Python 101

Michael

January 21, 2022

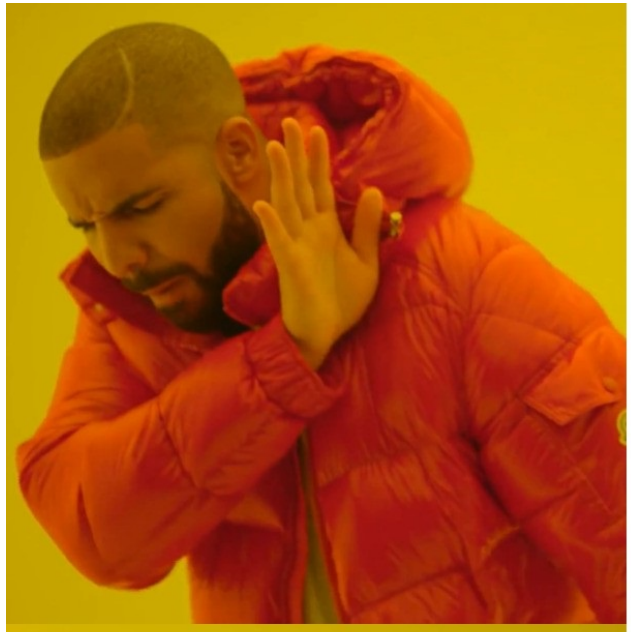
Outline

- What is python
- Variable and `print()`
- Arithmetic and Bitwise Operator
- Statement and Condition
- Function
- List
- For Loop and While Loop
- Dictionary
- Conclusion

How to use Jupyter Lab ?

1 What is Python

Python is a high-level general-purpose programming language created by Guido van Rossum in 1989. It is one of the most popular programming languages in the world



與家人朋友度過一年

一度溫馨的聖誕節

創造一個程式語言

1.1 Why Python ?

- Works on different OS
- Simple syntax
- Runs on interpreter systems
- Hundreds of libraries
- Can be applied to various fields

1.2 What can Python do ?

- Statistical analysis
- Backend (server) of web applications
- AI and machine learning
- Software applications

2 Variables and `print()`

2.1 Variables

As a coder, we need "variables" to store some data for further use. Here are some basic data types :

- Integers (We will focus on this !)
- Floating-Point Numbers
- Complex Numbers
- Strings
- Boolean Type

2.2 print()

print() is a function helping us to display the value of a variable.

Variables and `print()`

2.2.1 Example

```
a = int(3)
b = int(5)
print(b)
print(a)
```

2.3 Formatted `print`

We can print a formatted string (A string inside `f''` or `f""`)
You can refer to Python variables between { and }

Variables and `print()`

```
a = int(3)
b = str('Michael')
print(f'The value of a is {a}')
print(f'my name is {b}')
```

3. Arithmetic and Bitwise Operators

3.1 Arithmetic Operators

- `+`
- `-`
- `*`
- `//`
- `**`

Arithmetic and Bitwise Operators

Example

```
a = 987
print(a)
a = a - 87
print(a)
print(a // 5)
print(3 ** 3)
```

3.2 Bitwise Operators

Before going into this subsection, we need to understand what binary representation is.

Binary Representation

In decimal representation, 7050 is actually

$$7 \times 10^3 + 0 \times 10^2 + 5 \times 10^1 + 0 \times 10^0$$

What if the base is not 10 ?

Exercise 1

What is the binary representation of 32 ?

Arithmetic and Bitwise Operators

$$0 \times 2^0 + 0 \times 2^1 + 0 \times 2^2 + 0 \times 2^3 + 0 \times 2^4 + 1 \times 2^5 = 32$$

$$100000$$

Exercise 2

What is the binary representation of 102 ?

Arithmetic and Bitwise Operators

$$0 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 + 0 \times 2^3 + 0 \times 2^4 + 1 \times 2^5 + 1 \times 2^6 = 102$$

