Chapter 1 : Python Basic

1.1 Types of python files

- There are two ways to write python, one is using script (extension .py) and interactive python (extension .ipynb).
- Script can be run using interactive mode but interactive python file cannot be executed like normal python script.

1.2 Write and Read

- Python read characters and numeric from left to right, top to bottom.
- To write any character in Python requires single quote ‘’ or double quote “”. You can use either one but please be consistent and choose only one choice.
- To write numbers or numeric, one must be represented by variables.

Example printing character using single quote.

Input:

writeexample.py

```
1  print('Selamat Datang ke SIF1006')
```

Output:

```
Selamat Datang ke SIF1006
```

Example printing numeric values.

Input :

```
1  a = 10
2  b = 100
3  print(a,b)
```

Ouput:

```
10 100
```

Example printing both character and numeric:

Input

```
1  a = 10
2  b = 100
3  print('a=',a, 'b=',b)
```

Ouput:

```
a= 10 b= 100
```

1.3 Python Indentation

- Python uses indentation to indicate a block of code.
- The modern editors like VS Code, Pycharm, Sypder automatically create indentation for your code block when you write these command (for example):
  - if
  - else
  - function

- Example using indentation, see Chapter 2, Section 2.2

1.4 Python comments

- Comments in python start with a **#** and python will ignore them.
- Comments is important in writing program because it will act as simple explanation to your code.
- Comment also useful when we want to prevent Python from executing certain lines (this saves a lot of time to test the code because we do not need to delete the line).

1.5 Python variables

- There are several types of variables in Python : str, int, float, complex
- str is string.
- int is integer.
- float is float (for floating number)
- How we defined string (str)? Using double quotes for example “Saya” or single quotes for example ‘Saya’
- Variables names
  - Python variable names are case sensitive meaning a ≠ A. Example apple ≠ Apple
  - Variables name must start with alphabet or underscore. Example Orange or _Orange
  - Variables name cannot start with numbers. Example 1234 is not allowed
  - Variables name must be a single word. Example Myanswer (allowed) , My Answer (not allowed), My_answer (allowed).
  - Variables name cannot be the same as **Python keywords** for example: for, break, class

1.6 Python data types

- In Python language, data type is important for different type of application. Below is the list of build-in Python data type:

Types	Command
Text	str
Numeric	int, float, complex
Sequence	list, tuple, range
Set	set, frozenset
Boolean	bool

Input:

```python
a = 5
b = 10.0
c = 2j
d = 'Hello World'
e = [1,2,3]
f = (1,2,3)
g = False

print(type(a))
print(type(b))
print(type(c))
print(type(d))
print(type(e))
print(type(f))
print(type(g))
```

Output :

```
<class 'int'>
<class 'float'>
<class 'complex'>
<class 'str'>
<class 'list'>
<class 'tuple'>
<class 'bool'>
```

1.7 Python operators

- Python is a powerful for numerical calculations. It can handle both single and double precision calculations (up to 10^{55}).
-
- Below is the example of python operators

Operator	Name	Example
+	Addition	x+y
-	Subtraction	x-y
*	Multiplication	x*y
/	Division	x/y
%	Modulus	x%y
**	Exponential	x**y
//	Floor division	x//y

- Some of the operators are embedded in numpy (numerical python) module like exp (exponential). Depending on sensitivity of the numbers (singularity due to division by zero, one need to see either using numpy modules or basic python is more suitable).

Operator.py > ...

```
#Operators in Python

a = 5
b = 10

#addition
c = a+b
print(c)

#subraction
d = a-b
print(d)

#division
e = a/b
print(e)

#multiplication
f = a*b
print(f)

#division(integer)
g = a//b
print(g)

#modulus
h = a%b
print(h)

#exponential
i = a**10
print(i)
```

Output:

```
15
-5
0.5
50
0
5
9765625
```