



GBDi

Government Big Data Institute

สถาบันส่งเสริมการวิเคราะห์และบริหารข้อมูลขนาดใหญ่ภาครัฐ (สวช.)



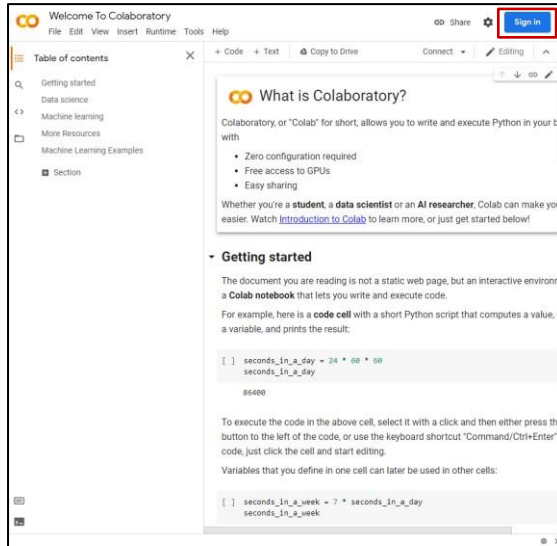
Python Basics

May 20, 2022

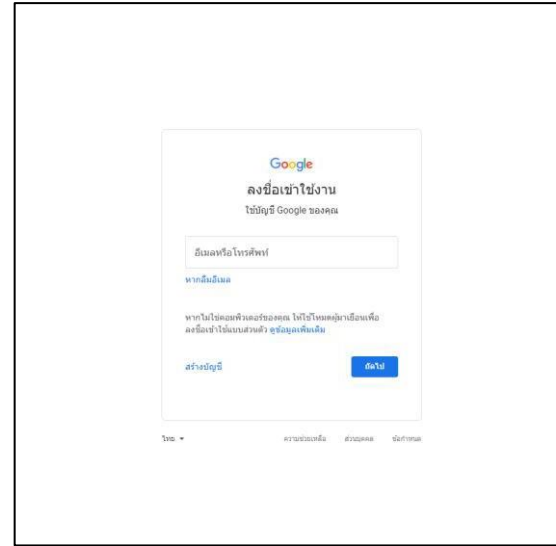
Chainarong Tumapha

Google Colab Setup (10 min.)

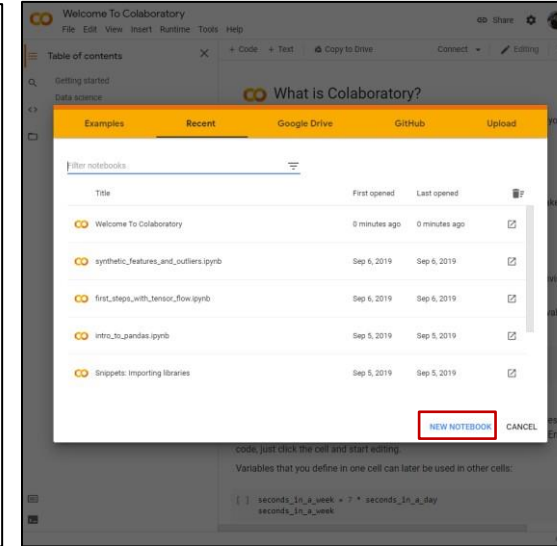
<https://colab.research.google.com>



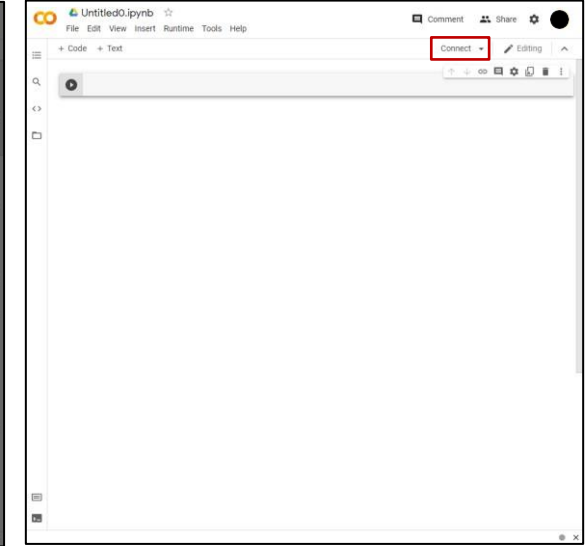
1. กด Sign in



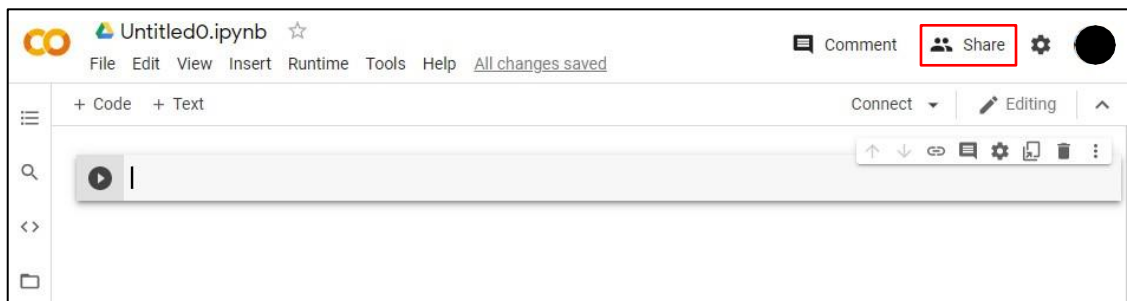
2. กรอกข้อมูล Gmail



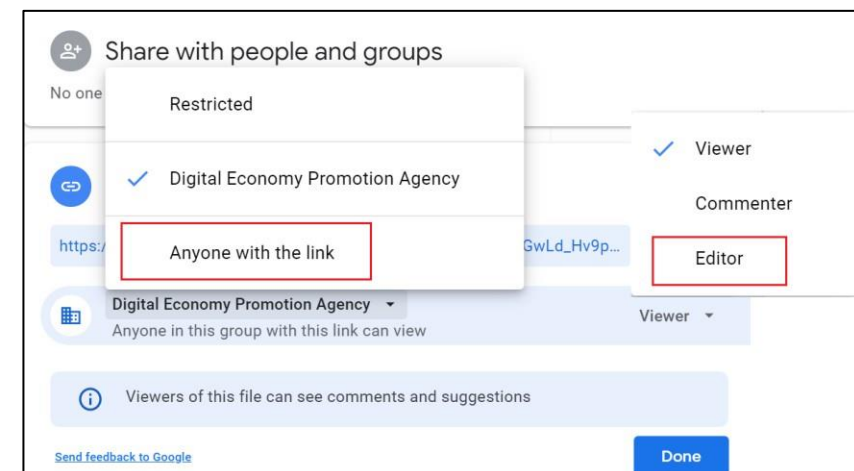
3. กด NEW NOTEBOOK



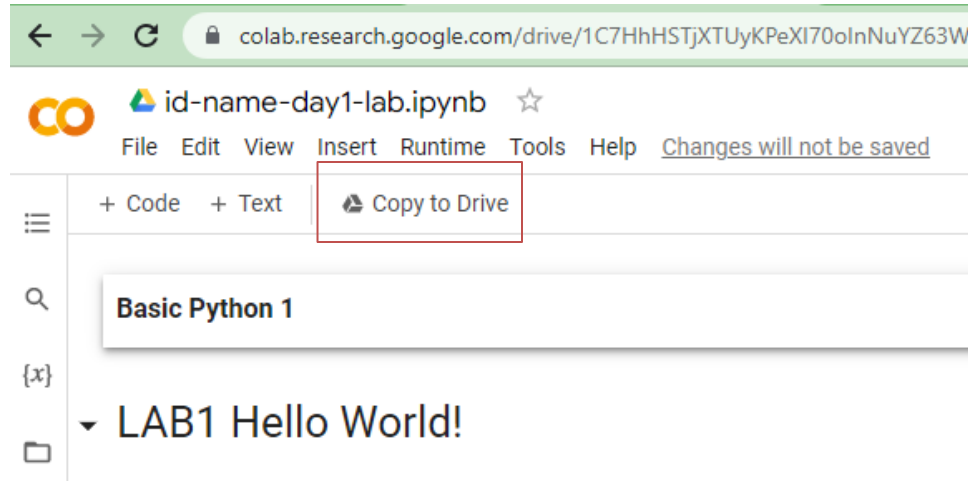
4. ตรวจสอบสถานะ



5. กด Share และปรับการเข้าถึงให้ทุกคนที่มี link สามารถแก้ไขได้




Google Colab Setup (10 min.)




