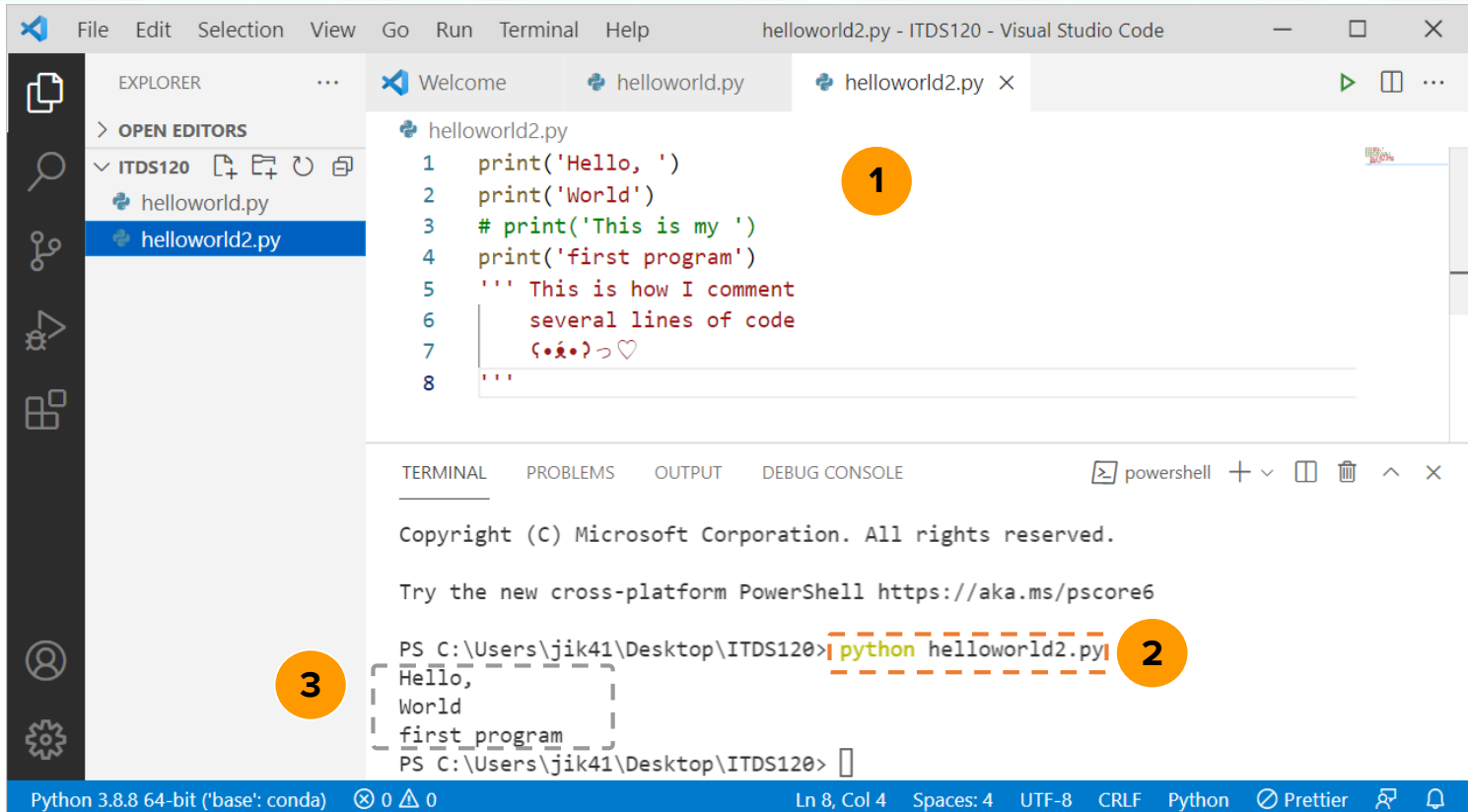


The screenshot shows the Replit IDE interface. At the top, there is a navigation bar with a menu icon, a logo, the text "itds120 / Lab/", a Python icon, a refresh icon, a play button (highlighted with a dashed orange box and a yellow circle with the number 2), and an "Edit project" button. Below the navigation bar, the main workspace is divided into two panels. The left panel shows a file named "main.py" with a list of icons on the left margin. The code in "main.py" is as follows:

```
1 print('Hello, ')\n2 print('World')\n3 # print('This is my ')\n4 print('first program')\n5 ''' This is how I comment\n6     several lines of code\n7     (｡•́•̀｡)っ♡\n8 '''
```

The right panel has tabs for "Console", "Shell", and "Markdown". The "Console" tab is active, showing the output of the program: "Hello, World" and "first program" on separate lines, followed by a prompt character. A dashed orange box highlights the output, and a yellow circle with the number 3 is placed next to it.







- สามารถ run คำสั่ง Python ใน Command prompt ได้ เช่น **Anaconda Prompt** ซึ่งเป็น Python interpreter
- เมื่อเปิด terminal หรือเปิด Anaconda Prompt แล้วพิมพ์ python จะมีสัญลักษณ์ **>>>** เพื่อให้พร้อมใช้งาน

```
Anaconda Prompt (Miniconda3) - python
(base) C:\Users\jik41>python
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915
