

# AI Mastery Course

## Module 1


Introduction to everything

### Section

- Python Functions, string manipulation & file handling

# Learning Objectives

Di akhir modul ini, kita akan dapat:

- Menggunakan built-in functions yang populer di Python, juga membuat user-defined functions
  - Memahami generators, menggunakan map dan filter functions
  - Memahami string manipulation
  - Memahami bagaimana menggunakan reading, writing dan appending data ke sebuah file di python
- 



# Agenda

01

## FUNCTIONS

- Python functions
- User defined functions
- Built-in functions

02

## ITERATORS

- Generators
- Map and filter
- Lambda functions

03

## STRINGS

- String manipulation in python

04

## FILE IO

- File Handling with python

05

## KESIMPULAN

- Kuis
- Ringkasan



01

## FUNCTIONS

- Python functions
- User defined functions
- Built-in functions

# Functions di Python



## User defined functions

Python memungkinkan pengguna untuk membuat fungsi yang ditentukan pengguna



## Built-in functions

Fungsi Python yang sudah tersedia sejak melakukan instalasi Python tanpa harus membuat fungsi lagi



## Lambda functions

Fungsi Lambda, juga disebut fungsi anonim atau tanpa nama, biasanya digunakan untuk mendefinisikan logika sederhana

# User defined functions in python

Fungsi adalah metode utama untuk menata kode dan dapat digunakan kembali dalam Python.

Blok pernyataan yang menerima beberapa argumen, melakukan beberapa fungsi, dan menyediakan output

Keyword def digunakan untuk membuat fungsi yang ditentukan pengguna dengan python

Hasil dari fungsi dinyatakan dengan pernyataan return

## Syntax

```
def function_name(arg1, arg2, ..., argN):  
    <statements>  
    return
```

# User defined functions in python

def keyword  
untuk fungsi

Nama fungsi

Input arguments

- Fungsi bisa menerima satu atau lebih  
input argument

In [1]:

```
1 def my_function(num1,num2):  
2     """ This function can be used to add two numbers  
3     num1: int / float / complex  
4     num2: int / float / complex  
5     """  
6     output = num1 + num2  
7     return output
```

Function  
documentation  
string

Value yang  
dikembalikan  
oleh fungsi

In [2]:

```
1 my_function(num1 = 2, num2 = 5)
```

Out[2]: 7

Calling  
functions

# User defined functions in python

Argument  
tanpa default  
value

Argument dengan default  
value  
– menjadikannya optional  
input argument

```
In [1]: 1 def my_function(num1, num2=5):  
2       output = num1 + num2  
3       return output
```

In [2]: 1 my\_function(5,9)

Out[2]: 14

In [3]: 1 my\_function(6)

Out[3]: 11

user bebas untuk  
mengubah argument  
ke 2 dari nilai  
defaultnya

Jika fungsi ini dipanggil  
dengan 1 argumen,  
nilai default 5 akan  
digunakan untuk  
argument ke 2



# User defined functions in python

Global  
Variable  
Local  
Variable

Local variable dari  
function tidak bisa  
diakses dari luar  
fungsi

```
In [1]: 1 x = 5
        2 def my_function(num1,num2):
        3     y = 2
        4     print(f"value of x is {x}")
        5     print(f"value of y is {y}")
        6     num3 = num1 + num2 + x
        7     return num3
```

Global variable can  
be accessed inside a  
function

```
In [2]: 1 my_function(4,3)

value of x is 5
value of y is 2
```

Out[2]: 12

```
In [3]: 1 print(y)
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-3-d9183e048de3> in <module>
----> 1 print(y)

NameError: name 'y' is not defined
```

# Built-in functions in python

## Standard

- `print()` - Mencetak objek ke text stream file, dipisahkan oleh `sep` dan diikuti oleh akhir
- `len()` - Mengembalikan panjang (jumlah item) dari suatu objek.
- `type()` - Dengan satu argumen, kembalikan tipe objek.
- `id()` - Mengembalikan "identitas" suatu objek.
- `input()` - Jika argumen prompt hadir, itu ditulis ke output standar tanpa tambahan baris baru.
- `help()` - Panggil sistem bantuan bawaan.

