Module 01

Function & List

Data Science Developer

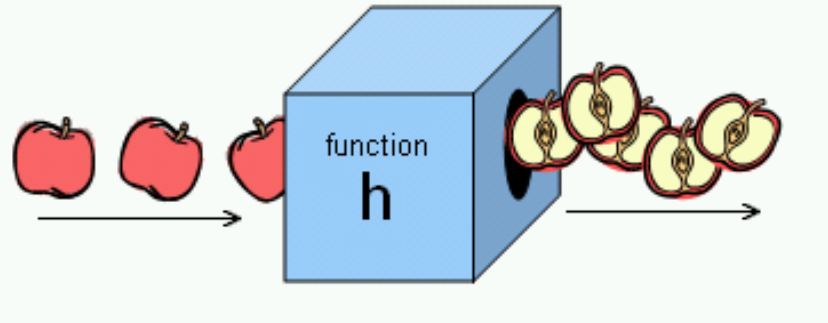
Purwadhika
Startup and Coding School

Outline

- Function:
  - Function without return
  - Function with return
  - A function inside another function
  - Optional parameter
- List
  - What is a list
  - Access list elements
  - List mutability : changes, add, remove
  - List operation : concatenation and repetition
  - List for loop and list comprehension

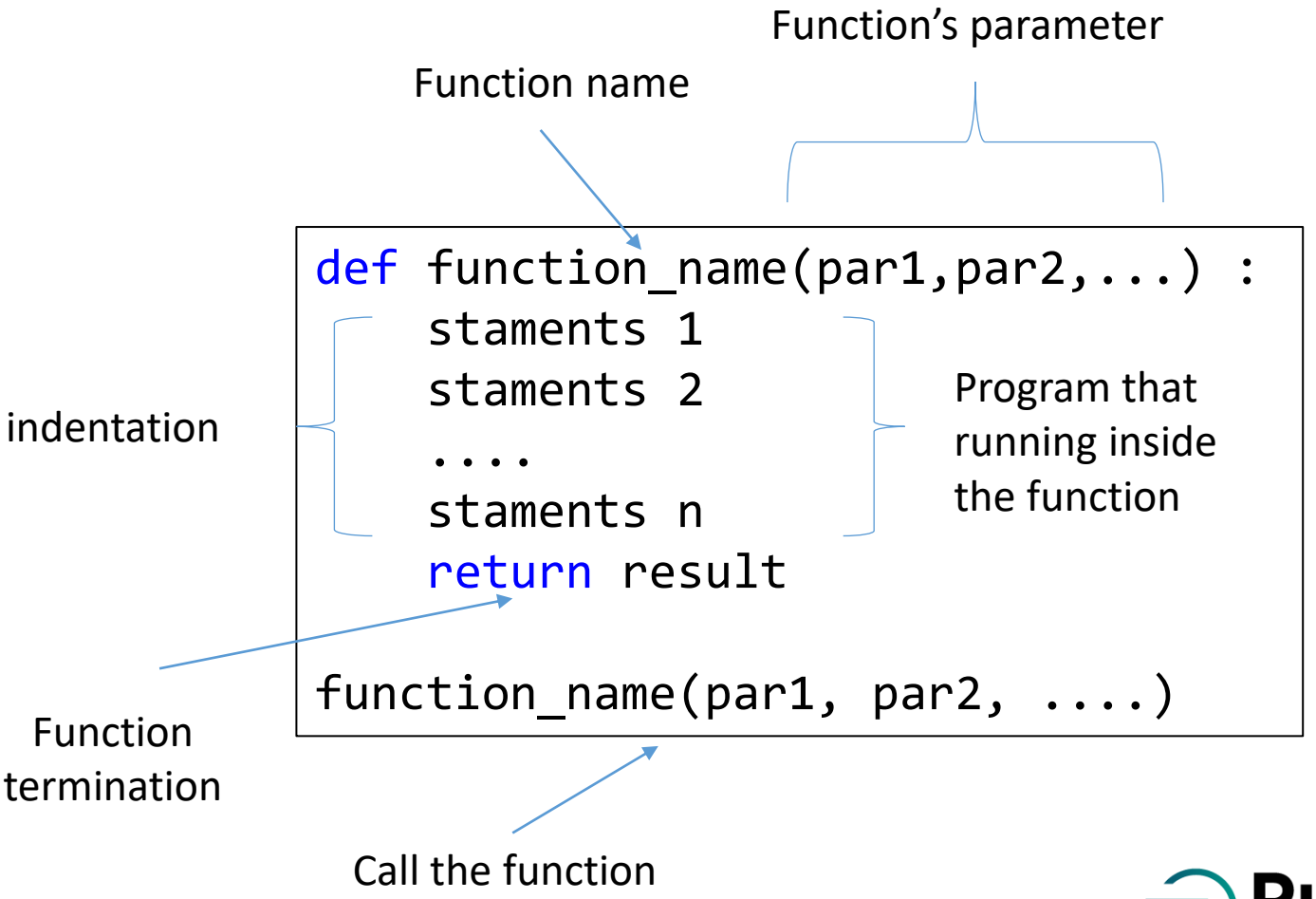
