

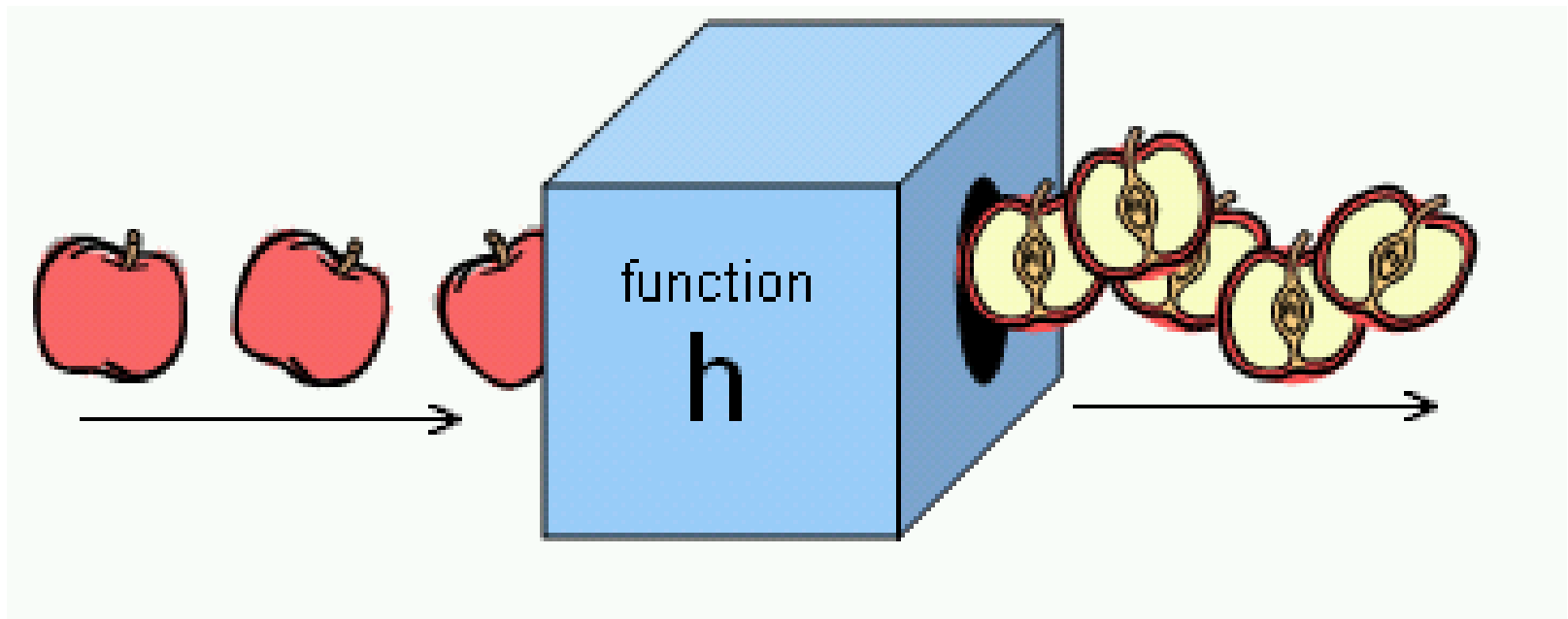
Module 01

Function & List

Data Science Developer

Function

Functions are blocks of code that can be named and reused.



Function

```
def contoh() :  
    print('Halo Dunia!');
```

```
contoh();
```

```
/*  
def namafunc(param) : prog  
*/
```

Function

```
x = 10;
```

```
y = 50;
```

```
def contoh() :  
    print(x+y)
```

```
contoh();
```

Function with a Parameter

```
def namaku(nama) :  
    print(nama + ' Susilo');
```

```
namaku('Adi');  
namaku('Budi');  
namaku('Caca');  
namaku('Dedi');
```

Function with 2 Parameters

```
def data(x,y) :  
    print(x+' Lahir th '+y);
```

```
data('Adi', '1990');  
data('Budi', '1991');  
data('Caca', '1992');  
data('Dedi', '1993');
```

Return Function

```
def total(x,y) :  
    z = x + y;  
    return z;
```

```
print(total(4,5));  
print(z);
```

/*

- z adalah local variabel dalam func total, tidak dapat dipanggil di luar func tsb.
- jika z tidak di-return maka total(4,5) = None

*/

Return Function

```
def total(x,y) :  
    z = x + y;  
  
print(total(4,5));
```

/*

- z adalah local variabel dalam func total, tidak dapat dipanggil di luar func tsb.
- jika z tidak di-return maka total(4,5) = None