6. Clone Colab โดยเข้าไปที่ <https://colab.research.google.com/drive/1C7HhHSTjXTUyKPeXI70oInNuYZ63WA8m?usp=sharing>
กดปุ่ม Copy to Drive

7. แก่ชื่อไฟล์เป็น id และชื่อภาษาอังกฤษของเรา
เช่น 001-Chainarong-day1-lab.ipynb
จากนั้นย้อนกลับไปทำข้อ 5 ตั้งค่าแชร์อีกรอบ

 Copy of id-name-day1-lab.ipynb ☆
File Edit View Insert Runtime Tools Help Last saved at 3:58 AM



 001-Chainarong-day1-lab.ipynb ☆
File Edit View Insert Runtime Tools Help All changes saved

Agreements

1. ท้ายคาบจะให้ส่งลิงค์ colab ผ่าน Form โดยจะต้องมีการทำ lab ในทุก lab (ไม่นับ extra)
2. code เฉลยจะถูกส่งให้หลังจากเลยกำหนดส่ง colab แล้ว
3. สุ่มแจกของรางวัลกับผู้ส่ง colab ในข้อแรกตามเวลาที่กำหนด
4. กรณีที่มีปัญหาให้ทักถาม TA หรือถามในแชทได้เลย

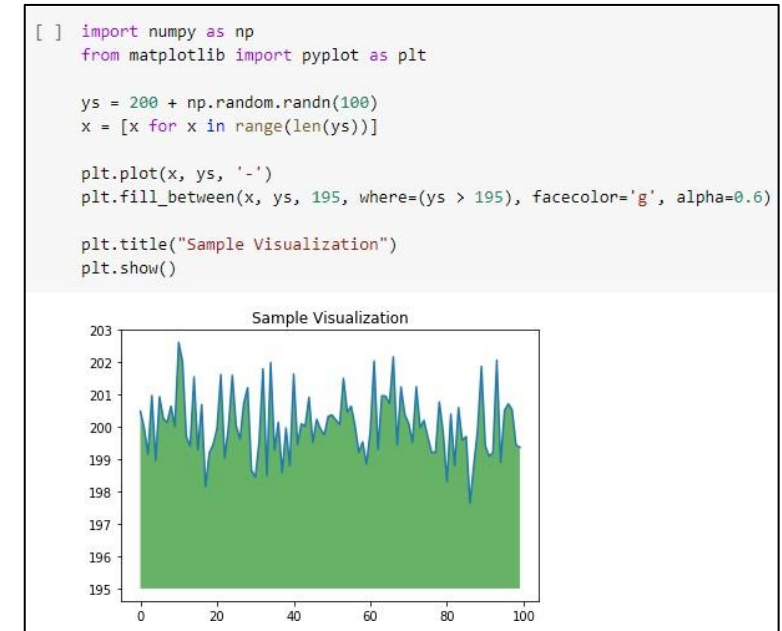
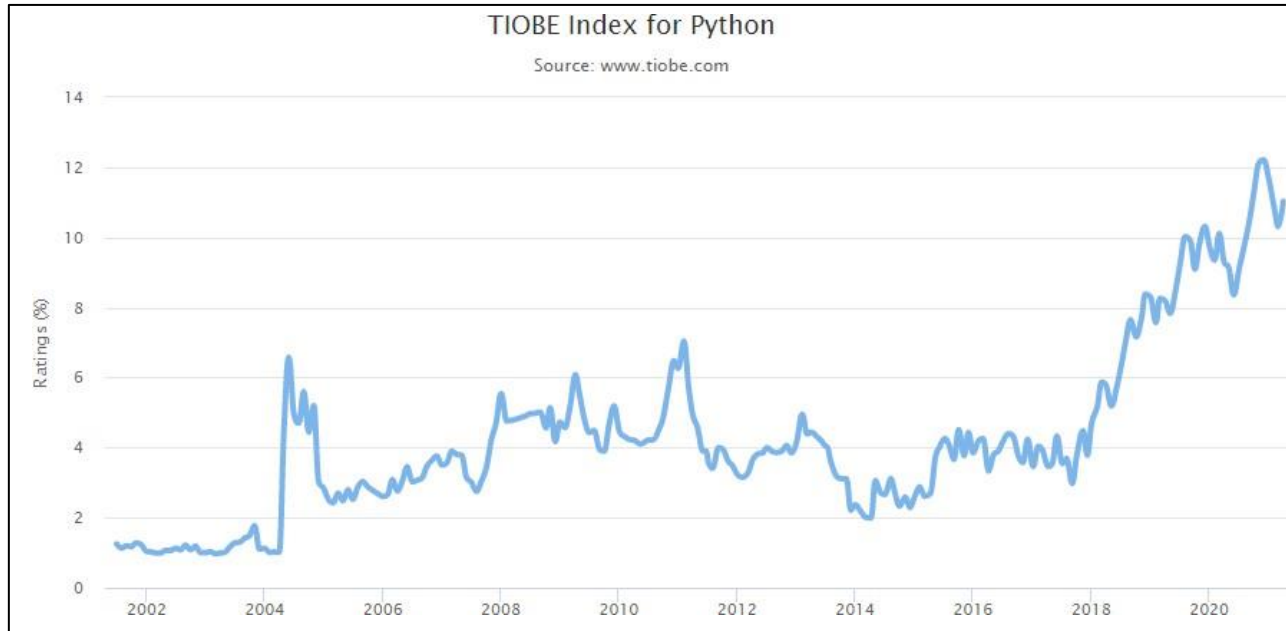


Motivation

ค่ามากที่สุด ค่าน้อยที่สุด ค่าเฉลี่ย และส่วนเบี่ยงเบนมาตรฐานของตัวเลขต่อไปนี้มีค่าเป็นเท่าใด

54, 71, 60, 54, 42, 64, 43, 89, 96, 38, 79, 52, 56, 92, 7, 8, 2,
83, 77, 87, 97, 79, 46, 78, 11, 63, 14, 94, 52, 41, 26, 77, 45, 56,
1, 61, 61, 61, 94, 68, 35, 43, 69, 6, 66, 67, 21, 12, 31, 36, 57,
43, 98, 10, 20, 16, 65, 25, 46, 24, 15, 11, 65, 13, 19, 36, 82, 9,
83, 9, 97, 46, 97, 60, 73, 3, 28, 12, 29, 11, 31, 41, 6, 69, 56,
26, 52, 9, 57, 92, 31, 66, 13, 71, 28, 18, 58, 2, 82, 0, 67, 27

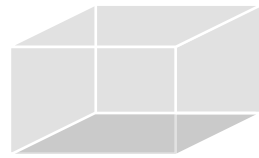
Why Python?



Google Colab

Reference: <https://www.tiobe.com/tiobe-index>
<https://colab.research.google.com>