## Operasi Matematika

- `min()` - Mengembalikan item terkecil dalam iterable atau terkecil dari dua atau lebih argumen
- `max()` - Mengembalikan item terbesar dalam iterable atau terbesar dari dua atau lebih argumen.
- `sum()` - Menjumlahkan awal dan item dari iterable dari kiri ke kanan dan mengembalikan total.
- `pow()` - Kembalikan *base* ke power *exp*
- `round()` - Mengembalikan angka yang dibulatkan ke presisi ndigit setelah titik desimal
- `abs()` - Mengembalikan nilai absolut dari suatu angka.

# Built-in functions in python

## Type Conversion

- Primitive type conversion: `int()`, `float()`, `bool()`, `str()`, `complex()`
- Non-primitive type conversion: `list()`, `tuple()`, `dict()`, `set()`
- Metode conversion lainnya: `ascii()`, `bin()`, `bytes()`, `hex()`, `chr()`, `bytearray()`, `ord()`

## Utility based functions

- Sequence based functions: `zip()`, `range()`, `map()`, `filter()`, `reversed()`, `enumerate()`, `iter()`, `sorted()`, `next()`
- OOPs related functions: `hasattr()`, `getattr()`, `setattr()`, `issubset()`, `isinstance()`, `object()`, `classmethod()`, `staticmethod()`, `super()`, `dir()`,

# Built-in functions in python

```
1 myList = [8,7,6,2,4,9,3]
2
3 print("mylist is ",mylist)
4
5 print("type of mylist is ",type(mylist))
6
7 print("length of mylist is ",len(mylist))
```

```
mylist is [8, 7, 6, 2, 4, 9, 3]
type of mylist is <class 'list'>
length of mylist is 7
```

```
1 print("sum of all values ",sum(mylist))
2
3 print("minimum value ",min(mylist))
4
5 print("Maximum value ",max(mylist))
6
7 print("Sorted list ",sorted(mylist))
```

```
sum of all values 39
minimum value 2
Maximum value 9
Sorted list [2, 3, 4, 6, 7, 8, 9]
```



02

## ITERATORS

- Generators
- Map and filter
- Lambda functions

# Generators

Generator adalah jenis fungsi khusus yang tidak mengembalikan nilai tunggal, melainkan mengembalikan objek iterator dengan urutan nilai. Dalam fungsi generator, yield statement biasanya digunakan daripada return statement.

## Syntax

```
def generator_function(arg1, arg2, ...,  
argN) :  
    <statements>  
    yield value1  
    yield value2
```

Mengakses  
data ukuran  
kecil



Menghitung  
data ukuran  
kecil



Hemat  
memori

# Generators

```
1  # A Python program to demonstrate use of
2  # generator object with next()
3
4  # A generator function
5  def simpleGeneratorFun():
6      yield 1
7      yield 2
8      yield 3
9
10 # x is a generator object
11 x = simpleGeneratorFun()
12 print("Type of x - ",type(x))
13
14 # Iterating over the generator object using next
15 print(next(x)) # In other option is x.__next__()
16 print(next(x))
17 print(next(x))
```

Type of x - <class 'generator'>

1  
2  
3

# Generators – communicating with generators



`next()`

Fungsi next dapat meminta pada generator sebuah nilai berikutnya.



`stopIteration`

Built-in exception yang digunakan untuk exit dari sebuah Generator.



`Send`

Mengizinkan caller untuk komunikasi dengan generator. (biasanya digunakan untuk coroutines)



# Map

fungsi `map()` adalah fungsi tingkat lanjut, ia menerima fungsi lain dan urutan 'iterables' sebagai parameter dan memberikan output setelah menerapkan fungsi ke setiap iterable dalam urutan.