64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more
information.
>>>
```



Type  $5+4+3+2+1$  then press “enter” in the console

```
Anaconda Prompt (Miniconda3) - python

(base) C:\Users\jik41>python
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915
64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more
information.
>>> 5+4+3+2+1
15
>>>
```



```
Anaconda Prompt (Miniconda3) - python

(base) C:\Users\jik41>python
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 5+4+3+2+1
15
>>> print('This is in Python interpreter')
This is in Python interpreter
>>> 2*2*2*2*2
32
>>> (8+3)/2
5.5
>>>
```

• สามารถพิมพ์คำสั่ง Python e.g. `print()` อื่น ๆ ใน console ได้

• Arithmetic **expressions** หรือ นิพจน์คณิตศาสตร์อื่น ๆ



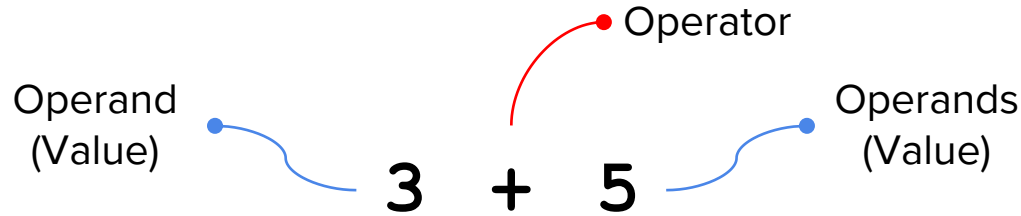
Mahidol University  
*Wisdom of the Land*



# Expression



- Expression เป็นกลุ่มของค่าต่าง ๆ (values, e.g 3, 5) มากระทำต่อกันโดยใช้ตัวดำเนินการคำนวณ (operators, e.g. +)



- ผลจากการคำนวณจาก Expression จะเป็นค่าผลลัพธ์เพียงค่าเดียว



- Arithmetic expressions คือ expressions ที่ใช้ตัวดำเนินการทางคณิตศาสตร์และตัวเลข (e.g  $2 + 2$ )
- เมื่อกล่าวถึง **data type** ของค่าตัวเลขสามารถแบ่งได้ ดังนี้
  - Integers หรือ **int**: จำนวนเต็ม  
Example:  $-2, -1, 0, 1, 2$
  - Floating-point numbers หรือ **float**: จำนวนจริงที่มีจุดทศนิยม  
Example:  $-2.59, 0.0, 0.5, 3.14159$



- Addition (การบวก)