Fundamental programming



Variables

เก็บข้อมูล



Function

เรียกใช้ชุดคำสั่ง



Flow control

ตัดสินใจ



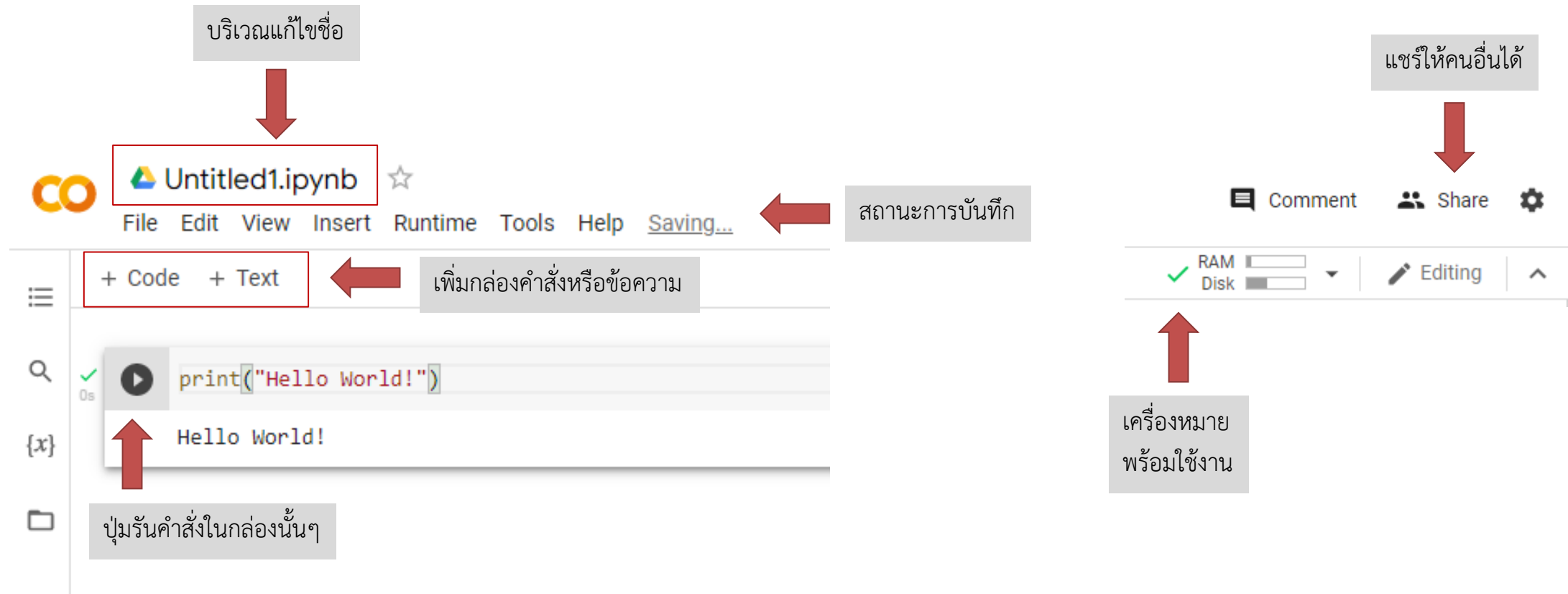
Loop

ทำซ้ำ

LAB1 : Hello World! (5 Min.)

พิมพ์คำสั่ง `print("Hello World!")` ลงใน Google Colab ตั้งชื่อไฟล์ว่า EZPython1.ipynb

Extra : ทำให้ output แสดง double quote ออกมาด้วย



The screenshot shows the Google Colab interface with several annotations in Thai:

- บริเวณแก้ไขชื่อ** (Rename area): Points to the filename `Untitled1.ipynb`.
- สถานะการบันทึก** (Save status): Points to the `Saving...` status in the top menu.
- เพิ่มกล่องคำสั่งหรือข้อความ** (Add code or text box): Points to the `+ Code` and `+ Text` buttons.
- ปุ่มรันคำสั่งในกล่องนั้นๆ** (Run command button in that box): Points to the play button icon next to the code cell.
- แชร์ให้คนอื่นได้** (Share with others): Points to the `Share` button in the top right.
- เครื่องหมายพร้อมใช้งาน** (Ready icon): Points to the green checkmark icon next to the RAM/Disk status.

The code cell contains the following code:

```
print("Hello World!")
```

The output of the code cell is:

```
Hello World!
```


Comments

Python will not run comments

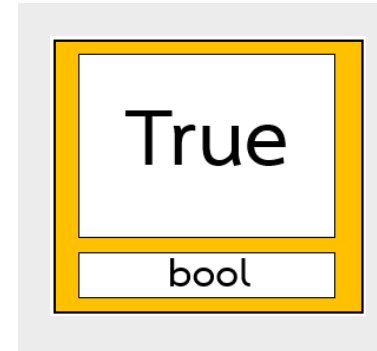
#ส่วนี่เขียวจะไม่ถูกนำไปประมวลผล
`print("Hello World!")`
`print(1) #This is a comment`

""""
ภายในนี้
ก็จะไม่ถูก
ประมวลผล
""""
`print([2])`

Basic Data Types



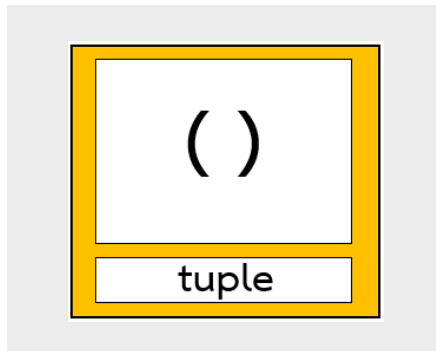
Numbers



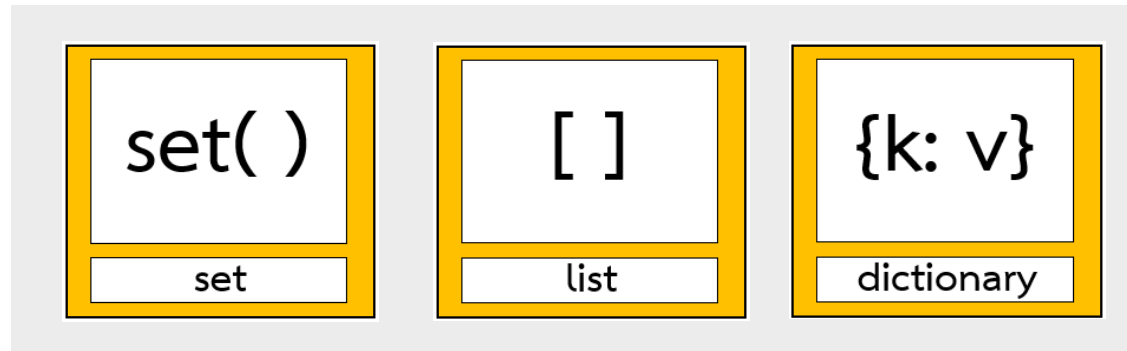
Logical values



Texts

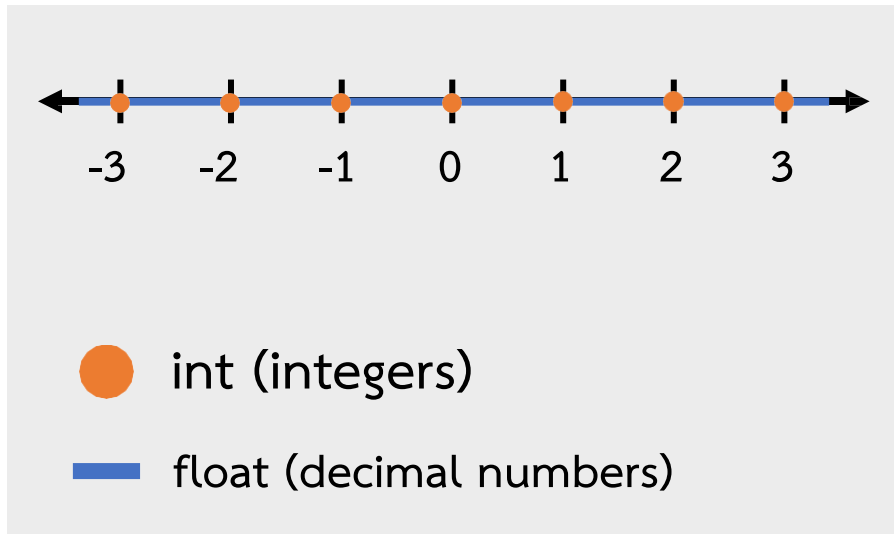


Immutable Containers
(เปลี่ยนค่าข้างในไม่ได้)



Mutable Containers
(เปลี่ยนค่าข้างในได้)