$$1100110$$

Arithmetic and Bitwise Operators

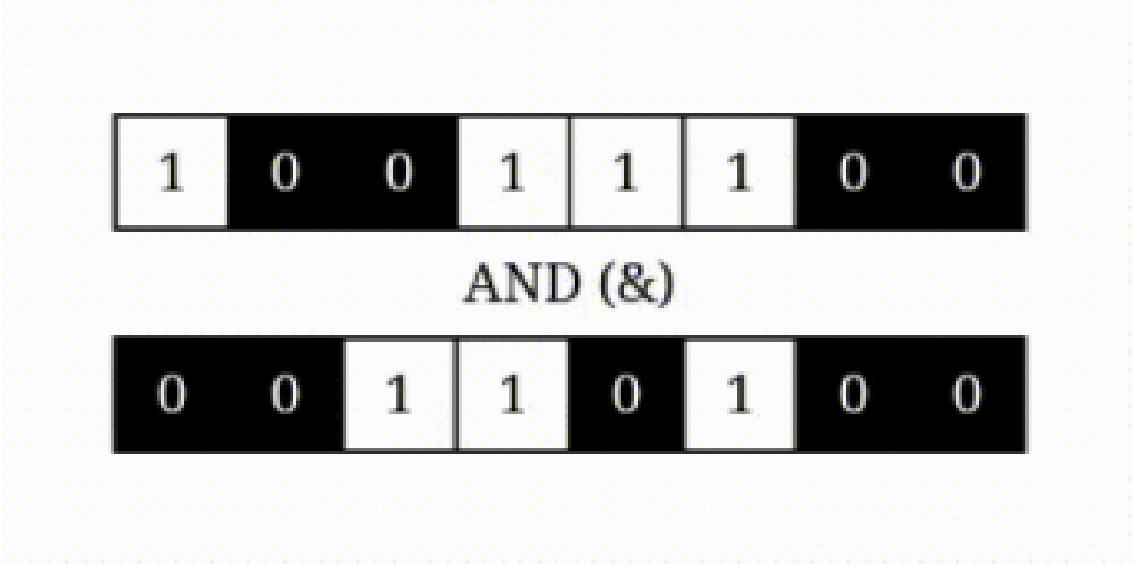
List of Bitwise Operators

- &
- |
- ^
- >>
- <<

Bitwise AND

The bitwise operator AND($\&$) would output the intersection of the two numbers.

