Python Documentation

By Noor Raihan

Chapter 1 : Installation

Windows Version

1. Visit https://www.python.org/downloads/ and choose which version are suitable. In this case, we are using Python3 since it have the latest update.
2. After finish the downloading, run the executable Installer and make sure to **tick on *Add Python 3.x to the path***

3. Verify if the Python is successfully installed.

    - Open the command prompt.
    - Type 'python' and press enter.
    - The output should be like this if the python is successfully installed.

```
C:\Users\User>python
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May  3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

4. Verify if pip is executable through the python command.

    - Open the command prompt.

- Type this command in the command prompt and press enter.

```
python -m pip -V
```

- If the installation successfull, you may see the pip version displayed as below.

```
C:\Users\User>python -m pip -V
pip 21.1.2 from C:\Users\User\AppData\Local\Programs\Python\Python39\lib\site-packages\pip (python 3.9)
```

5. Now you have successfully install python in your windows machine.

Ubuntu version

1. Open the terminal.
2. Enter the command below in the terminal.

```
sudo apt install python3
```

3. Enter 'python' in the terminal.
4. Your're done.

Chapter 2 : Basic Syntax

1. Executing Python program

All python programs and files are ended with **.py** as file extension. It can be executed by using the terminal or command prompt. Both are using the same concept and same command.

You can try create a file called **hello.py** and type this code in the file.

```
print("Hello world")
```

After that, type the following command in the terminal and make sure the terminal directory are in the same file as the python file.

```
python hello.py or python3 hello.py
```

```
python-learn ➤ python3 hello.py
Hello world
```

2. Printing and displaying text

All programming language start with displaying **Hello World**. In Python3, displaying text are very simple.

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

3. Comments for life

Commenting are very important in explaining what's goin on with our code. It will make the code more readable.

This is how we comment in python3

```
#This is the comment
```

```
#This code display hello world
print("hello world")
```

```
#This is how
#we multiline
#comment
```

4. Variablesssssss

Declaring variable in python are quite simple. You do not need to declare any particular data type and even change the same variable with another datatype.

```
number = 7 #data type is int
number = "seven" #data type is string now
```

You can display the variable using print right away.

```
number = 7
print(number)
```

You can display multiple variable too.

```
x = "This is"
y = "number"
num = 7
print(x, y, num)
```

Print the length of the string

```
x = "Hello"
print(len(x)) #output would be 5
```

Access every letter in string

```
x = "hello"
print(x[0]) #output would be "h"
```

If you really want to set the data type, you can use casting on it.

```
number = str(7) #now it "7"
number = int(7) #now it 7
number = float(7) #now it 7.0
```

5.Playing with numbers and operators

Check which number is minimum

```
print(min(4,7)) #output would be 4
```

Check which number is maximum

```
print(max(4,7)) #output would be 7
```

Round up the decimal

```
print(round(3.5)) #output would be 4
```

Using operater in python

```
x = 3
y = 6

print(x + y) #output 9
print(x - y) #output -1
print(y / x) #output 2.0
print(y % x) #output 0
```

6. Conditional If..Else

In conditional statement in python, indention are very important and wrong indention can return to syntax error.

If statement

```
a = 20
b = 150

if b > a:
        print("b is larger than a")

b is larger than a
```

elif statement

```
a = 20
b = 20

if b > a:
        print("b is larger than a")
elif b == a:
        print("b is equal to a")

#output would be "b is equal to a"
```

else statement

```
a = 200
b = 50

if b > a:
        print("b is larger than a")
elif b == a:
        print("b is equal to a")
else
        print("a is larger than b")

#output would be "a is larger than b"
```

And statement (both condition must return true)

```
a = 200
b = 33
c = 500

if a > b and c > a:
        print("Both conditions are True")
```

Or statement

```
a = 200
b = 33
c = 500

if a > b or a > c:
         print("At least one of the conditions is True")
```

Nested if

```
x = 41

if x > 10:
         print("Above ten,")
         if x > 20:
                  print("and also above 20!")
         else:
                  print("but not above 20.")
```

Pass statement (used when there is nothing to put in if statement and wanted to avoid any error)

```
a = 33
b = 200

if b > a:
         pass
```

7. User input

User input is very important in every programming language. This is how we receive user input in python3.
Remind: input in python always return as string

```
name = input("What is your name: ")
print("Your name is", name)
```

8. Array

Array is very important in a programming language. Array make us as programmer able to store multiple temporary data or value in single variable.

we can imagine the array as matrices in modern mathematics

```
[12,15,16
 13,18,20
 10,12,78]
```

similar to matrices right? let's convert it into the array

```
nums = [[12,15,16]
                [13,18,20]
                [10,12,78]]
```

this is what we called 2 dimensional array and it's similar to the matrices concept and we can access it like this

```
print(nums[0][1]) #access the first row and second data which is 15
```

This is how we can create an basic array in the python

```
cars = ["Proton", "Perodua", "MYVI"]
```

So we can access the above array by mentioning its index and it's always start with 0

```
print(cars[0]) #Proton
```

we also can append and remove the array

```
cars.append("Tesla") #add Tesla into the array
cars.pop(1) #remove Perodua from the array
```

9. Loop

There is two type of loops in python.

- **while** loops
- **for** loops

While Loop

while loops is a loops that can execute as long as the condition is **true**.

```
i = 1 #declare the counter

while i < 6: #condition for while loop
        print(i)
        i += 1 #update the counter
```

break statement (it will stop the loop if the condition is true)

```
i = 1

while i < 6:
        print(i)

        if i == 3: #if i is equal to 3 then the loop will stop immediately
                break

        i += 1
```

continue statement (it will stop the currect loop status and continue to the next loop)

```
i = 0

while i < 6:

        i += 1
        if i == 3: #if i is equal to 3 then it will continue to next loop
                continue

        print(i)
```

For loops

for loop is used to iterate over sequence such as list and arrays

```
cars = ["Proton", "Honda", "Toyota"]
```

```
for x in cars:
        print(x)
```

interestingly it also can be used to iterate through string

```
for x in "proton":
        print(x)
```

break statement also can be used in for loops

```
cars = ["Proton", "Honda", "Toyota"]

for x in cars:
        print(x)
        if x == "Honda":
                break
```

same goes to the continue statement

```
cars = ["Proton", "Honda", "Toyota"]

for x in cars:
        if x == "Honda":
                continue
        print(x)
```

if we have our own counter and range, we can use **range()** in for loops as counter

```
#it will loop 6 times
for x in range(6):
        print(x)
```

we also can put starting point using range (range(start, end))

```
for x in range(2, 6):
        print(x) #2,3,4,5
```

10. Function

Function is very important in programming and we can have abstract function or user-defined function

There got operation to get the value and this we called as function
eg: TodayDate(), MaxValue(), Min()

Some of the function already built-in python and we call it abstract function
eg: max(), min(), round()

Function can be implemented in python by using **def** (definition)

```
def hello():
        print("hello")

hello() #output is "hello"
```

def in python also can received any parameter

```
def hello(text):
        print(text)

hello("hello")
```

or return anything

```
def hello():
        return "hello"

print(hello()) #output is "hello"
```

```
def isTrue():
        return True

print(isTrue()) #output is True
```

but what if we does not know the value but we know that the value is increase by summation or subtraction??

We will use the technique called RECURSIVE!

Let's take a look in factorial concept.
We knew that factorial 3 is equal to 3×2×1=6 but how do it in python?

```
def factor(num):
        if num == 1:
                return num
        else
                return num*factor(num-1)

print(num(3))
```