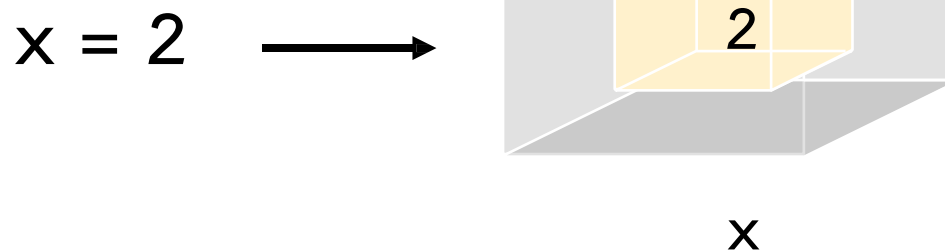
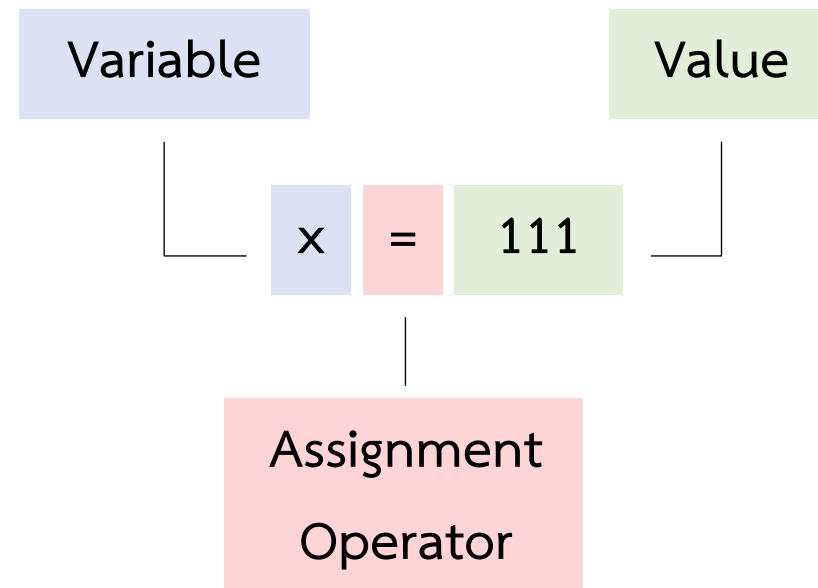
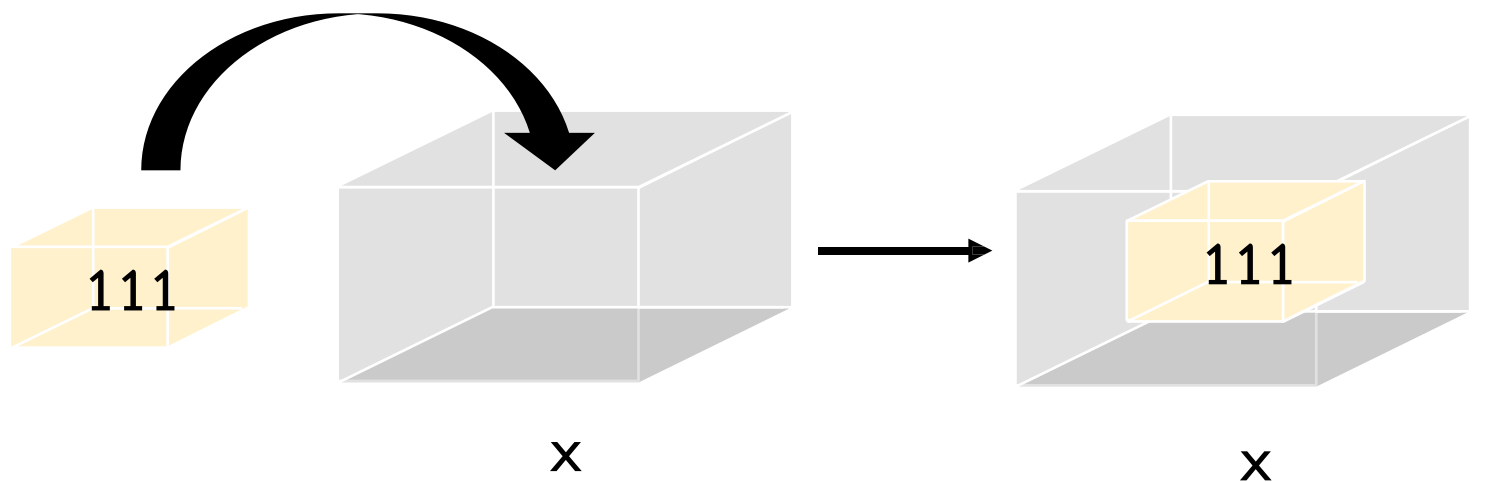
Numbers: int and float



Operation	Operator	Example
Addition	+	$11 + 2 = 23$
Subtraction	-	$11 - 2 = 9$
Multiplication	*	$11 * 2 = 22$
Division	/	$11 / 2 = 5.5$
Integer Division	//	$11 // 2 = 5$
Congruent Modulo	%	$11 \% 2 = 1$
Exponentiation	**	$11 ** 2 = 121$

There are more advanced operations, but they require an additional package, such as [math](#) and [numpy](#).

Variables



Operations with Assignment Operators

Operation	Operator	Example	Equivalent form
Addition	<code>+=</code>	<code>x += 2</code>	<code>x = x + 2</code>
Subtraction	<code>-=</code>	<code>x -= 2</code>	<code>x = x - 2</code>
Multiplication	<code>*=</code>	<code>x *= 2</code>	<code>x = x * 2</code>
Division	<code>/=</code>	<code>x /= 2</code>	<code>x = x / 2</code>
Integer Division	<code>//=</code>	<code>x //= 2</code>	<code>x = x // 2</code>
Congruent Modulo	<code>%=</code>	<code>x %= 2</code>	<code>x = x % 2</code>
Exponentiation	<code>**=</code>	<code>x **= 2</code>	<code>x = x ** 2</code>

Note that the operations in the table can operate to not only 2, but also all other numbers.

LAB2: Expressions (5 min.)

1. ค่า x มีค่าเท่าใดจากการคำนวณต่อไปนี้

$x = -73$

$x += 15$

2. ค่า $my_variable$ มีค่าเท่าใดจากการคำนวณต่อไปนี้

$my_variable = 157$

$my_variable \% = 10$

3. ค่า y จากการคำนวณต่อไปนี้ มีค่าเท่าใด

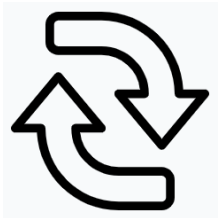
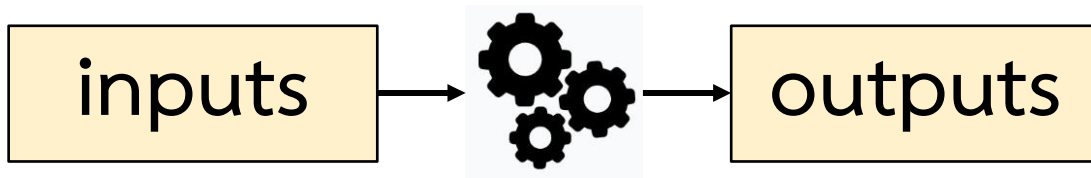
$y = 28 / 13$

4. กำหนดให้ $y = x + 9$ (ใช้ค่า x จากข้อ 1) จงหาค่าของ y หลังจากการคำนวณต่อไปนี้

$y /= 10$

ส่งคำตอบมาในช่องแชทในรูปแบบ
[คำตอบ1,คำตอบ2,คำตอบ3,คำตอบ4]

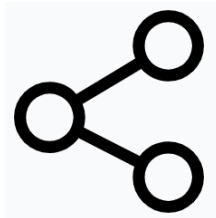
Functions



Reuse



Shorten



Share

Please visit [this link](#) for more information.

Function declaration

Parameters

```
def function_name(input1, input2, ...):  
    '''  
    docstring: description of the function  
    '''  
    statement1  
    statement2  
    ...  
    return output1, output2, ...
```

Function called

Arguments

```
x = function_name(input1, input2, ...)
```

Functions

```
def show_student(name, organization="No organization"):
    print("Name:",name)
    print("Organization:",organization)
    print('\n') #Enter
```

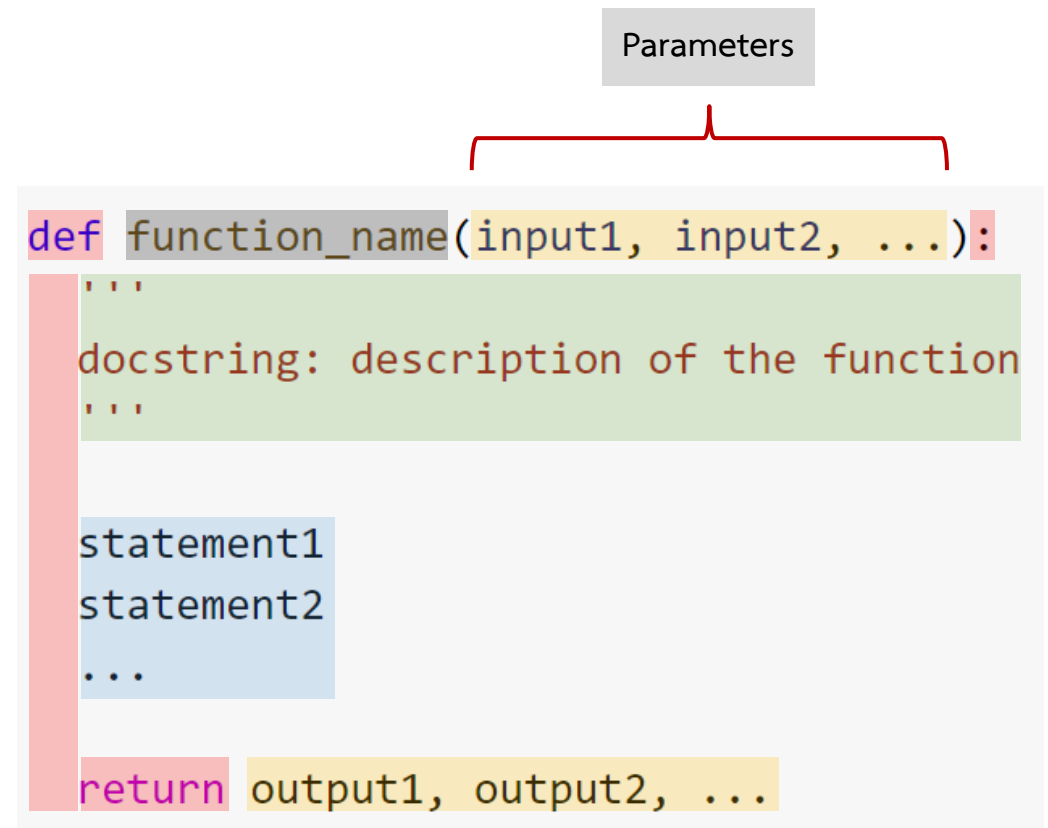
```
show_student("Mr.A")
```

```
show_student("Mr.A","GBDi")
```

```
show_student(name="Mr.A")
```

```
show_student(organization="GBDi", name="Mr.A")
```