*/

Return Function

```
def total(x,y) :  
    z = x + y;  
    print(z);  
  
print(total(4,5));
```

/*

- z adalah local variabel dalam func total, tidak dapat dipanggil di luar func tsb.
- jika z tidak di-return maka total(4,5) = None

*/

Fn inside Fn

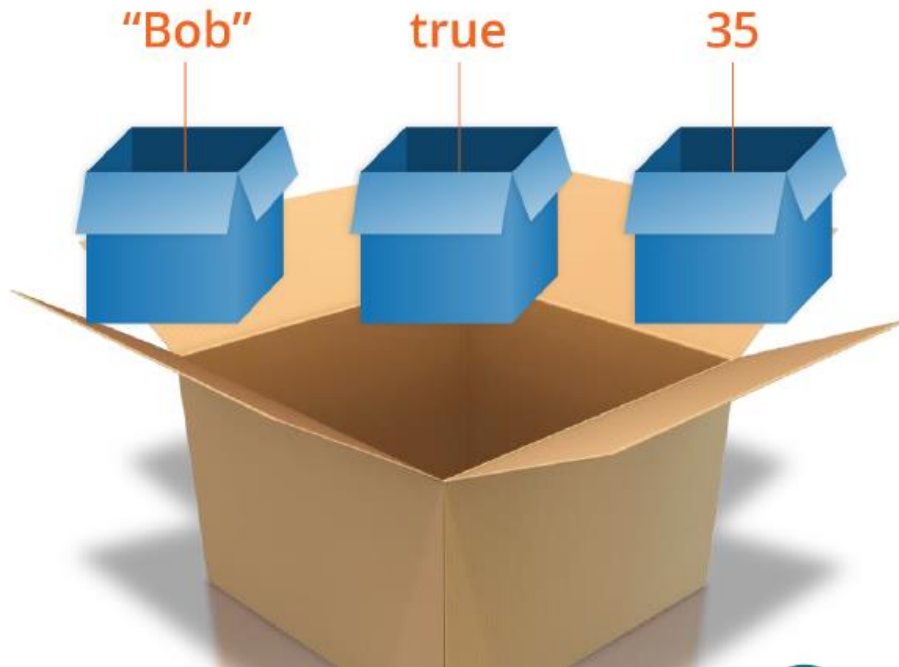
```
def kali(x) :  
    if (x < 2) :  
        return 1;  
    else :  
        return (x * tiga());  
  
def tiga() :  
    return 3;  
  
print(kali(5));
```

Default Parameter

```
def kali(angka1 = 5, angka2 = 2) :  
    return angka1 * angka2;  
  
print(kali(angka2=4));
```

List

Arrays are container-like values that can hold other values. The values inside an array are called elements.



List

```
mobil1 = 'Alya';  
mobil2 = 'Xenia';  
mobil3 = 'Avanza';
```

```
#####
```

```
mobil = ['Alya', 'Xenia', 'Avanza'];  
mobil = [  
    'Alya',  
    'Xenia',  
    'Avanza'  
];
```

Access List Value

```
mobil = ['Alya', 'Xenia', 'Avanza'];
```

```
print(mobil);  
print(mobil[0]);  
print(mobil[1]);  
print(mobil[2]);  
print(mobil[3]);
```

Access List Value

```
buah = ['Jeruk', 'Nanas', 'Apel'];
```

```
for item in buah :  
    print(item);
```

Access List Value

```
buah = ['Jeruk', 'Nanas', 'Apel', 'Mangga'];
```

```
print(buah[1:]);
```

```
print(buah[:3]);
```

```
print(buah[2:4]);
```

```
print(buah[:]);
```


Change List Value

```
buah = ['Jeruk', 'Nanas', 'Apel', 'Mangga'];
```

```
buah[1] = 'Kelapa';  
print(buah);
```

List append & pop

```
buah = ['Jeruk', 'Nanas', 'Apel', 'Mangga'];
```

```
buah.append('Kelapa');
```

```
print(buah);
```

```
buah.pop();
```

```
buah.pop();
```

```
print(buah);
```

List inside List and Diff Type Data

```
listTest = [1, 'hi', ['hello', 2, True, [0, 1]]]
```

```
print(listTest[1]);  
print(listTest[:2]);  
print(listTest[2]);  
print(listTest[2][1:]);  
print(listTest[2][2]);  
print(listTest[2][3][0]);
```

Solve It! #1

**Buatlah algoritma
untuk mengurutkan
elemen array berikut:
 $x = [40, 100, 1, 5, 25, 10]$**

Solve It! #2

**Buatlah algoritma untuk
menentukan elemen
tertinggi & terendah,
dari array berikut:**

$x = [40, 100, 1, 5, 25, 10]$