## Syntax

```
map(function, iterables)
```

# Map

```
1 # a function for data transformation, to be applied on an iterable object
2 def my_function(a):
3     return a*a
4 # iterable object
5 mylist = [2,4,5,9,6,3]
6
7 # Applying map
8 x = map(my_function, mylist) #x is the map object
9
10 print(x)
11 print(type(x))
12 print(list(x))
```

```
<map object at 0x00000245444EB2B0>
<class 'map'>
[4, 16, 25, 81, 36, 9]
```

# Filter

fungsi `filter()` adalah fungsi tingkat lanjut, digunakan untuk menghasilkan daftar keluaran nilai yang mengembalikan nilai `true` saat fungsi dipanggil.

## Syntax

```
filter(function, iterables)
```

# Filter

```
1 # a function for data filtering, to be applied on an iterable object
2 def get_even(a):
3     if a%2==0:
4         return True
5     else:
6         return False
7
8 # iterable object
9 mylist = [2,4,5,9,6,3,8,7]
10
11 # Applying filter
12 x = filter(get_even, mylist) #x is the filter object
13
14 print(x)
15 print(type(x))
16 print(list(x))
```

```
<filter object at 0x0000024544504C10>
<class 'filter'>
[2, 4, 6, 8]
```

# Lambda function

Fungsi Lambda juga disebut sebagai fungsi anonim adalah fungsi yang tidak memiliki nama. Fungsi lambda digunakan untuk merepresentasikan ekspresi pendek untuk waktu yang singkat.



## Syntax

```
lambda arguments: expression
```

# Lambda function

```
1 # Use of Lambda functions
2
3 square = lambda x:x**2
4
5 square(5)
```

25

```
1 # Using lambda functions with map
2
3 mylist = [4,5,3,2,5,8]
4
5 squarelist = map(lambda x:x**2, mylist)
6
7 print(list(squarelist))
```

[16, 25, 9, 4, 25, 64]



03

## STRING

- String manipulation in python

# String Manipulation



Dalam Python, String adalah array byte yang mewakili karakter Unicode.



String baris tunggal dapat dibuat menggunakan tanda kutip tunggal atau ganda. String multiline dapat dibuat menggunakan tanda kutip tiga.



Python menyediakan banyak seperangkat operator, fungsi, dan metode untuk bekerja dengan string.



# String manipulation menggunakan operators

Operator + melakukan operasi matematika dengan Python. Cukup menulis dua literal string bersama-sama dan menggabungkannya.

Operator \* dapat digunakan untuk mengulang string beberapa kali.

Operator "in" dapat digunakan untuk memeriksa apakah substring ada di dalam string atau tidak

```
1 # Python String Operations
2 str1 = 'Hello'
3 str2 = 'World!'
4
5 # using +
6 print('str1 + str2 = ', str1 + str2)
7
8 # using *
9 print('str1 * 3 =', str1 * 3)
```

```
str1 + str2 = HelloWorld!
str1 * 3 = HelloHelloHello
```

```
1 str3 = "Hello world from python"
2
3 # using "in"
4 print("world" in str3)
5
6 # using "not in"
7 print("Hello" not in str3)
```

```
True
False
```

