

## Pengertian

1. Fungsi pada python adalah kumpulan perintah atau baris kode yang dikelompokkan menjadi satu kesatuan untuk kemudian bisa dipanggil atau digunakan berkali-kali.
2. Sebuah fungsi dapat menerima parameter, dapat mengembalikan suatu nilai, dan dapat dipanggil berkali-kali secara independen.

## Keuntungan

1. Membagi kode program menjadi bagian-bagian kecil dengan tugasnya masing-masing.
2. Membuat kode program menjadi lebih "reusable" dan lebih terstruktur.

## Sintaks

```
def function_name():  
    Logic Program
```

Cara memanggil fungsi dengan menuliskan function\_name diikuti kurung buka tutup () beserta parameter jika ada

```
function_name()
```

Buatlah fungsi dengan nama "helloPython" yang berfungsi untuk mencetak "Welcome in Python Language"

In [2]:

```
def helloPython() :  
    print("Welcome in Python Language")  
  
helloPython()
```

Welcome in Python Language

## Fungsi dengan parameter

Sebuah fungsi dapat menerima parameter ataupun argumen yang merupakan suatu nilai/variabel yang dilemparkan ke dalam fungsi untuk diproses lebih lanjut.

## Sintaks

```
def function_name(param):  
    Logic Program
```

Cara memanggil fungsi dengan menuliskan function\_name diikuti kurung buka tutup () beserta parameter

```
function_name(param)
```

Buatlah fungsi dengan nama "fullName" yang mempunyai parameter "firstname" dan "lastname" digunakan untuk mencetak "firstname" dan "lastname"

In [3]:

```
def fullName(firstname, lastname) :  
    print(firstname, lastname)  
  
fullName("Axa", "Rajandrya")
```

Axa Rajandrya

Parameter fungsi diperbolehkan menggunakan lebih dari 1, dimana paramater tersebut ada yang wajib harus diisi dan ada yang tidak harus diisi

```
def function_name(param_1, param_2, param_3,...):  
    Logic Program
```

Cara memanggil fungsi dengan menuliskan function\_name diikuti kurung buka tutup () beserta parameter

```
function_name(param_1, param_2, param_3)
```

Buatlah fungsi dengan nama maxValue yang mempunyai parameter "val\_1","val\_2" dan "val\_3" yang bertujuan untuk mencari nilai terbesar dari 3 nilai tersebut

In [4]:

```
def maxVal(val_1, val_2, val_3) :  
    val = [val_1, val_2, val_3]  
    print(max(val))  
  
maxVal(95, 81, 94)
```

95

Parameter opsional digunakan pada fungsi dengan cara memberikan nilai default, artinya nilai parameter sudah diberikan terlebih dahulu tanpa dipanggil

```
def function_name(param_1, param_2, param_3 = 'Nilai'):  
    Logic Program
```

Buatlah fungsi "countCircleArea" dengan 2 parameter yaitu "phi" dan "diameter" dimana parameter phi mempunyai nilai default 3.14

In [6]:

```
def countCircleArea(diameter, phi=3.14) :  
    print(phi*diameter)  
  
countCircleArea(5)
```

15.700000000000001

Fungsi dengan 2 Parameter opsional juga dapat dilakukan pada Python.

```
def info(suhu, daerah='Sukabumi', satuan = 'Celcius'):  
    print(f"Suhu sekarang di {daerah} : {suhu} {satuan} ")
```

Adapun Cara memanggilnya

```
info(30)
```

In [7]:

```
def info(suhu, daerah='Sukabumi', satuan = 'Celcius'):  
    print(f"Suhu sekarang di {daerah} : {suhu} {satuan} ")  
  
info(30)
```

Suhu sekarang di Sukabumi : 30 Celcius

## Fungsi dengan Return Value

Yaitu fungsi dimana akhir dari programnya adalah nilai kembalian atau nilai balik. Artinya nilai dalam fungsi dapat ditampung lagi ke variabel lain untuk digunakan operasi lebih lanjut.

Buatlah fungsi dengan return value yang digunakan untuk mengecek sebuah bilangan termasuk bilangan negatif, bilangan netral(0) dan bilangan positif

In [8]:

```
def cekVal(val) :  
    return val  
  
print(cekVal("Test"))
```

Test

## Soal Latihan

1. Buatlah fungsi untuk menjumlahkan total nilai dari list
2. Buatlah fungsi untuk mencari nilai terbesar dari sekumpulan list
3. Buatlah fungsi untuk menjumlahkan 2 buah list

In [12]:

```
#Soal 1  
def sumList(list) :  
    sum_list = sum(list)  
    print(sum_list)  
  
ex_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
sumList(ex_list)
```

55

In [13]:

```
#Soal 2
def maxList(list) :
    max_list = max(list)
    print(max_list)

ex_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
maxList(ex_list)
```

10

In [14]:

```
#Soal 3
def countList(list_1, list_2) :
    sum_list_1 = sum(list_1)
    sum_list_2 = sum(list_2)
    total = sum_list_1 + sum_list_2
    print(total)

ex_list_1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
ex_list_2 = [20, 50, 30, 70]
countList(ex_list_1, ex_list_2)
```

225

In [ ]: