

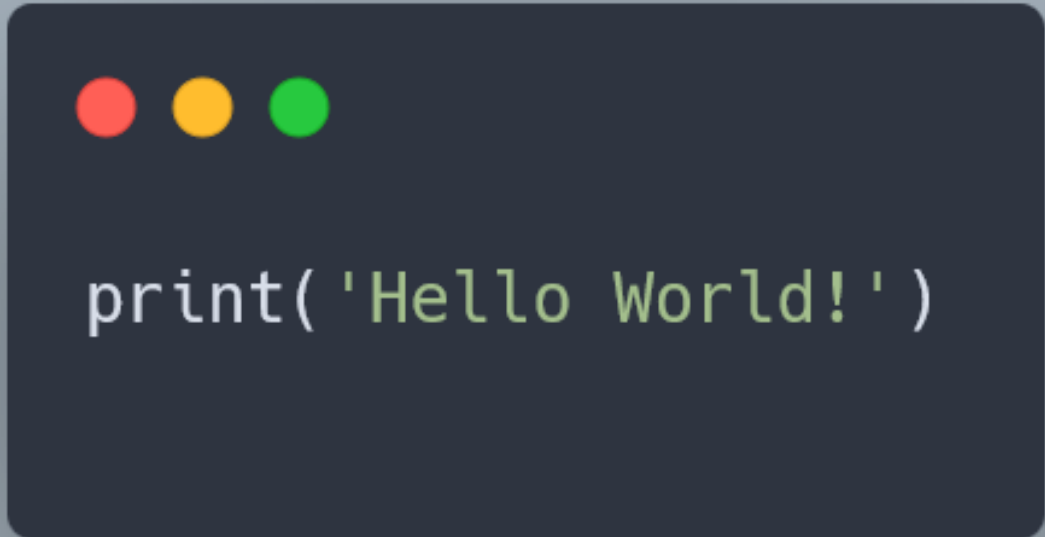
KELAS OTOMESYEN



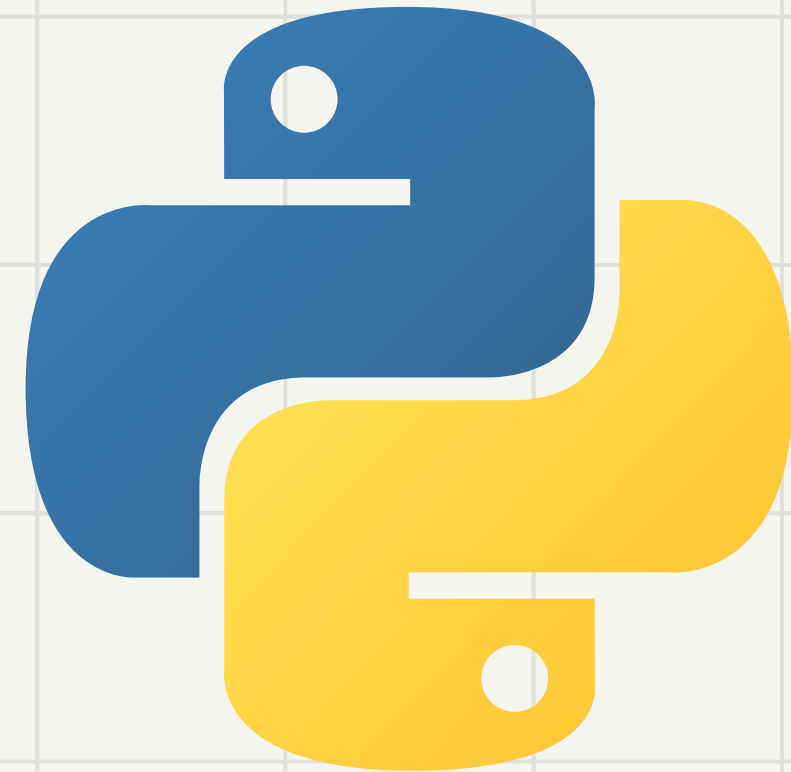
Python Basic



Hello World!



```
print('Hello World!')
```



Running Python

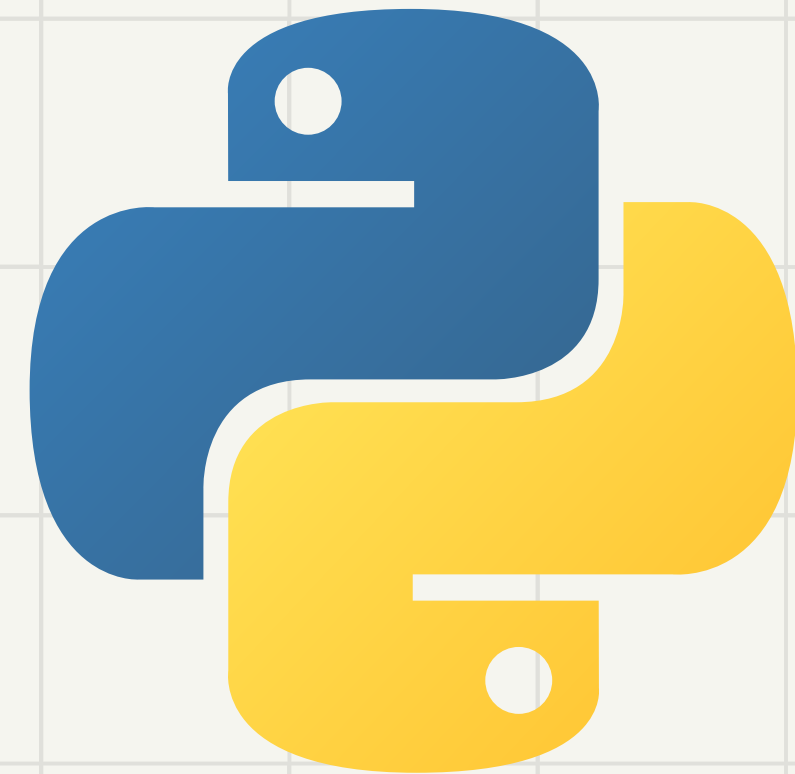
Python IDE

```
C:\Users\IdeJongkok>python
```

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

Command Prompt

```
C:\Users\IdeJongkok>python namafile.py
```



Comment

```
#ini comment  
print("Hello, World!") #ini juga comment  
  
#print("ini gak bakal di run codenya, karena ini comment juga")  
  
"""  
ini juga comment  
tapi multiline  
biasa dipake buat deskripsi sesuatu  
"""
```



Variabel



```
x = 5  
y = "Marsudi"  
nilai = 2  
Nilai = [3,2,1]  
NILAI = True  
nilai_bilangan = 3  
_nilaiBilangan = 4
```



```
nilai bilangan = 1  
nilai-bilangan = 1  
1nilai = 1
```

Tipe Data

TEXT TYPE: STR

NUMBER: INT, FLOAT

STORE DATA TYPES: LIST, TUPLE, SET, DICT

BOOLEAN TYPE: BOOLEAN

Number



```
x = 1    # int
```

```
y = 2.3  # float
```

```
z = 4j    # complex
```

String



```
print("joko")  
print('joko')
```

```
a = "darmadji"  
print(a)
```

```
a = """Lihat kebunku penuh dengan bunga,  
ada yang putih dan ada yang merah,  
setiap hari kusiram semua,  
mawar melati semuanya indah."""  
print(a)
```

String adalah tipe data berupa rangkaian (Sequence) yang anggotanya berurut (ordered), dan memiliki indeks.



```
#tiap karakter pada string dapat diambil sesuai dengan indexnya  
a = 'Tuan Sugeng'  
print(a[1])
```

```
#Char pada string dapat di-iterasi  
for x in 'Broto':  
    print(x)
```

```
#Jumlah char pada suatu string dapat ditampilkan  
a = 'Marsudi'  
print(len(a))
```


Casting / Ubah tipe data

`int()` - mengubah data menjadi integer

`float()` - mengubah data menjadi float

`str()` - mengubah data menjadi string / text



Boolean



```
#Boolean
```

```
True
```

```
False
```

```
print(10 > 9)  
print(10 == 9)  
print(10 < 9)
```

Setiap String akan bernilai True, kecuali string kosong.

Setiap number bernilai True, kecuali 0.

List, Tupel, Set, dan Dictionary adalah True, kecuali yang kosong.

List

digunakan untuk menyimpan beberapa item dalam satu variabel



```
mylist = ["rengginang", "peyeum", "ulen"]
```

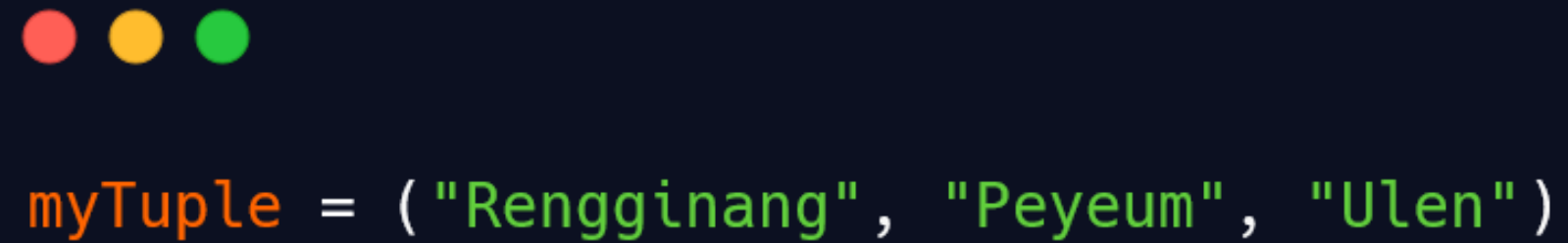
Karakteristik item:

- Ordered
- dapat diubah
- memungkinkan nilai duplikat.

Item list ter-index, item pertama memiliki index [0], item kedua memiliki index [1] dll.

Tuple

digunakan untuk menyimpan beberapa item dalam satu variabel



```
myTuple = ("Rengginang", "Peyeum", "Ulen")
```

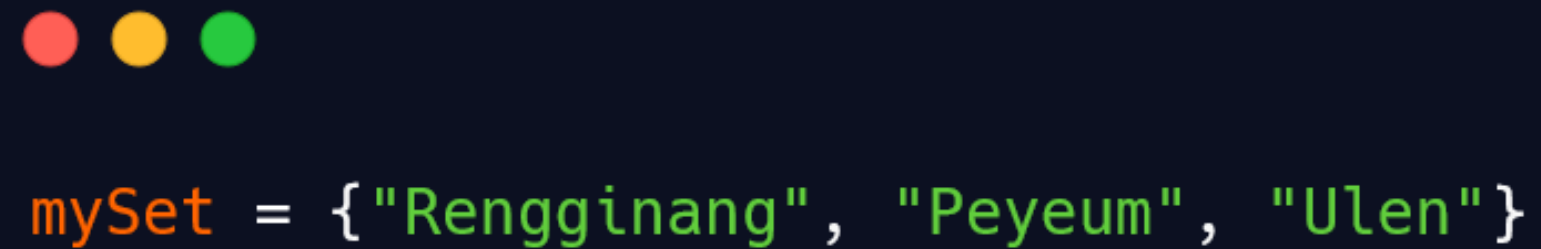
Karakteristik item:

- Ordered
- tidak dapat diubah
- memungkinkan nilai duplikat.

Item Tuple ter-index, item pertama memiliki index [0], item kedua memiliki index [1] dll.

Set

digunakan untuk menyimpan beberapa item dalam satu variabel



```
mySet = {"Rengginang", "Peyeum", "Ulen"}
```

Karakteristik item:

- Unordered
- tidak dapat diubah
- tidak boleh ada nilai duplikat.

Tidak ada index

Dictionary

digunakan untuk menyimpan beberapa item dalam satu variabel

```
ini_dict = {  
    "nama": "sentot",  
    "umur": 56,  
    "pekerjaan": "Ketua RT"  
}  
  
print(ini_dict["nama"])
```

Karakteristik item:

Pada Python versi 3.7 keatas, Dict ordered.

Di Python 3.6 dan sebelumnya, Dict unordered.

- dapat diubah
- duplikat (key) tidak diperbolehkan.

Dict memiliki Key & Value.

Python Operators

Operator aritmatika

Operator	Deskripsi	Contoh
+	Penjumlahan	<code>x + y</code>
-	Pengurangan	<code>x - y</code>
*	Perkalian	<code>x * y</code>
/	Pembagian	<code>x / y</code>
%	Modulo	<code>x % y</code>
**	Eksponensial	<code>x ** y</code>
//	Floor division	<code>x // y</code>

Python Operators

Operator perbandingan

Operator	Deskripsi	Contoh
==	Sama dengan	x == y
!=	Tak sama dengan	x != y
>	Lebih dari	x > y
<	Kurang dari	x < y
>=	Lebih atau sama dengan	x >= y
<=	Kurang atau sama dengan	x <= y

Python Operators

Operator Logika

Operator	Deskripsi	Contoh
and	akan menghasilkan nilai True apabila kedua statement benar	<code>x<5 and x<10</code>
or	akan menghasilkan nilai True apabila satu statement benar	<code>x<5 or x<4</code>
not	akan menghasilkan nilai False apabila menghasilkan benar	<code>not(x<5 and x<10)</code>

Condition / if statement



```
a = 20  
b = 30
```

```
if b > a:  
    print("b lebih besar dari a")  
elif a == b:  
    print("a dan b sama")  
else:  
    print("a lebih besar dari b")
```

Loops

while Loop

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

Dengan while Loop, kita dapat mengeksekusi satu set statement selama kondisinya True.

for Loop

```
bapack = ["bambang", "saron", "juned"]
for x in bapack:
    print(x)
```

for Loop digunakan untuk iterasi sebuah Sequence / Rangkaian (yaitu List, Tupel, Dict, Set, atau String).

Function

```
def sapa(nama):  
    """Fungsi ini untuk menyapa seseorang  
    sesuai nama yang dimasukkan sebagai parameter"""  
    print("Oi, " + nama + ". wasap broh?")  
  
# pemanggilan fungsi  
sapa('Jajang')  
# output: Oi, Jajang. wasap broh?
```

```
def ngitung(a,b):  
    c = a + b  
    return c  
  
# pemanggilan fungsi  
hasil = ngitung(2,3)  
print(hasil)  
# output: 5
```

Argument



```
def ini_fungsi(nama):  
    print(nama + " Joko")
```

Argument



```
def ini_fungsi(nama, jabatan):  
    print(nama + " adalah seorang " + jabatan)
```

Arbitrary Argument

```
def inifungsi(*warga):  
    print(warga[2] + " adalah Warga RT 01")
```

```
my_function("Darmadji", "Kokom", "Ubay")
```

REFERENSI

- <https://www.w3schools.com/python/>
- <https://automatetheboringstuff.com/>