* Parameter ที่ไม่มีค่า default ต้องมาก่อน Parameter ที่มีค่า default เสมอ



```
def function_name(input1, input2, ...):
    '''
    docstring: description of the function
    '''

    statement1
    statement2
    ...

    return output1, output2, ...
```


Functions

What if we use functions
from other source ?

example

The `input()` function takes input from the user and returns it.

Example

```
name = input("Enter your name: ")
print(name)

# Output:
# Enter your name: James
# James
```

Run Code >>

input() Syntax

The syntax of `input()` function is:

```
input([prompt])
```

inputs

input() Parameters

The `input()` function takes a single optional argument:

- **prompt (Optional)** - a string that is written to standard output (usually screen) without trailing newline

outputs

input() Return Value

The `input()` function reads a line from the input (usually from the user), converts the line into a string by removing the trailing newline, and returns it.

If EOF is read, it raises an `EOFError` exception.

LAB3: Student information (10 min.)

1. ให้เพิ่ม parameter ที่ชื่อ score เก็บคะแนนที่เป็นจำนวนเต็มใน show_student แล้วแสดงคะแนนหลัง organization

```
show_student(organization="GBDi", name="Mr.A",score=33)  
show_student(organization="GBDi", name="Mr.B",score=40)
```

```
Name: Mr.A  
Organization: GBDi  
Score: 33
```

```
Name: Mr.B  
Organization: GBDi  
Score: 40
```

2. เปลี่ยนจากการกำหนดค่า score ใน function show_student เป็นรับค่าจากคีย์บอร์ดแทน

Extra: แสดง % คะแนนที่ทำได้จากคะแนนเต็ม 88 คะแนน

Enter score:

Type checking and Casting

Type checking

```
print("1+1:", '1'+'1')  
print(type('1'))  
print("1+1:", 1+1)  
print(type(1))
```

```
1+1: 11  
<class 'str'>  
1+1: 2  
<class 'int'>
```

Type casting

```
x = int('5')  
print('x is', type(x), 'and value is', x)
```

```
x = float(5)  
print('x is', type(x), 'and value is', x)
```

```
x = str(5.0)  
print('x is', type(x), 'and value is', x)
```

```
x is <class 'int'> and value is 5  
x is <class 'float'> and value is 5.0  
x is <class 'str'> and value is 5.0
```

LAB4: Type casting (10 min.)

3. แสดง % คะแนนที่ได้จากคะแนนเต็ม 88 คะแนน

```
show_student(organization="GBDi", name="Mr.A", score=score)
```

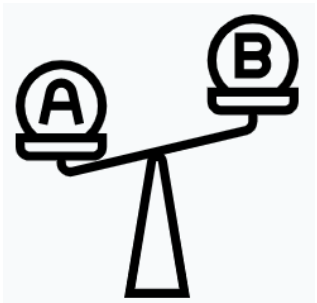
```
Enter score: 44  
Name: Mr.A  
Organization: GBDi  
Score: 44  
Score (%): 50.0
```

Extra: แสดง % ในรูปของทศนิยม 1 ตำแหน่ง

Logical Values: bool (Boolean)



Can be either
True or False



Obtained from
comparison

Operation	Operator	Example
is greater than	<code>></code>	<code>x > y</code>
is less than	<code><</code>	<code>x < y</code>
is greater than or equal to	<code>>=</code>	<code>x >= y</code>
is less than or equal to	<code><=</code>	<code>x <= y</code>
is equal to	<code>==</code>	<code>x == y</code>
is different from	<code>!=</code>	<code>x != y</code>

The equals sign here does not refer to the assignment operator.

Logical Operators

Expression	Result
True and True	True
True and False	False
False and True	False
False and False	False

Conjunction
(and, &)

“True เมื่อทั้งหมด True”

Expression	Result
True or True	True
True or False	True
False or True	True
False or False	False

Disjunction
(or, |)

“False เมื่อทั้งหมด False”

Expression	Result
Not True	False
Not False	True

Negation
(not, ~)

LAB5 : Boolean Values (5 Min.)

กำหนดให้ $x = 82$ จงหาค่า Boolean ของ expression ต่อไปนี้

1. $-x > 15$

2. $x \neq 28$

3. $x \leq 100$

4. $x == 3$

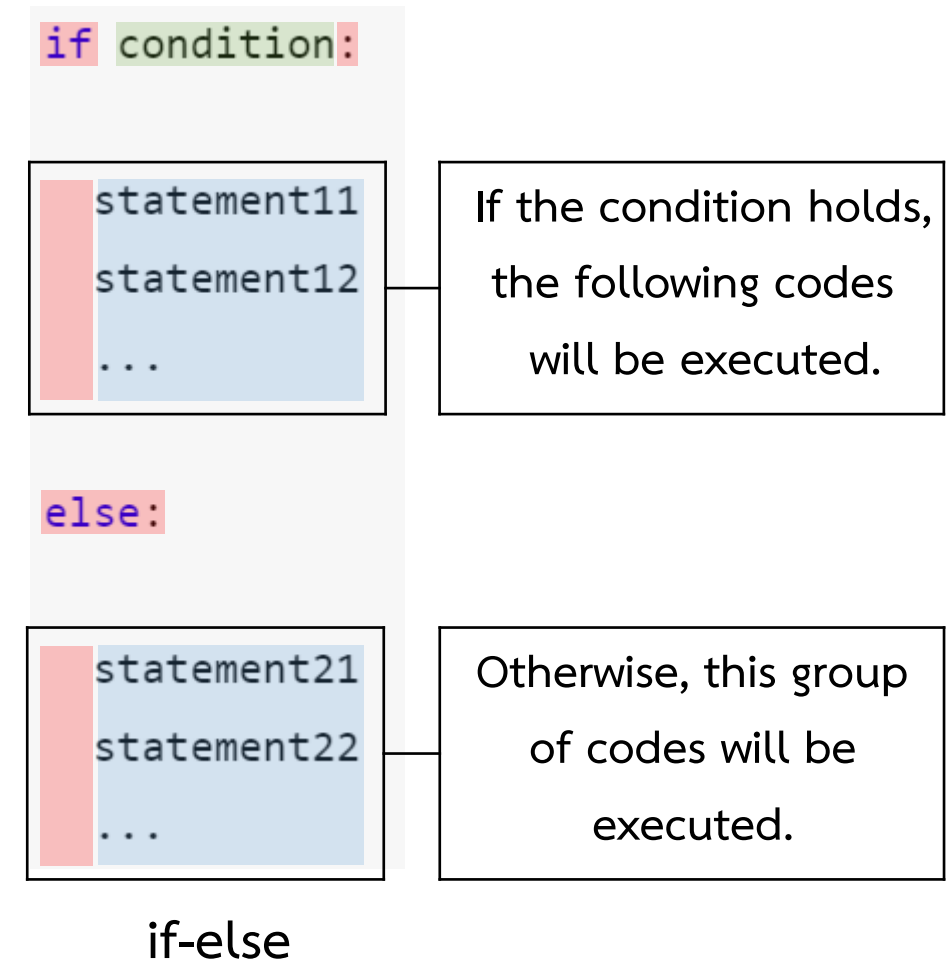
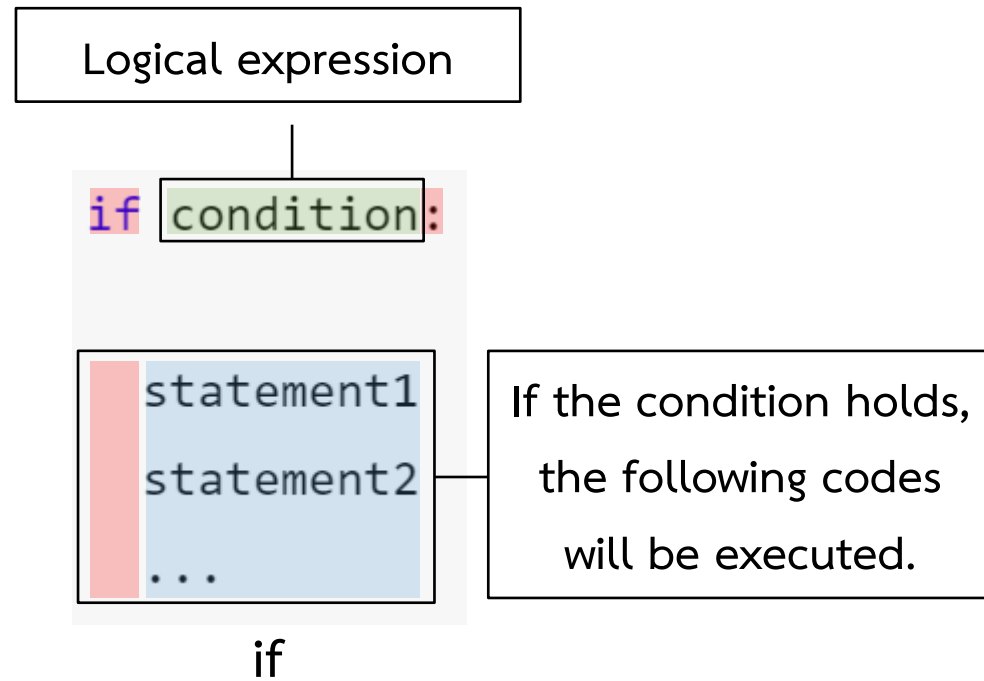
5. $(x < 0) \text{ and } (x == -17)$