# String Manipulation menggunakan built-in methods

Method	Description
<code>capitalize()</code>	Mengubah karakter pertama menjadi huruf besar
<code>count()</code>	Mengembalikan berapa kali nilai tertentu muncul dalam string
<code>encode()</code>	Mengembalikan versi string yang disandikan
<code>endswith()</code>	Mengembalikan nilai true jika string diakhiri dengan nilai yang ditentukan
<code>find()</code>	Mencari string untuk nilai yang ditentukan dan mengembalikan posisi di mana ia ditemukan
<code>format()</code>	Memformat nilai yang ditentukan dalam string
<code>index()</code>	Mencari string untuk nilai yang ditentukan dan mengembalikan posisi di mana ia ditemukan
<code>join()</code>	Mengubah elemen dari iterable menjadi string
<code>lower()</code>	Mengubah string menjadi huruf kecil
<code>maketrans()</code>	Mengembalikan tabel terjemahan untuk digunakan dalam terjemahan
<code>partition()</code>	Mengembalikan tupel di mana string dipisah menjadi tiga bagian
<code>replace()</code>	Mengembalikan string di mana nilai tertentu diganti dengan nilai tertentu
<code>split()</code>	Membagi string pada pemisah yang ditentukan, dan mengembalikan daftar
<code>splitlines()</code>	Membagi string pada jeda baris dan mengembalikan daftar
<code>startswith()</code>	Mengembalikan nilai true jika string dimulai dengan nilai yang ditentukan
<code>strip()</code>	Mengembalikan versi string yang dipangkas
<code>swapcase()</code>	Tukar kasus, huruf kecil menjadi huruf besar dan sebaliknya
<code>title()</code>	Mengonversi karakter pertama setiap kata menjadi huruf besar
<code>translate()</code>	Mengembalikan string yang diterjemahkan
<code>upper()</code>	Mengubah string menjadi huruf besar
<code>zfill()</code>	Mengisi string dengan jumlah nilai 0 yang ditentukan di awal

# String manipulation menggunakan built-in methods

```
1 # Common string methods
2 x = "hello World from PYTHON"
3
4 print(x.upper()) # converting to upper case
5 print(x.lower()) # converting to lower case
6 print(x.split(" ")) #splitting string to a list of words
7 print("wold found in x ",x.find("World")) # finding a substring
8 print(x.replace("World","Earth")) # replacing a word with other word
```

```
HELLO WORLD FROM PYTHON
hello world from python
['hello', 'World', 'from', 'PYTHON']
wold found in x 6
hello Earth from PYTHON
```

# String Manipulation menggunakan built-in methods

```
1 # Python string format() method
2
3 # default(implicit) order
4 default_order = "{}, {} and {}".format('John','Bill','Sean')
5 print('\n--- Default Order ---\n',default_order)
6
7 # order using positional argument
8 positional_order = "{1}, {0} and {2}".format('John','Bill','Sean')
9 print('\n--- Positional Order ---\n',positional_order)
10
11 # order using keyword argument
12 keyword_order = "{s}, {b} and {j}".format(j='John',b='Bill',s='Sean')
13 print('\n--- Keyword Order ---\n',keyword_order)
```

```
--- Default Order ---
John, Bill and Sean
```

```
--- Positional Order ---
Bill, John and Sean
```

```
--- Keyword Order ---
Sean, Bill and John
```



04

## FILE IO

- File Handling with python

# File handling di Python

- fungsi `open()` menerima dua argumen, nama file dan mode akses di mana file diakses.
- Fungsi mengembalikan objek file yang dapat digunakan untuk melakukan berbagai operasi seperti membaca, menulis, dll.

Access mode	Operation	Position of file pointer in file
r	Read only	Beginning
rb	Read only in binary format	Beginning
r+	Read and write both	Beginning
rb+	Read and write both in binary format	Beginning
w	Write only, overwrite existing file, creates new if no file exist	Beginning
wb	Write only in binary format, overwrite existing file, creates new if no file exist	Beginning
w+	Write and read modes, overwrites existing file, creates new file if not no file exist	Beginning
wb+	Read and write in binary format	Beginning
a	Append mode, creates new file if no file exist	End
ab	Append mode in binary format, creates new file if no file exist	End
a+	Append and read, creates new file if no file exist	End
ab+	Append and read in binary format	End