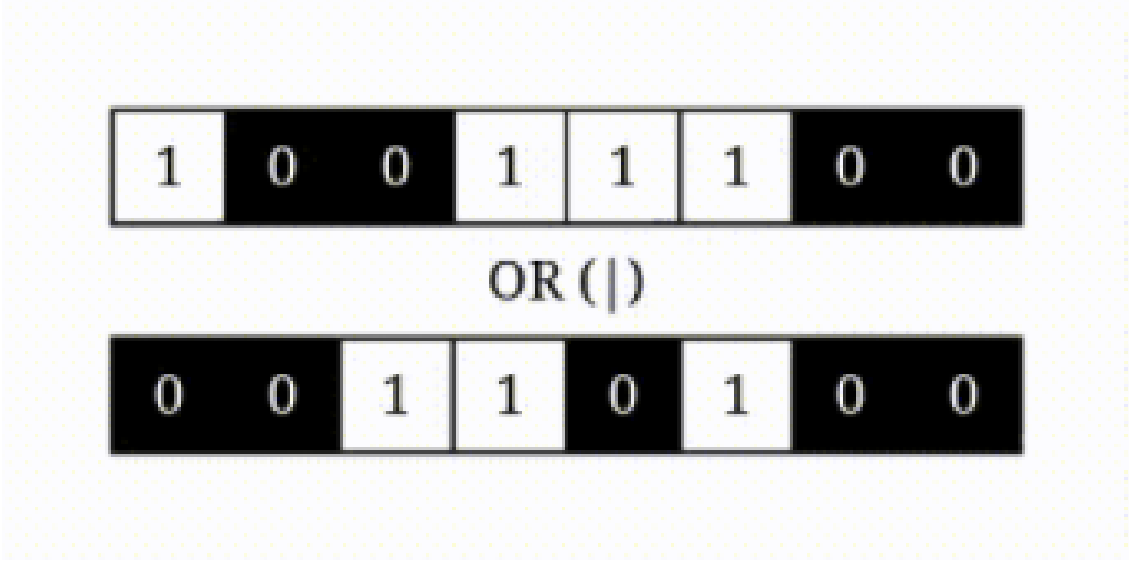
Arithmetic and Bitwise Operators



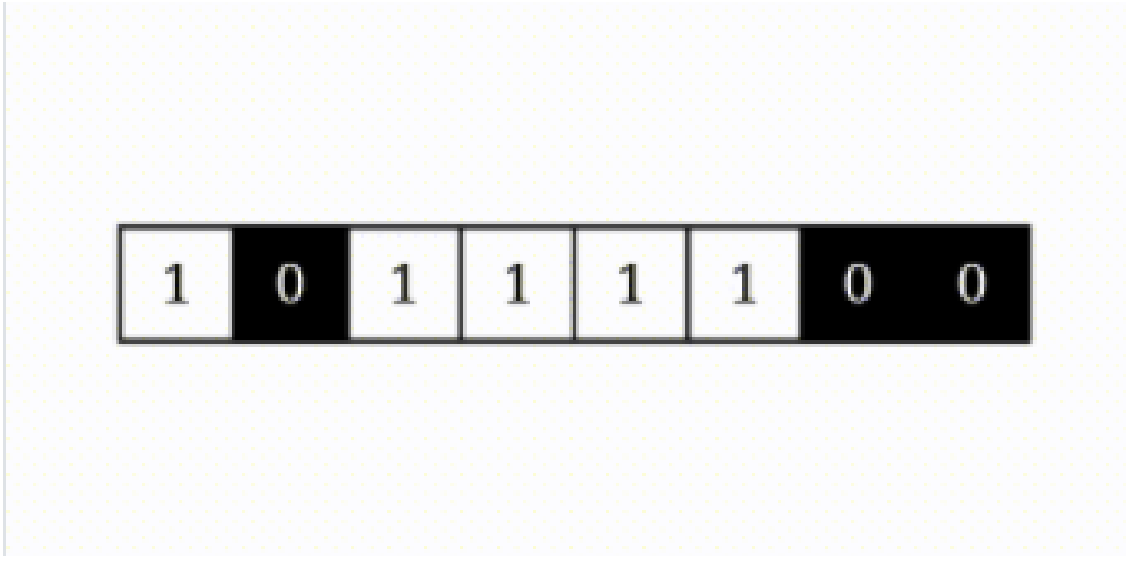
Arithmetic and Bitwise Operators

Bitwise OR

Arithmetic and Bitwise Operators

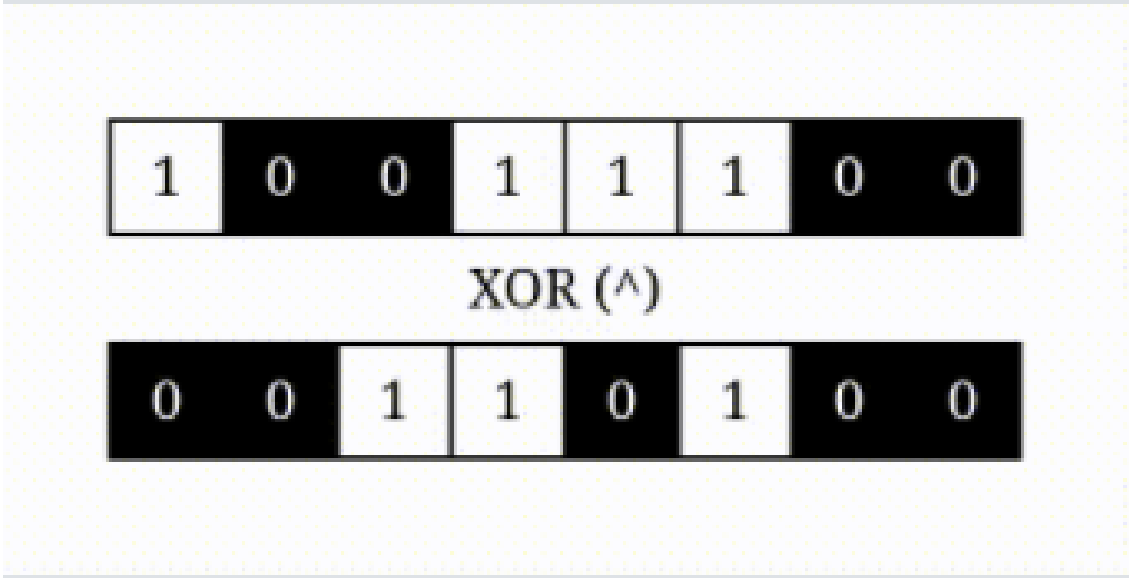


Arithmetic and Bitwise Operators

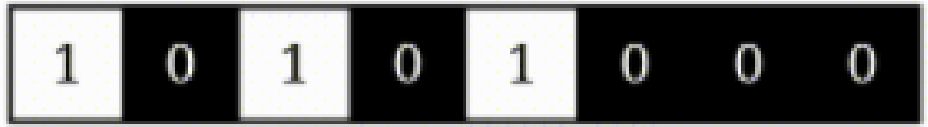


Bitwise XOR

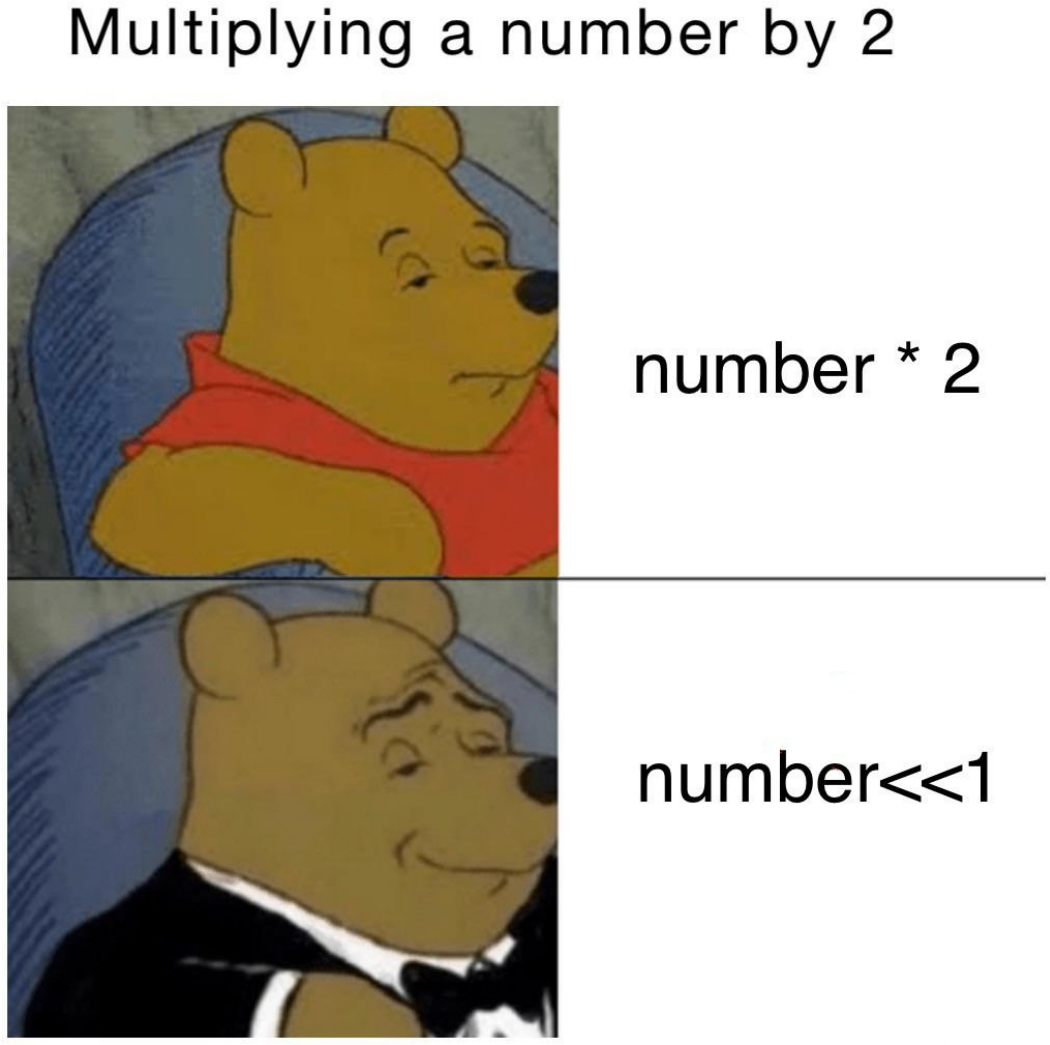
Arithmetic and Bitwise Operators



Arithmetic and Bitwise Operators



Arithmetic and Bitwise Operators



Arithmetic and Bitwise Operators

Assignment Operators

- `=`
- `+=`
- ⋮
- `//=`
- `&=`
- ⋮
- `<<=`

Arithmetic and Bitwise Operators

Example

```
a = 5
a = a + 5
print(a)
a += 5
print(a)
```

4 Statement and Conditions

Statement and Conditions

Sometimes, we may want our program do different things based on different conditions.

- `if`
- `else`
- `elif`

Example

```
a = 529
if (a % 2 == 0):
    print("Even")
else :
    print("Odd")
```

Statement and Conditions

Moreover, things are usually complicated so we need some "conjunctions".

- `and`
- `or`

Statement and Conditions

Example

```
a = 200
b = 33
c = 500
if (a > b and c > a) :
    print("Both conditions are True")
if (b < a or c < a) :
    print("At least one of the conditions is True")
```

Exercise

Write a program to check whether a number is between 1500 and 2700, then check if it's divisible by 7 or 5. If not, your program need to output whether the number is "Out of range" or "Not divisible".

Hints

1. Store the number in a variable
2. Write a if statement to check whether the number is in range
3. Write a if statement inside the previous one and check if the number is divisible by 7 or 5
4. print anything you want if both statement is true
5. Figure out the rest by yourself !

Answer

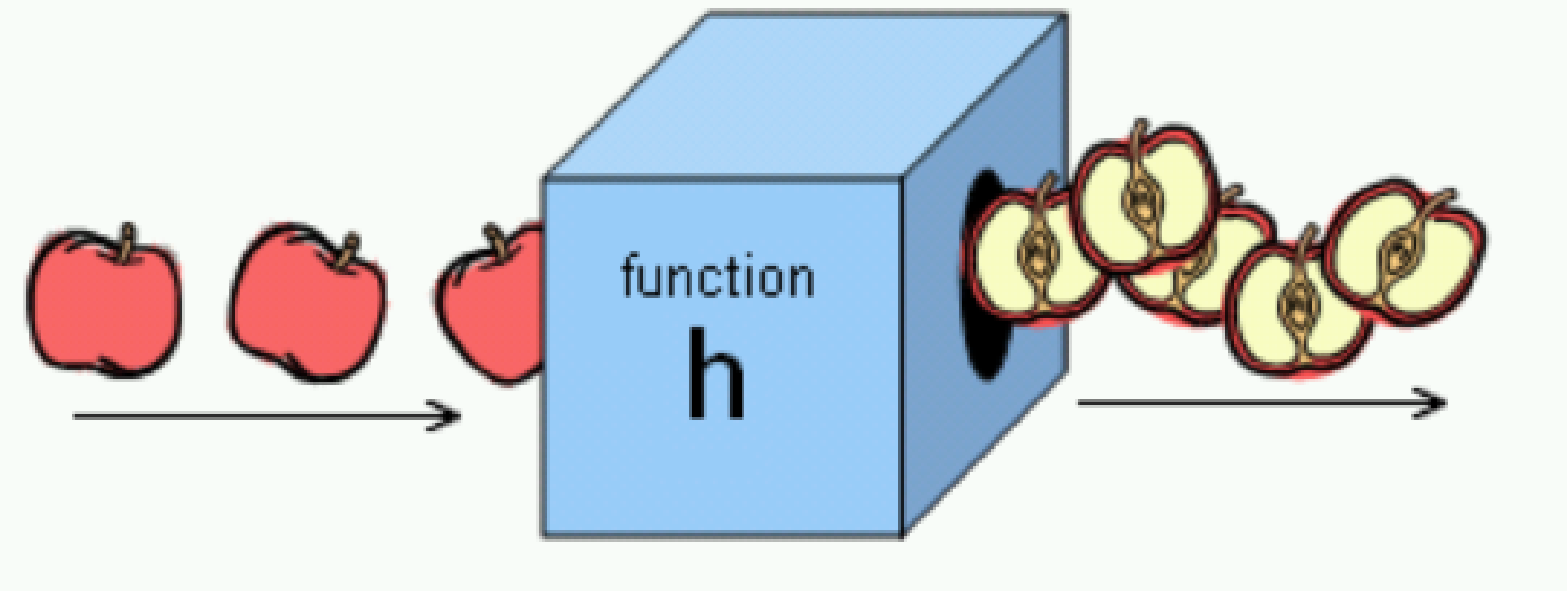
```
a = 438
if (a >= 1500 and a <= 2700) :
    if (a % 7 == 0 or a % 5 == 0):
        print("Test successful")
    else :
        print("Not divisible")
else:
    print("Out of range!")
```

5 Functions

What are Functions

A function is a block of code that does certain calculations and return the result to you.

Functions



Functions

Example

```
def myFunction(num1, num2) :
    return num1**2 + num2**2
```

Why Functions?