6. $\text{not } (x + 10 < 36) \text{ or } (x \geq 82)$

7. $\text{not } ((x + 10 < 36) \text{ or } (x \geq 82))$

8. $(x > 4) \text{ and } (x < 5) \text{ or } (x \neq -6)$

If and If-Else Statements



If-Elif-Else Statements

```
if condition1:
```

```
statement11  
statement12  
...
```

```
elif condition2:
```

```
statement21  
statement22  
...
```

```
else:
```

```
statementN1  
statementN2  
...
```

If the condition1 does not hold but the condition2 is satisfied, execute these codes.

if-elif-else

```
x = 10  
if x>10:  
    print("x is more than 10")  
elif x<10:  
    print("x is less than 10")  
else:  
    print("x is equal to 10")
```

x is equal to 10

LAB6 : Grading (10 min.)

พัฒนาฟังก์ชัน `show_student` ให้ตัดเกรดตาม percent ที่ได้โดย
ถ้าเกิน 80% ให้ได้ A, 60-80% ได้ B และต่ำกว่า 60 ได้ C

Extra: เพิ่มอีก 1 เงื่อนไข ถ้าต่ำกว่า 40 ได้ D

```
show_student(organization="GBDi", name="Mr.A", score=score)
```

```
Enter score: 80  
Name: Mr.A  
Organization: GBDi  
Score: 80  
Score (%): 90.9  
Grade: A
```

Texts: str

`x = 'Hello World!'`

or

`x = "Hello World!"`

value	H	e	l	l	o		W	o	r	l	d	!
index	0	1	2	3	4	5	6	7	8	9	10	11
	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

`x[0]`

`x[5]`

`x[9]`

`x[-12]`

`x[-7]`

`x[-3]`

Texts: str

`x = 'Hello World!'`

or

`x = "Hello World!"`

value	H	e	l	l	o		W	o	r	l	d	!
index	0	1	2	3	4	5	6	7	8	9	10	11
	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

`x[0]` `'H'`

`x[5]` `' '`

`x[9]` `'l'`

`x[-12]` `'H'`

`x[-7]` `' '`

`x[-3]` `'l'`

Basic Operations

Concatenation

‘สวัสดี’ + ‘จ้า’ → ‘สวัสดีจ้า’

‘Thank you’ * 3 → ‘Thank youThank youThank you’

Length

len(‘Hello World’) → 11

Membership

‘H’ in ‘Hello World’ → True

‘H’ not in ‘Hello World’ → False

String Slicing

x = 'Hello World!'

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

x [start : stop : step]

1. x[1: 9]

3. x[2: 12: 1]

2. x[0: 11: 2]

4. x[3: -2: 2]

LAB7: String Slicing

x = 'Hello World!'

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

x [start : stop : step]

x[1: 9]

'ello Wor'

x[2: 12: 1]

'llo World!'

x[0: 11: 2]

'HloWr d'

x[3: -2: 2]

'l ol'

Basic String Methods

กำหนดให้ $x = \text{'28-04-2 021'}$,

			2	8	-	0	4	-	2		0	2	1		
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Operation	Syntax	Output	Description
Find	<code>x.find('-')</code>	5	Return the first index found. Otherwise, return -1.
Replace	<code>x.replace('0', '??')</code>	<code>'28-??4-2 ??21'</code>	Replace the first string with the second one.
Strip	<code>x.strip()</code>	<code>'28-04-2 021'</code>	Remove whitespaces of both two ends.

LAB8: String Combos (20 min.)

1. ให้ `x = 'Python is fun!'` จงทำให้เป็น `'!nuf si nohtyP'` โดยการใช้ string slicing

2. สร้างฟังก์ชัน `remove_title` ในการตัดคำนำหน้าชื่อ โดยกำหนดให้ input เป็น text ที่อาจมีเว้นวรรคและบาง input อาจจะไม่มีการนำหน้า

คำนำหน้าที่ต้องตัดได้แก่ นาย นาง นางสาว Mr. Mrs. Miss

```
remove_title('นาง สมศรี นายนารัตน์')
```

```
'สมศรี นายนารัตน์'
```

```
remove_title('นายนายะ มาโดยจุ')
```

```
'นายะ มาโดยจุ'
```

Lunch Time

Containers: list

‘Hello World!’

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

Why just strings?

-25	‘What’	1.1	‘!’	True
0	1	2	3	4



Operation	Operator	Example
Concatenation	+	$x + y$
	*	$x * 10$
Length	<code>len()</code>	<code>len(x)</code>
Membership	<code>in</code>	<code>5 in x</code>
	<code>not in</code>	<code>5 not in x</code>

`x[start : stop : step]`

```
x + [-3.14, False, 'list']
```

```
[-25, 'What', 1.1, '!', True, -3.14, False, 'list']
```

```
len(x)
```

```
5
```

```
5 in x
```

```
False
```

```
5 not in x
```

```
True
```

```
x[1]
```

```
‘What’
```

```
x[1][2]
```

```
‘a’
```

```
x[3:]
```

```
['!', True]
```

```
x[0] = 28
```

```
x
```

```
[28, 'What', 1.1, '!', True]
```

List Methods

กำหนดให้ `x = []`

Operation	Syntax	Output	Description
Append	<code>x.append(9)</code>	<code>[9]</code>	Add an element at the end of the list.
Insert	<code>x.insert(0, False)</code>	<code>[False, 9]</code>	Add an element at a specified index.
Remove	<code>x.remove(9)</code>	<code>[False]</code>	Remove an element.
Copy	<code>x.copy(), x[:]</code>	Another list	Copy a list.

Containers: tuple

List

-25	'What'	1.1	‘!’	True
0	1	2	3	4

Everybody! Freeze!