Function

- Functions are blocks of code that can be named and reused.
- Actually, we already used functions in. e.g. : input, print, int. these function already defined in python.
- each functions has a name. e.g. : input, print, int.



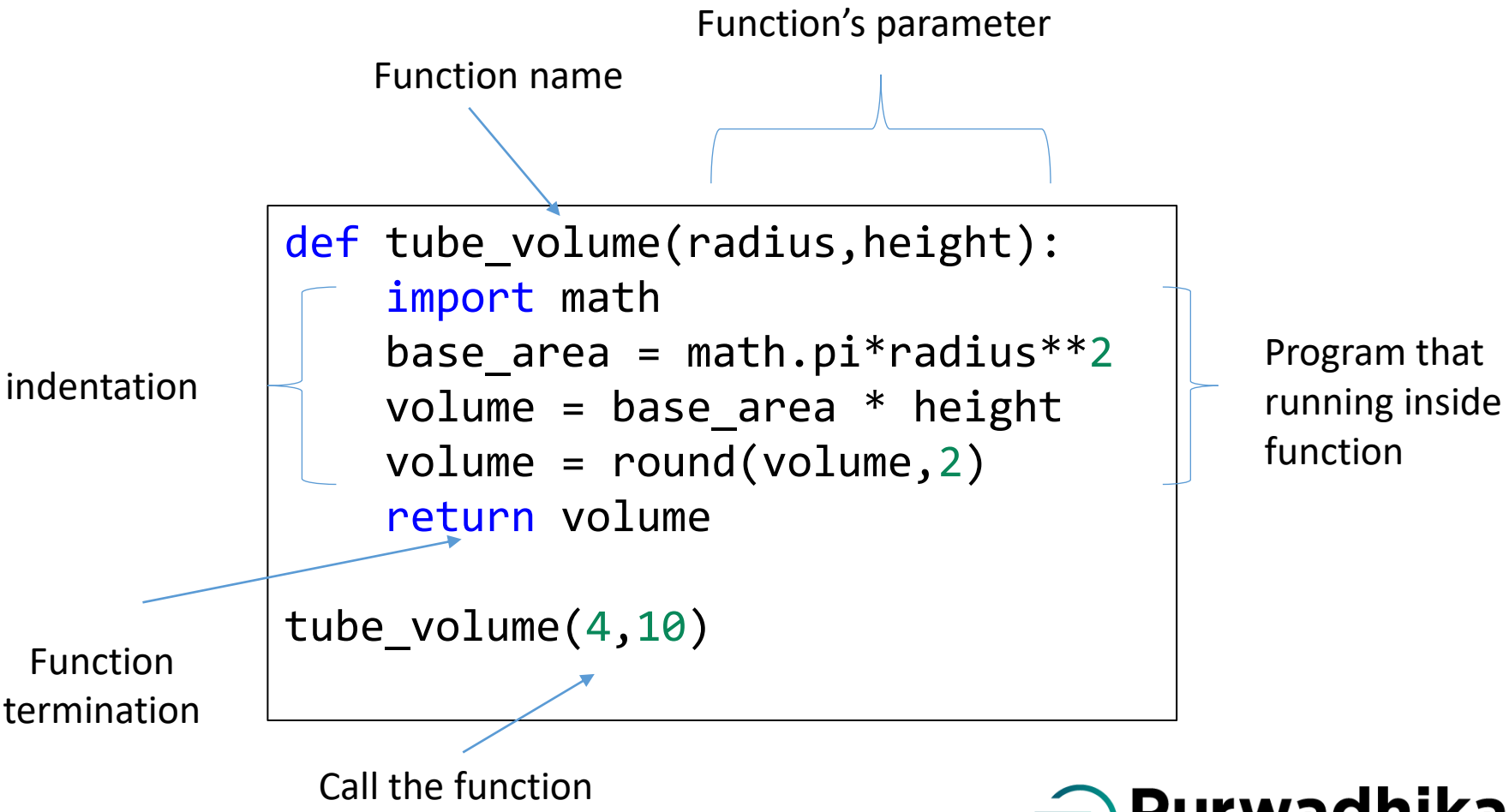
Purwadhika
Startup and Coding School

Structure of a function



Purwadhika
Startup and Coding School

A function to compute volume of a tube



Purwadhika
Startup and Coding School

Why do we need function ?