$2 + 2$	$=$	
$-12.5 + 24.2$	$=$	
$4.0 + 1$	$=$	

- Subtraction (การลบ)

$2 - 2$	$=$	
$-12.5 - 24.2$	$=$	
$4.0 - 1$	$=$	





- Multiplication (การคูณ)

$$2 * 2 =$$

$$-2.5 * 1.0 =$$

$$4 * 5.0 =$$

- Exponent (ยกกำลัง)

$$-2 ** 3 =$$

$$5.0 ** 2 =$$

$$2 ** 0.5 =$$



- Division (การหาร)

$$5 / 2 =$$

$$5 / 5 =$$

$$5 / 5.0 =$$

$$2 / 5.0 =$$

- Integer division (การหารเลขจำนวนเต็ม)

$$5 // 2 =$$

$$5.0 // 2.0 =$$

$$2 // 5.0 =$$



- Modulus (การหาเศษจากการหาร)

$$5 \% 2 = 1$$

$$60 \% 10 =$$

$$22 \% 7 =$$

$$7 \% 10 =$$

$$10 \% 2 =$$



คำถาม: หากเรามีตัวดำเนินการทางคณิตศาสตร์หลายตัว จะทำอย่างไร

$$5 * 0 + 4 - 3 // 2 ** 2 + 7$$

เราจะเรียงลำดับการคำนวณ (**order of operation**) ของโจทย์นี้  
อย่างไร

Order of operations = Precedence



**Table 1-1:** Math Operators from Highest to Lowest Precedence

Operator	Operation	Example	Evaluates to . . .
**	Exponent	2 ** 3	8
%	Modulus/remainder	22 % 8	6
//	Integer division/floored quotient	22 // 8	2
/	Division	22 / 8	2.75
*	Multiplication	3 * 5	15
-	Subtraction	5 - 2	3
+	Addition	2 + 2	4



**Exercise:**       $5 * 0 + 4 - 3 // 2 ** 2 + 7$

**Step 1:**

Expression:  $5 * 0 + 4 - 3 // 2 ** 2 + 7$

เมื่อทำจากซ้ายไปขวา จะเห็นได้ว่า **\*\*** มีลำดับความสำคัญสูงสุด;

Result:                       $5 * 0 + 4 - 3 // 4$  (Highest precedence)

**Step 2:**

Expression:  $5 * 0 + 4 - 3 // 4 + 7$

เมื่อทำจากซ้ายไปขวา จะเห็นได้ว่า **//** มีลำดับความสำคัญสูงสุด

Result:                       $5 * 0 + 4 - 0 + 7$



## Step 3:

Expression:  $5 * 0 + 4 - 0 + 7$

เมื่อทำจากซ้ายไปขวา จะเห็นได้ว่า  $*$  มีลำดับความสำคัญสูงสุด;

Result:  $0 + 4 - 0 + 7$

Step 4:  $0 + 4 - 0 + 7$

Step 5:  $0 + 4 + 7$

Step 6:  $4 + 7$

Final result:  $11$



**Exercise Q1:**     $5 - 1 \% 4 * 3$

**Exercise Q2:**     $2 * 3 ** 2 // 3 / 3$





```
Anaconda Prompt (Miniconda3) - python

(base) C:\Users\jik41>python
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915
64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more
information.
>>> 5 - 1 % 4 * 3
2
>>> 2 * 3 ** 2 // 3 / 3
2.0
>>>
```



- ห้ามใส่ตัวดำเนินการ 2 ตัวข้าง ๆ กัน

```
>>> 5 + ** 3
      File "<stdin>", line 1
        5 + ** 3
            ^
      SyntaxError: invalid syntax
```