-25	'What'	1.1	‘!’	True
0	1	2	3	4



Operation	Operator	Example
Concatenation	+	x + y
	*	x * 10
Length	len()	len(x)
Membership	in	5 in x
	not in	5 not in x

x [start : stop : step]

```
x + (-3.14, False, 'list')  
  
(-25, 'What', 1.1, '!', True, -3.14, False, 'list')
```

```
x[0] = 28  
-----  
TypeError                                Traceback (most recent call last)  
<ipython-input-62-2b4c671115a9> in <module>()  
----> 1 x[0] = 28  
  
TypeError: 'tuple' object does not support item assignment
```

SEARCH STACK OVERFLOW

Tuple Methods

กำหนดให้ $x = (-25, \text{'What'}, 1.1, \text{'!'}, \text{True})$

-25	'What'	1.1	‘!’	True
0	1	2	3	4

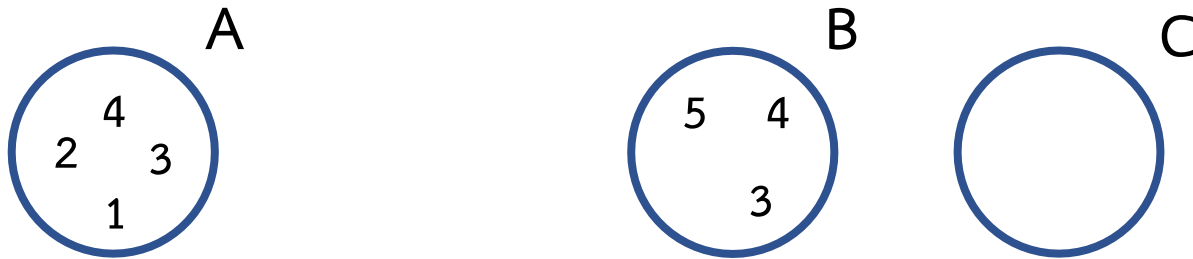
Operation	Syntax	Output	Description
Count	<code>x.count(1.1)</code>	1	Count frequency of an input
	<code>x.count('~')</code>	0	
Index	<code>x.index('!')</code>	3	Find the location of an input in a tuple
	<code>x.index(111)</code>	error	

LAB9 : list and tuple (5 min.)

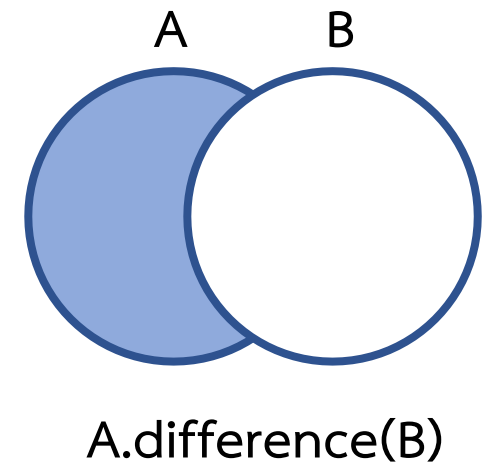
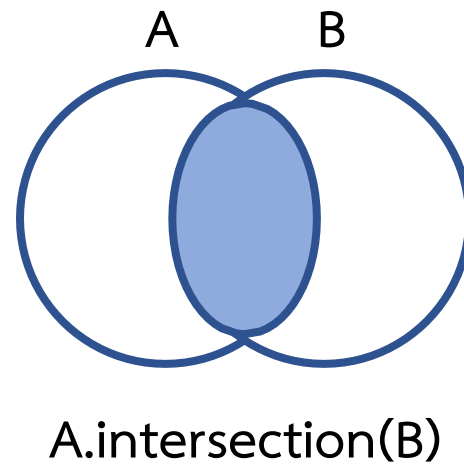
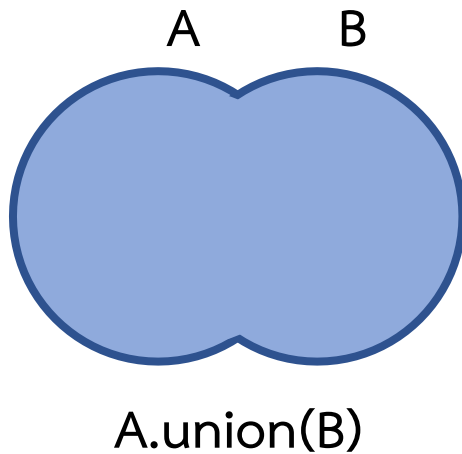
1. กำหนดให้ `x = ('Python', [1, False, 3], [True, [3.14]])`
หาขนาดและ data type ของ `x` และถ้าหากต้องการเข้าถึงค่า 3.14 จาก `x` ต้องใช้คำสั่งใด
2. หากต้องการเปลี่ยนค่า `False` ให้เป็น `True` สามารถทำได้หรือไม่ อย่างไร
3. หากต้องการเปลี่ยนค่า `'Python'` ให้เป็น `'Data'` สามารถทำได้หรือไม่ อย่างไร
4. กำหนดให้ `y = [1, 2, 4, 5, 6]` ให้เพิ่มค่า 3 เข้าไปใน `y` โดยให้อยู่ระหว่างค่า 2 และ 4

Containers: set

$A = \{ 1, 2, 3, 4, 4, 4 \}$ $B = \{ 3, 4, 5 \}$ $C = \text{set}()$



Operation	Operator	Example
Length	<code>len()</code>	<code>len(A)</code>
Membership	<code>in</code>	<code>5 in A</code>
	<code>not in</code>	<code>5 not in A</code>



Containers: dictionary

List

-25	'What'	1.1	'!'	True
0	1	2	3	4



Why need index?

-25	'What'	1.1	'!'	True
'1'	0	'How'	-3.14	4

key

value

'1'

0

False

-3.14

4



-25

'what'

1.1

'!'

True

```
x = {'1': -25, 0: 'What', 'How': 1.1, -3.14: '!', 4: True}
```

x['1']

-25

x[0]

'What'

x['How']

1.1

x[-3.14]

'!'

x[4]

True

x.keys()

```
dict_keys(['1', 0, 'How', -3.14, 4])
```

x.values()

```
dict_values([-25, 'What', 1.1, '!', True])
```

x.items()

```
dict_items([('1', -25), (0, 'What'), ('How', 1.1), (-3.14, '!'), (4, True)])
```

Dictionary methods

กำหนดให้ `x = { 'A': 'Ant', 'B': 'Bird' }`

Operation	Syntax	Output	Description
Delete	<code>x.pop('A')</code>	<code>'Ant'</code>	Delete a value in the specified index (key) and print the deleted value.
Copy	<code>x.copy()</code>	-	Copy a dictionary.
Update	<code>x.update({ 'C': 'Cat' })</code>	-	Add another dictionary to x.

รวมถึง `x.keys()`, `x.values()` และ `x.items()`

LAB10: dictionary (15 Min.)

สร้างฟังก์ชัน `convert_month` ในการแปลง
เดือนจากตัวเลขเป็นชื่อเดือน โดยกำหนดให้
input จะเป็นรูปแบบของ `dd-mm-yyyy`

```
convert_month('19-05-2564')
```

```
'19 พฤษภาคม 2564'
```

```
convert_month('09-09-2549')
```

```
'9 กันยายน 2549'
```

```
convert_month('31-12-2500')
```

```
'31 ธันวาคม 2500'
```

For-Loop

```
for var in container:
```

```
statement1
statement2
...
```

Count-controlled loop

```
for char in 'Python!':
    print(char)
```

P
y
t
h
o
n
!

String

```
for item in ('This', True, 3.14, -2):
    print(item)
```

This
True
3.14
-2

Tuple

```
for num in [1, 2, 3, 4, 5]:
    print(num)
```

range(1, 6)

1
2
3
4
5

List

```
for elm in {'A', 'B'}:
    print(elm)
```

B
A

Set

The exact syntax of the range function is range(start, stop, step).

For-Loop

```
for var in container:
```

```
statement1  
statement2  
...
```

Count-controlled loop

```
x = {'1':-25, 0:'What', 'How':1.1, -3.14:'!', 4:True}  
for key in x.keys():  
    print(key)
```

```
1  
0  
How  
-3.14  
4
```

```
for value in x.values():  
    print(value)
```

```
-25  
What  
1.1  
!  
True
```

```
for item in x.items():  
    print(item)
```

```
('1', -25)  
(0, 'What')  
('How', 1.1)  
(-3.14, '!')  
(4, True)
```

Dictionary

The exact syntax of the range function is range(start, stop, step).

LAB11: For-Loop (20 Min.)

1. หาผลรวมของจำนวนเต็มตั้งแต่ 1 ถึง 500

2. หาผลรวมของจำนวนคู่ตั้งแต่ 1 ถึง 500

3. สร้างฟังก์ชัน `mul_table` สำหรับแสดงตารางสูตรคูณ ยกตัวอย่างเช่น `mul_table(2)` จะแสดงสูตรคูณของแม่ 2

```
mul_table(2)
```

```
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
2 x 11 = 22
2 x 12 = 24
```

```
mul_table(25)
```

```
25 x 1 = 25
25 x 2 = 50
25 x 3 = 75
25 x 4 = 100
25 x 5 = 125
25 x 6 = 150
25 x 7 = 175
25 x 8 = 200
25 x 9 = 225
25 x 10 = 250
25 x 11 = 275
25 x 12 = 300
```

4. สร้างฟังก์ชัน `mul_tables` สำหรับแสดงตารางสูตรคูณหลายตาราง เช่น `mul_tables([1,3,5])`

```
mul_tables([3,5])
```

```
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30
3 x 11 = 33
3 x 12 = 36
```

```
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
5 x 11 = 55
5 x 12 = 60
```

Examples: While-Loop

```
while condition:  
    statement1  
    statement2  
    ...
```

Event-controlled loop

```
idx = 0  
text = 'Python!'  
while idx < len(text):  
    print(text[idx])  
    idx += 1
```

P
y
t
h
o
n
!

```
num = 1  
while num < 6:  
    print(num)  
    num += 1
```

1
2
3
4
5

LAB12: While-Loop (15 Min.)