- Functions are blocks of code that can be named and reused. We can reuse that blocks of code as many times as we want
- If we reuse the blocks of code without making it into function, there are two risk, typo and we have to change everything inside the blocks of code.

Examples :

- Function to compute area of a circle, function to compute volume of a cube, function to compute volume of a tube, etc

Function with return

- A function often compute a return value. The return value is returned back to the caller
- When an function reaches a return statement the function will stop executing
- Function can return any value with certain type
- Function that return any value : input, int, abs
- For example:

```
def diff(a, b):
    if a >= b:
        return a-b
    else:
        return b-a

x = 5
y = 9
print(diff(x,y))
```

```
def times(a, b):
    return a*b

x = 5
y = 9
print(times(x,y))
```

Purwadhika
Startup and Coding School

Function with return

```
def tube_volume(radius,height):
    import math
    base_area = math.pi*radius**2
    volume = base_area * height
    volume = round(volume,2)
    return volume

print(tube_volume(4,10))
```

- Function execute any statement inside the function when called
- Different input parameter different output

Purwadhika
Startup and Coding School

Function without return and arguments

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

contoh() # call the function
```

```
x = 10
y = 50

def contoh() :
    print(x+y)

contoh()
```

- Function execute any statement inside the function when called

Purwadhika
Startup and Coding School

Function without return

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

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

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

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

- Function execute any statement inside the function when called
- Different input parameter different output

Local variable vs global variable

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

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

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

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

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


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

- z is local variable in total function, z cant be called outside the function
- If total function doesn't have return then the return will be None , total(4,5) = None

Purwadhika
Startup and Coding School

Local variable vs global variable

- Local variable : variable that defined inside a function
- Global variable : variable that defined outside any function

```
def func1():
    return (x + 10)


x = 5
print(func1(), x)
```

```
def func2():
    x += 1
    return (x + 10)

x = 5
print(func2(), x)
```

- func1: 15 5
  - x is a global variable and used as
- func2 : UnboundLocalError: local variable 'x' referenced before assignment
  - Local variable ‘x’ undefined
  - func2 expect local variable ‘x’

Purwadhika
Startup and Coding School

Local variable vs global variable

```
def func3():
    global x
    x += 1
    return (x + 10)

x = 5
print(func3(), x)
```

```
def func4():
    x = 1
    return (x + 10)


x = 5
print(func4(), x)
```

- func3: 16 5
  - x is a global variable and then can be used as local variable using global syntax
- func4 : 11 5
  - local variable ‘x’ used

Purwadhika
Startup and Coding School

A function called inside another function

```
def kali(x) :
    if (x < 2) :
        return 1
    else :
        return (x * tiga())

def tiga() :
    return 3

print(kali(5))
```

A function can be called inside another function:

- Here, we define two function: kali and tiga
- Each function has a return
- We call function tiga inside function kali
- We call function kali
- Function kali execute function tiga

Purwadhika
Startup and Coding School

Optional Parameter

- We define a function named print_triangle to print triangle with certain character
- There can be two parameters for print_triangle : n and ch
- n should always be defined
- ch is optional (can be defined or not), if ch is not defined ch will have * as value.

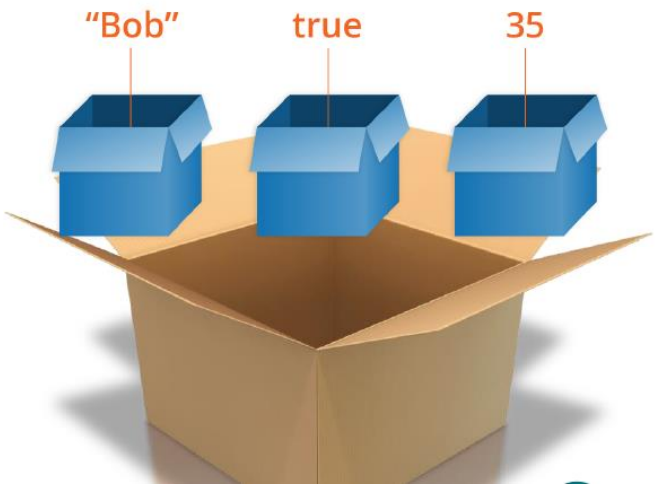
```
def print_triangle(n, ch = '*'):
    for i in range(n):
        for j in range(i+1):
            print(ch, end = '')
        print('')