- สามารถใช้วงเล็บ ( ) ในการแบ่งกลุ่มการคำนวณได้

```
>>> ((5 + 3)**2)//4
16
```



Mahidol University  
*Wisdom of the Land*



# Common Data Type in Python



## Data Type เป็นการจำแนกประเภทของค่าข้อมูล

**Table 1-2: Common Data Types**

Data type	Examples
Integers	-2, -1, 0, 1, 2, 3, 4, 5
Floating-point numbers	-1.25, -1.0, -0.5, 0.0, 0.5, 1.0, 1.25
Strings	'a', 'aa', 'aaa', 'Hello!', '11 cats'

ตัวเลข

ข้อความ

Table from <https://automatetheboringstuff.com/2e/chapter1/>



- การสร้างค่า String ใน Python ข้อความนั้น ๆ ต้องอยู่ระหว่าง Single quote ( ' ) เสมอ

**Example:**

```
'I am a string'
```

```
'2'
```

```
'3+5//5 is an expression'
```

```
>>> 'Hello'
'Hello'
>>> 'Hello'*5
'HelloHelloHelloHelloHello'
>>> 'Hello' + 'World'
'HelloWorld'
>>> 'ITDS120' + ' ' + 'Fundamentals of Programming'
'ITDS120 Fundamentals of Programming'
```



```
>>> 1+1
2
>>> '1'+'1'
'11'

>>> 1+2
[REDACTED]

>>> '1'+'2'
[REDACTED]

>>> '1'+' '+'2'
[REDACTED]
```

```
>>> 2.5*2
5.0
>>> '2.5'*2
'2.52.5'

[REDACTED]

>>> '4.0'*3
[REDACTED]
```

**\*\* สามารถทดสอบผลลัพธ์ของ Expression เหล่านี้  
ใน Python Interpreter ได้ :)**



# Variable and Assignment




- **Variable** หรือ ตัวแปร เปรียบเสมือนกล่องที่ไว้เก็บค่าของข้อมูลในคอมพิวเตอร์ ซึ่งสามารถเก็บในค่าของ Expression หรือ input จากผู้ใช้
- กฎการตั้งชื่อตัวแปรใน Python ได้แก่
  - ประกอบด้วย alphanumeric characters และ underscores (A-z, 0-9, and \_)
  - ห้ามเริ่มต้นด้วยตัวเลข
  - เป็น **Case-sensitive** หมายถึง ตัวพิมพ์ใหญ่และตัวพิมพ์เล็กถือว่าแตกต่างกัน เช่น Apple  $\neq$  apple  $\neq$  ApPLe  $\neq$  aPPLe
- Example: `_mynumber`, `input2`, `weight`, `height`





- ในการสร้างตัวแปร สามารถทำได้โดยพิมพ์ชื่อตัวแปร ใส่ Assignment ( = ) และค่าเริ่มต้นของตัวแปรนั้น ๆ
- ค่าเริ่มต้นของตัวแปร จะเป็นค่าใด ๆ ประเภทใดก็ได้ ได้แก่ Expression หรือ Value หรือ ตัวแปร

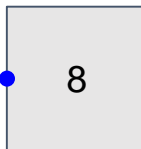
Variable  **myvar**   Value (Type: int)

-  เป็นตัวดำเนินการการใส่ค่า เรียกว่า **assignment operator** โดยจะทำการใส่ค่าที่ประมวลได้จากด้านขวา ใส่ตัวแปรทางด้านซ้าย



```
>>> my_var=3+5  
>>> print(my_var)  
8
```

my\_var



สร้างตัวแปร my\_var และใส่ค่าเริ่มต้นเป็น Expression ซึ่งมีค่าเป็น 8

```
>>> my_var2='Hello'  
>>> print(my_var2)  
Hello
```

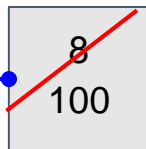
my\_var2



สร้างตัวแปร my\_var2 และใส่ค่าเริ่มต้นเป็น 'Hello' ซึ่งเป็น String

```
>>> my_var=100  
>>> print(my_var)  
100
```

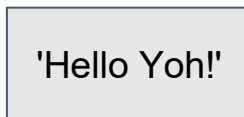
my\_var



my\_var ถูกแทนค่า (Reassignment) ด้วยค่า 100

```
>>> my_var2 = my_var2 + ' Yoh!'  
>>> print(my_var2)  
Hello Yoh!
```

my\_var2



my\_var2 = **my\_var2** + ' Yoh!'  
my\_var2 = 'Hello Yoh!'