- Don't have to copy and paste your code everywhere
- Prevent inconsistency
- Easy to manage

If we dont use functions...

```
print("Cherry, Happy New year !")
print("Brian, Happy New Year !")
print("Hubert, Happy New Year !")
print("Tommy, Happy New Year !")
```

With functions

```
def greetings(name):
    print(f"{name}, Happy New Year !")
greetings("Cherry")
greetings("Brian")
greetings("Hubert")
greetings("Tommy")
```


收到罐頭祝福的朋友
你：Function讚啦

Exercise

We define $A \oplus B = A \times B + A$,
write a Python code to calculate $(10 \oplus 4) * (5 \oplus 8) - (2^8)$
(Requirement : Define a function `oplus` to do $\oplus$ operation)

Answer

```
def oplus (A, B):
    return (A * B + A)
print (oplus(10, 4) * oplus(5, 8) - 2**8)
# 1994
```

6 Lists

Motivation

Suppose that we have 5 numbers
How do we store it ?

```
num1 = 3
num2 = 6
num3 = 23
num4 = 97
num5 = 7414
```

Lists

What if there are 1000 numbers ?

We can put them inside a `list` !

Constructing a List

Lists can be created using square brackets.

`my_list = [5, 60, 95, 33, 83]`

Or you could use `list()` :

- `list()` ⇒ []
- `list([1, 2, 3])` ⇒ [1, 2, 3]
- `list(range(5))` ⇒ [0, 1, 2, 3, 4]
- `list(range(0, 10, 2))` ⇒ [0, 2, 4, 6, 8]

List

Range Functions !

It creates a sequence of numbers

- `range(6)` → `[0, 1, 2, 3, 4, 5]`
- `range(1, 7, 2)` → `[1, 3, 5]` (No 7 !)

Lists

We use "index" to access elements in a list.
The first item in lists has index 0, the second item has index 1, etc.
For example, we can use `my_list[2]` to access the third element in the list.

Lists

Examples

L = [5, 10, 15, 20, 25, 30, 35, 40, 45, 50]

$$\texttt{L[2]} \Rightarrow 15$$
$$\texttt{L[-2]} \Rightarrow 45$$
$$\texttt{L[2:5]} \Rightarrow [15, 20, 25]$$
$$\texttt{L[:3]} \Rightarrow [5, 10, 15]$$
$$\texttt{L[6:]} \Rightarrow [35, 40, 45, 50]$$

Add or Remove Elements

- Use `append()` to add element to the end of the list.
  e.g. `my_list.append(50)`
- Use `insert()` to add element to a specific index of the list.
  e.g. `my_list.insert(i, elem)`
- Use `remove()` to remove an element in the list.
  e.g. `my_list.remove(60)`
- Use `pop()` to remove an element in a specific index.
  e.g. `my_list.pop(1)`

Some Functions of Lists

$$\texttt{len([5, 3, 1])} \Rightarrow 3$$
$$\texttt{max([1, 2, 3, 4, 5])} \Rightarrow 5$$
$$\texttt{min([0, 55, 3, 75])} \Rightarrow 0$$
$$\texttt{sum([1, 2, 3, 4, 5])} \Rightarrow 15$$

Exercise

Write a python program to find and remove the largest number in a list, and insert the sum of the list at the end.

Lists

Answer

```
numbers = [15, 67, 23, 99, 25, 44, 73]
maximum = max(numbers)
numbers.remove(maximum)
numbers.append(sum(numbers))
print(numbers)
```

7 For Loop and While Loop

For Loop

We can use for loops to make our program do repetitive things e.g. add from 1 to 5

```
num = 0
for i in range (1, 6):
    num += i
print (num)
```

For Loop and While Loop

You could also use it to iterate through a list

```
L = [5, 2, 88]
for i in L :
    print (i)
```

While Loop

Execute a set of statements as long as a condition is true.

```
i = 0
while (i < 5):
    print (i)
    i += 1
```

Break

We use `break()` to break out of a loop.

Continue

We use `continue` to skip rest of the code and start a new iteration.

Exercise

Write a python code to print from 1×1 to 9×9

For Loop and While Loop

Answer

```
for i in range (1, 10):
    for j in range (1, 10):
        print(f'{i} x {j} = {i*j}')
    print('\n')
```

Dictionary

Dictionary

Suppose we have a list that stores informations about a person.
`["Michael","Chen","NTU","IM","Clown","2001-4-19"]`

Dictionary

What attribute does each index represents ?

Dictionary can help you !

Dictionary

```
thisDict = {
    "First_name": "Michael",
    "Last_name": "Chen",
    "School": "NTU",
    "Department": "IM",
    "Job": "Clown",
    "Birthday": "2001-4-19"
}
```

Conclusion

Conclusion

Now that you've learned the basic syntax for Python,
you can explore various packages for Python !
For example, `numpy`, `pandas`, `matplotlib`, `scipy`...

Thank You For Listening !