รับค่าจำนวนเต็มบวกจากคีย์บอร์ดแล้วบวกไปเรื่อยๆ จนกว่าจะใส่ตัวอักษรที่ไม่ใช่ตัวเลข

```
Enter positive integer: 12
Enter positive integer: 5
Enter positive integer: -1
17
```

Example

Check if all the characters in the text are numeric:

```
txt = "565543"

x = txt.isnumeric()

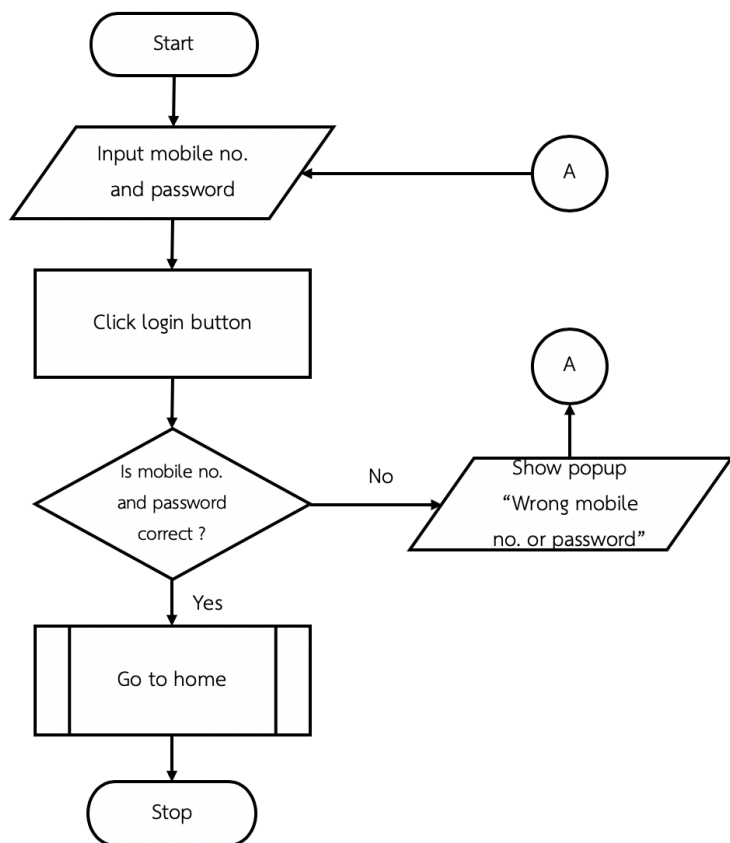
print(x)
```

Try it Yourself »

HINT: https://www.w3schools.com/python/ref_string_isnumeric.asp

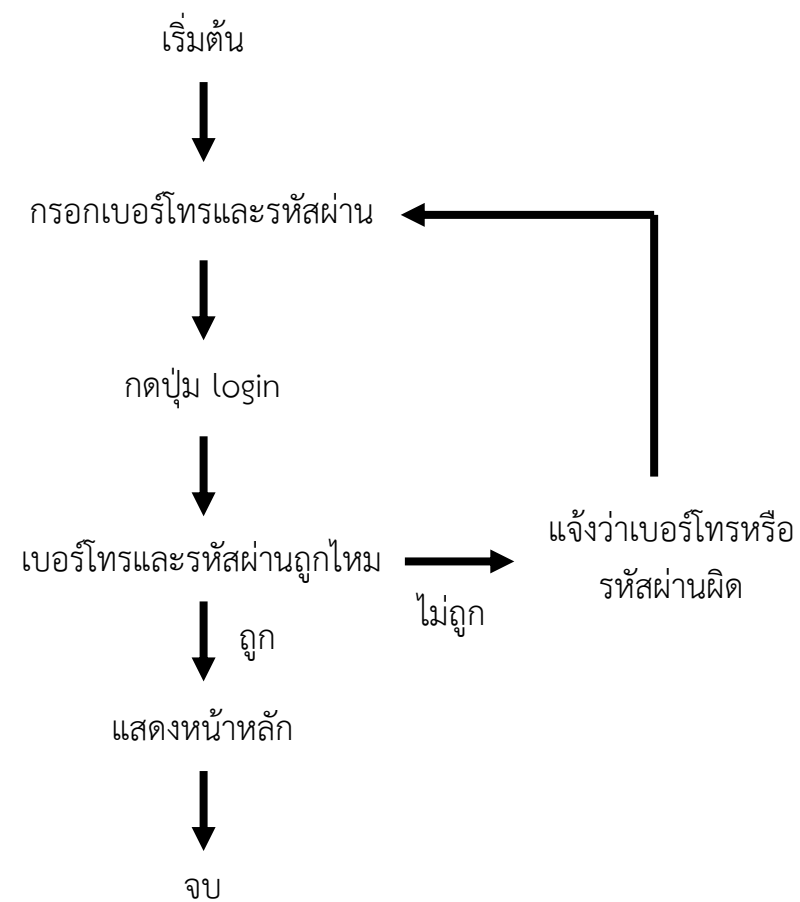
Programming design

ออกแบบฟังก์ชันการลงชื่อเข้าใช้งาน



<https://blog.nextzy.me/flowchart-101-f3ef25c3a92a>

เขียนให้ง่ายแต่
ไม่เป็นทางการ



LAB13 : num2word งานกลุ่ม (20 Min.)

ออกแบบฟังก์ชัน num2word ในการแปลงจำนวนเต็มที่ไม่เกิน 1 ล้านเป็นข้อความ
ตัวแทนกลุ่ม capture ภาพ/ขั้นตอนการทำงานของฟังก์ชัน โพสต์ในแชทห้องรวม

Extra: เขียน code ตามแบบที่คิดไว้

```
num2word(123451)
```

'หนึ่งแสนสองหมื่นสามพันสี่ร้อยห้าสิบเอ็ด'

```
num2word(2021)
```

'สองพันยี่สิบเอ็ด'

```
num2word(1000001)
```

กรุณาใส่จำนวนเต็มที่มีค่าไม่เกิน 1 ล้าน

```
num2word('Hello')
```

กรุณาใส่จำนวนเต็มที่มีค่าไม่เกิน 1 ล้าน

Bonus: Getting JSON data from APIs

JSON (JavaScript Object Notation) – A famous data-interchange format. It looks like dictionary in python.

```
{  
  'body': 'quia et suscipit sunt rem eveniet architecto',  
  'id': 1,  
  'title': 'sunt aut facere repellat provident',  
  'userId': 1  
}
```

JSON

```
[{  
  'body': 'quia et suscipit sunt rem eveniet architecto',  
  'id': 1,  
  'title': 'sunt aut facere repellat provident',  
  'userId': 1  
}]
```

List of JSON

```
import requests  
r = requests.get('https://jsonplaceholder.typicode.com/posts') #JSON API URL  
data = r.json()
```

Bonus : Web data (10 Min.)

นำ data ที่ได้มาหาจำนวน post ของแต่ละ user

```
import requests  
r = requests.get('https://jsonplaceholder.typicode.com/posts') #JSON API URL  
data = r.json()
```

```
userId: 1 has 10 post(s).  
userId: 2 has 10 post(s).  
userId: 3 has 10 post(s).  
userId: 4 has 10 post(s).  
userId: 5 has 10 post(s).  
userId: 6 has 10 post(s).  
userId: 7 has 10 post(s).  
userId: 8 has 10 post(s).  
userId: 9 has 10 post(s).  
userId: 10 has 10 post(s).
```



GBDi

Government Big Data Institute

สถาบันส่งเสริมการวิเคราะห์และบริหารข้อมูลขนาดใหญ่ภาครัฐ (สวช.)

Follow us on



GBDi

gbdi.depa.or.th

Facebook



Twitter



govbigdata

Blockdit



YouTube

Government Big Data Institute
(GBDi)



Line
Official

@gbdi