'Hello'



Mahidol University  
*Wisdom of the Land*



# Basic Input / Output



- สามารถใช้คำสั่ง `input()` เพื่อรับค่า input จากผู้ใช้เพื่อเก็บค่านั้น ๆ ในตัวแปร

```
>>> myname = input()  
_
```

myname

- `input()` **จะรอ** ข้อความจาก Keyboard จนกว่าผู้ใช้จะกด ENTER
- `myname` จะ **เก็บค่า** จาก Keyboard ในรูปแบบ String



- สามารถใช้คำสั่ง `print()` เพื่อแสดงผลลัพธ์ (เช่น ค่าต่าง ๆ , expression, หรือตัวแปร) ที่ใส่ไว้ในวงเล็บบนจอ console

```
>>> print(2+5)
7
>>> print('2+5')
2+5
>>> print(my_var)
100
>>> print(my_var+2)
102
>>> print(my_var2)
Hello Yoh!
>>> print(my_var2*2)
Hello Yoh!Hello Yoh!
```

my\_var

100

my\_var2

'Hello Yoh!'



```
>>> myname = input()
Jidapa
>>> print('Hello ' + myname)
Hello Jidapa

>>> myage = input()
25
>>> print(myage + 1)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: can only concatenate str
(not "int") to str
```



Note: Input จะถูกเก็บค่าใน  
รูปแบบ String ถึงแม้ว่ารูป  
ข้อความจะเป็นตัวเลข

หากต้องการนำตัวเลขจาก  
input ไปใช้ จะต้องเปลี่ยน  
ประเภทของข้อมูลหรือ  
**Data type conversion**  
(จาก str เป็น int หรือ  
float) ก่อนการ  
ดำเนินการอื่น ๆ



- `int()` เป็นคำสั่งการเปลี่ยนประเภทข้อมูลจากค่าเดิมเป็น `int`
- `float()` เป็นคำสั่งการเปลี่ยนประเภทข้อมูลจากค่าเดิมเป็น `float`

```
>>> myage = input()
25
>>> print(int(myage) + 1)
26
>>> print(float(myage) + 1)
26.0

>>> '1234'+'5'
'12345'
>>> int('1234') + float('5')
1239.0
```

`int()` และ `float()` จะสามารถ  
เปลี่ยนประเภทค่าข้อมูลได้ ก็ต่อเมื่อ  
String นั้น ๆ อยู่ในรูปแบบตัวเลข

**\*\* สามารถลองพิมพ์ `int('H')`  
เพื่อดูผลลัพธ์ได้**



- `str()` เป็นคำสั่งการเปลี่ยนประเภทข้อมูลจากค่าเดิมเป็น String

```
>>> myage = input()
25
>>> print(int(myage) + 1)
26
>>> print('Your age next year is ' + str(int(myage)+1))
Your age next year is 26

>>> str(100.5)
'100.5'
>>> print(str(100.5) + '124')
100.5124
```





Mahidol University  
*Wisdom of the Land*



# Write a Simple Python Program



1 **Analyze** การวิเคราะห์ปัญหาและวิธีการแก้ไขปัญหา

**Tool:** Pseudocode or Flowchart

2 **Code** เขียนชุดคำสั่งในภาษาโปรแกรม

Python Basics :)