n = int(input())
print_triangle_1 (n) # ch will be * (default)
print_triangle_1 (n, '#') ch will be #
```

Purwadhika
Startup and Coding School

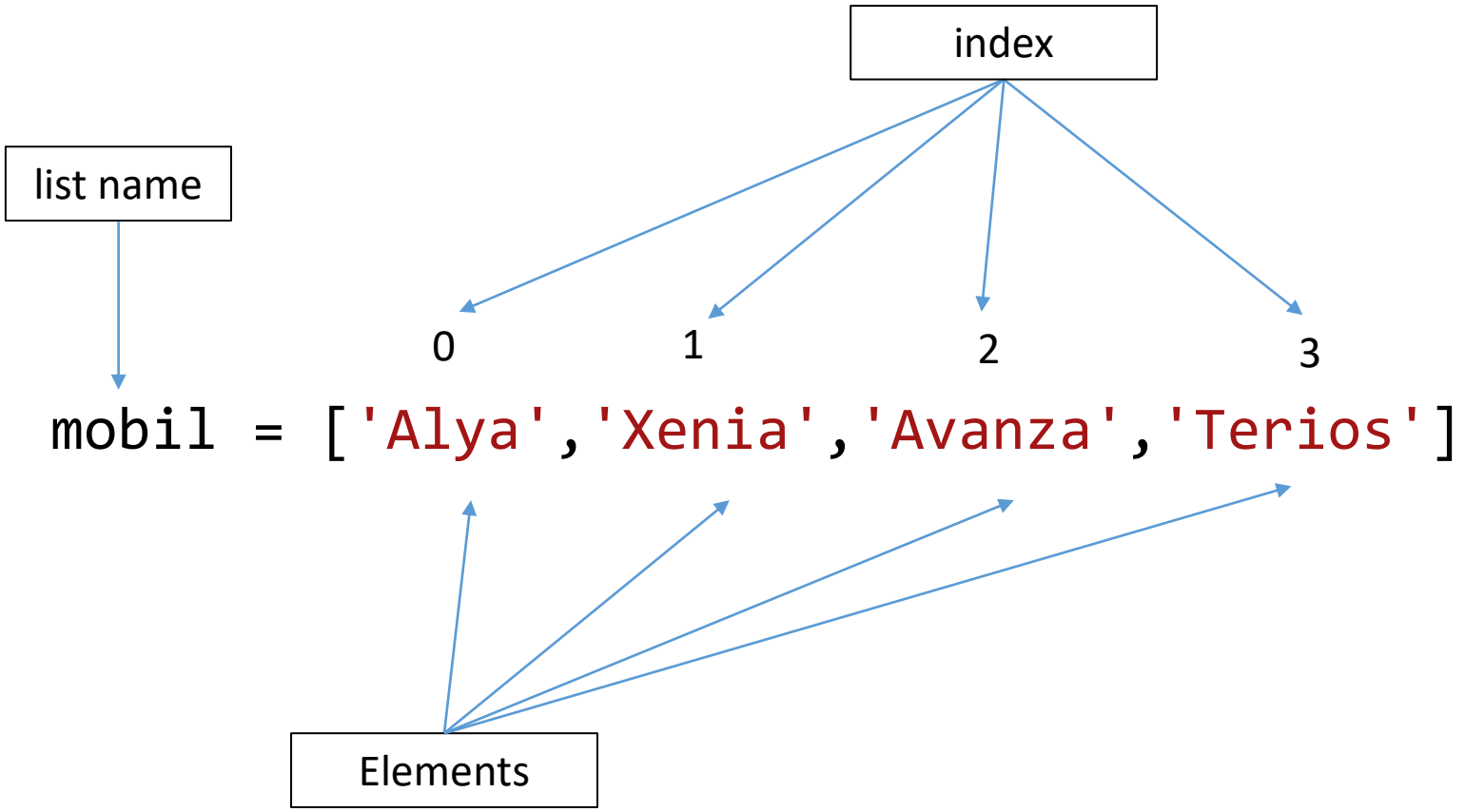
List

- Arrays are container-like values that can hold other data value.
- The data inside an array are called elements.
- Each elements can have different type
- Elements inside list can be changed and can contain some or even all of its elements with the same value



Purwadhika
Startup and Coding School

Structure of A List



Purwadhika
Startup and Coding School

How to Define List

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

mobil_list1 = ['Alya','Xenia','Avanza']
mobil_list2 = [
    'Alya',
    'Xenia',
    'Avanza'
]
mobil_list3 = [mobil1, mobil2, mobil3]

print(mobil_list1)
print(mobil_list2)
print(mobil_list3)
```