3 **Run** รันชุดโปรแกรมเพื่อดูผลลัพธ์

4 **Test** การทดสอบโปรแกรม



จงเขียนโปรแกรมเพื่อคำนวณพื้นที่สี่เหลี่ยม ขนาดกว้าง  $W=5$  และ สูง  $H=20$

## 1 Pseudocode

1. รับค่า  $h$  (ความสูง) และ  $w$  (ความกว้าง)
2. หาพื้นที่สี่เหลี่ยม โดย  $area = h \times w$
3. แสดงผลลัพธ์  $area$

Expected output:

## 2 Python Code: `calRectArea.py`

1. `# รับค่า h (ความสูง) และ w (ความกว้าง)`
2. `w=5`
3. `h=20`
4. `# หาพื้นที่สี่เหลี่ยม โดย area = h x w`
5. `area=h*w`
6. `#แสดงผลลัพธ์ area`
7. `print(area)`



จงเขียนโปรแกรมเพื่อคำนวณพื้นที่สี่เหลี่ยม ขนาดกว้าง  $W=5$  และ สูง  $H=20$

### 3 Run the Program (python calRectArea.py)

```
1 # Define w=5, h=20
2 w=5
3 h=20
4 # Calculate area: h x w
5 area=h*w
6 # Display the area
7 print(area)
```

```
PS C:\Users\jik41\Desktop\ITDS120> python calRectArea.py
100
PS C:\Users\jik41\Desktop\ITDS120>
```

### 4 Test the Program

Expected output: 100  
Program output: 100



จงเขียนโปรแกรมเพื่อคำนวณพื้นที่สี่เหลี่ยม โดยรับ Input เป็นความกว้าง  $w$  และความสูง  $h$

### 1 Pseudocode

1. Input  $w, h$
2. Calculate area:  $h \times w$
3. Display the area

### 2 Python Code: `calRectArea2.py`

```
1. # Input w and h
2. w=input()
3. h=input()
4. # Calculate area: h x w
5. area=float(h) * float(w)
6. # Display the area
7. print(area)
```



จงเขียนโปรแกรมเพื่อคำนวณพื้นที่สี่เหลี่ยม โดยรับ Input เป็นความกว้าง  $w$  และความสูง  $h$

### 3 Run the Program (python calRectArea2.py)

```
calRectArea2.py > ...
1 # Input w and h
2 w=input()
3 h=input()
4 # Calculate area: h x w
5 area=float(h)* float(w)
6 # Display the area
7 print(area)
```

```
PS C:\Users\jik41\Desktop\ITDS120> python calRectArea2.py
10.5
2
21.0
PS C:\Users\jik41\Desktop\ITDS120> 
```

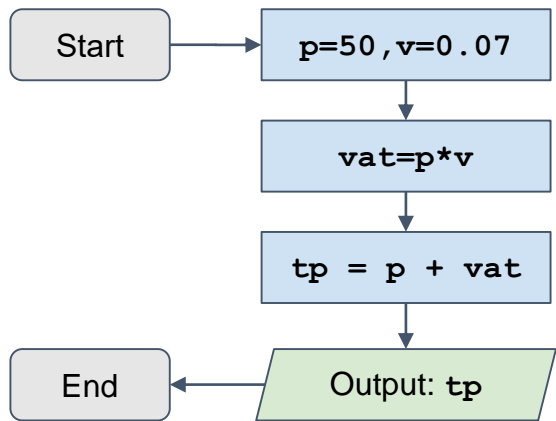
### 4 Test the Program

- Create your own test case to check the algorithm



## จงเขียนโปรแกรมเพื่อคำนวณราคาสินค้าหลัง VAT โดยราคาสินค้า 50 บาทและ VAT 7%

### 1 Flowchart



Expected output:

### 2 Python Code: `calTotalPrice.py`

```
1. # Define p=40 , v=0.07
2. p=50
3. v=0.07
4. # Calculate: vat=p*v
5. vat=p*v
6. # Calculate: tp=p+vat
7. tp=p+vat
8. # Display tp
9. print(tp)
```