List

```
vocab = ['variable', 'Looping', 'Operation', 'logic']
numeric_value = [121, 21, 33, -3, 233, 22]
mixed_list = [23.0, 22, 10/2, [34,32.1,23], 'numbers']
newlist = [vocab, mixed_list]
empty_list = []

print(vocab)
print(numeric_value)
print(mixed_list)
print(newlist)
print(empty_list)
```

Access List Value

- Access using [ ] : `listname[index]`
- to check the number of elements in list : `len(listname)`
- Index start from 0 and then 1 2 3 and so on left to right
- Index can be negative, start from right to left elements with -1 -2, -3, and so on

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

print(len(mobil))
print(mobil[0])
print(mobil[1])
print(mobil[2])
print(mobil[-1])
print(mobil[-2])
```

Purwadhika
Startup and Coding School

Access List Value

```
vocab = ['variable', 'Looping', 'Operation', 'logic']
mixed_list = [23.0, 22, 10/2, [34,32,23], 'numbers']
newlist = [vocab, mixed_list]

print(mixed_list[3])
print(mixed_list[-2])
print(mixed_list[3][0])
print(newlist[0])
print(newlist[1])
print(newlist[0][0])
```

Access List Value using slice

- Access using [ ] : `listname[start:end+1]`
- For example :
  - `listname[1:4]` : Will access list elements with index 1 2 and 3
  - `listname[:4]` : Will access list elements with index 0 1 2 and 3
  - `listname[1:]` : Will access list elements with index 1 until the last element of the list
  - `Listname[:]` : Will access all elements

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

print(buah[1:])
print(buah[:3])
print(buah[2:4])
print(buah[:])
```

Purwadhika
Startup and Coding School

Change Elements inside list

- List are mutable
- Elements inside list can be changed based on needs.

```
buah = ['Jeruk', 'Nanas', 'Apel', 'Mangga']
buah[1] = 'Kelapa'
buah[2] = 'Belimbing'
print(buah)
```

```
vocab = [
      'variable', 'Looping', 'Operation‘,
      'logic', 'control'
]
vocab[1:3] = ['expression', 'list']
print(vocab)
```

Purwadhika
Startup and Coding School

Change Elements inside list

```
alist = ['a', 'b', 'c', 'd']

alist[1:3] = ['x', 'y']
print(alist)
alist[1:3] = []
print(alist)
alist[1:1] = ['b', 'b']
print(alist)
alist[4:4] = ['e']
print(alist)
```

Change Elements inside list

Look at the output of this program closely

```
buah = ['Jeruk', 'Nanas', 'Apel', 'Mangga']
buah2 = buah
buah2[1] = 'Kelapa'

print(buah)
print(buah2)
```

Add and Remove List Elements

- Elements inside list can be added or can be removed.

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

# add from the right
buah.append('Kelapa')
print(buah)

# remove from the right
buah.pop()
buah.pop()
print(buah)
```

Purwadhika
Startup and Coding School

List Deletion

- Elements inside list can be removed using del syntax.

```
a = ['one', 'two', 'three', 'four']


del a[1]
print(a[1])
```

```
alist = ['a', 'b', 'c', 'd', 'e', 'f']
del alist[1:5]
print(alist)
```

Purwadhika
Startup and Coding School

Concatenation and Repetition

- + operator : concatenation
- * operator : repeat list

```
vocab = ['variable', 'Looping', 'Operation', 'logic']
numeric_value = [121, 21, 33, -3, 233, 22]
newlist = [12, 11, 33] + ['Looping', 'Operation']
newlist2 = vocab + numeric_value

print([32,22]+[3,34])
print(newlist)
print(newlist2)
```

```
alist = [1,2,3,4]

print(alist*4)
print(alist*4)
print(alist*0)
print(alist*-1)
```

Purwadhika
Startup and Coding School

List for Loops

- Elements inside list can be accessed using loop one by one

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

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

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

for pos in range(len(buah)) :
    print(item[pos])
```

List Comprehension

- We want to transform each element inside list
- e.g. : each elements times 2, each elements times itself, etc.

```
alist = [i for i in range(5)]
print(alist)

mylist = [1,2,3,4,5]
yourlist = [item ** 2 for item in mylist]
print(yourlist)

alist = [4,2,8,6,5]
blist = [num*2 for num in alist if num%2==1]
print(blist)
```

Purwadhika
Startup and Coding School

Solve It! #1

Buatlah algoritma untuk mengurutkan elemen array berikut:

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

Purwadhika
Startup and Coding School

Solve It! #2

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

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

Purwadhika
Startup and Coding School