จงเขียนโปรแกรมเพื่อคำนวณราคาสินค้าหลัง VAT โดยราคาสินค้า 50 บาทและ VAT 7%

### 3 Run the Program (`python calTotalPrice.py`)

```
1 # Define p=40 , v=0.07
2 p=50
3 v=0.07
4 # Calculate: vat=p*v
5 vat=p*v
6 # Calculate: tp=p+vat
7 tp=p+vat
8 # Display tp
9 print(tp)
```

```
PS C:\Users\jik41\Desktop\ITDS120> python calTotalPrice.py
53.5
PS C:\Users\jik41\Desktop\ITDS120>
```

### 4 Test the Program

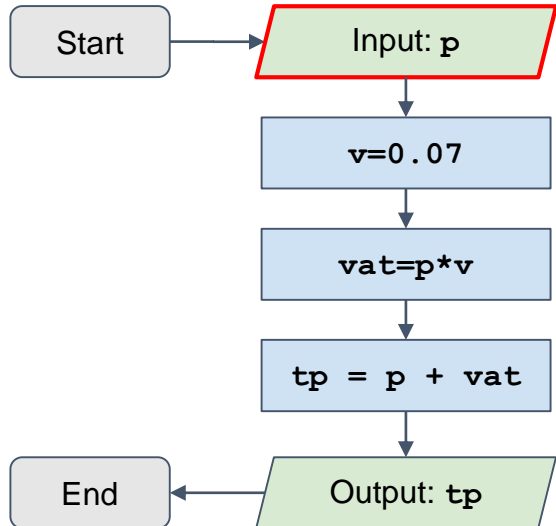
Expected output: 53.5  
Program output: 53.5





## จงเขียนโปรแกรมเพื่อคำนวณราคาสินค้าหลัง VAT 7% โดยรับ Input เป็นราคาสินค้า

### 1 Flowchart



### 2 Python Code: calTotalPrice2.py

```
1. # Input p and convert to float
2. p = input()
3. p = float(p)
4. # Define v=0.07
5. v = 0.07
6. # Calculate: vat=p*v
7. vat = p*v
8. # Calculate: tp=p+vat
9. tp = p+vat
10. # Display tp
11. print(tp)
```



จงเขียนโปรแกรมเพื่อคำนวณราคาสินค้าหลัง VAT 7% โดยรับ  
Input เป็นราคาสินค้า

### 3 Run the Program (python calTotalPrice2.py)

```
calTotalPrice2.py
1 # Input p and convert to float
2 p=input()
3 p=float(p)
4 # Define v=0.07
5 v=0.07
6 # Calculate: vat=p*v
7 vat=p*v
8 # Calculate: tp=p+vat
9 tp=p+vat
10 # Display tp
11 print(tp)
```

```
PS C:\Users\jik41\Desktop\ITDS120> python calTotalPrice2.py
100
107.0
PS C:\Users\jik41\Desktop\ITDS120>
```

### 4 Test the Program

Create your own test case to check the algorithm



- การตั้งชื่อตัวแปรควรมีความเชื่อมโยงกับจุดประสงค์ในการนำไปใช้

**Example:** width  $\rightarrow$  w, height  $\rightarrow$  h, VAT  $\rightarrow$  vat,  
student name  $\rightarrow$  stu\_name

Area of circle  $\rightarrow$  areaOfCircle

- สามารถใช้ \_ หรือ camelCase ในการตั้งชื่อตัวแปร
- ควรใช้ comments ในการอธิบายคำสั่งให้เหมาะสม เช่น ทุก ๆ ขั้นตอนที่สำคัญ

```
# Input p and convert to float
p=input()
p=float(p)
```



Mahidol University  
*Wisdom of the Land*



KEEP  
CALM  
AND  
CODE  
ON

keep-calm.net

Thanks to: <https://keep-calm.net/keep-calm-and-code-on.html>

# Happy Coding :)