# Menulis data di file baru

```
1 # open the myfile.txt in write mode.
2 # this will create a new file if no such file exists.
3 file_pointer = open("myfile.txt", "w")
4
5 # writing the content to the file
6 file_pointer.write(''Python is easy, powerful and simple.
7 It is the fastest-growing programing language
8 ''')
9
10 # closing the opened the file
11 file_pointer.close()
```

myfile.txt - Notepad

File Edit Format View Help

**Python is easy, powerful and simple.**  
**It is the fastest-growing programing language**



# Menambahkan data di file

```
1 # open the myfile.txt in append mode.  
2 file_pointer = open("myfile.txt", "a")  
3  
4 # appending the content to the file  
5 file_pointer.write("Python has an easy syntax and user-friendly interaction.")  
6  
7 # closing the opened the file  
8 file_pointer.close()
```

myfile.txt - Notepad

File Edit Format View Help

**Python is easy, powerful and simple.  
It is the fastest-growing programming language  
Python has an easy syntax and user-friendly interaction.**



# Membaca data dari sebuah file

```
1 # open the myfile.txt in read mode.  
2 # it will cause error, if no such file exist  
3 file_pointer = open("myfile.txt", "r")  
4  
5 # reading the content of the file  
6 data = file_pointer.read()  
7 print(type(data))  
8 print(data)  
9  
10 # closing the opened the file  
11 file_pointer.close()
```

<class 'str'>

Python is easy, powerful and simple.

It is the fastest-growing programming language

Python has an easy syntax and user-friendly interaction.



05

## KESIMPULAN

- Kuis
- Ringkasan

# Kuis

## Pertanyaan

**Keyword apa yang digunakan untuk membuat fungsi generator?**

- A. Return
- B. Lambda
- C. Yield
- D. None of above



# Kuis

## Pertanyaan

**Keyword apa yang digunakan untuk membuat fungsi generator?**

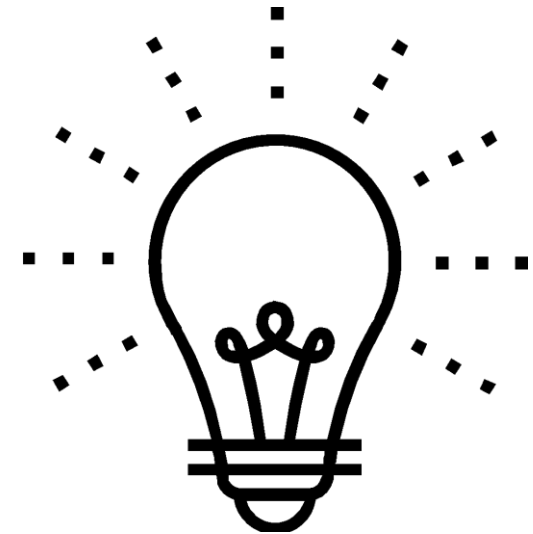
- A. Return
- B. Lambda
- C. Yield
- D. None of above

**Answer - C**



# Ringkasan

- Fungsi adalah metode utama dan paling penting untuk menata kode agar dapat digunakan kembali di Python
- Python Generator membantu dalam menghemat memori dengan melakukan pemrosesan data secara efisien termasuk dalam melakukan perhitungan.
- Fungsi map dan filter dapat digunakan untuk melakukan transformasi dan pemfilteran data pada objek yang dapat diubah
- Python mendukung penggunaan operator seperti +, \*, "in" dan "not in" untuk memanipulasi string, selain itu string python memiliki banyak metode bawaan yang dapat digunakan untuk memanipulasi string
- Fungsi Python open() dapat digunakan untuk menangani file, membantu dalam membaca, menulis, menambahkan data ke file





# TERIMA KASIH

## Orbit Future Academy

PT Orbit Ventura Indonesia  
Center of Excellence (Jakarta Selatan)  
Gedung Veteran RI, Lt.15  
Unit Z15-002, Plaza Semanggi  
Jl. Jenderal Sudirman Kav.50, Jakarta  
12930, Indonesia

- 📁 Jakarta Selatan/Pusat
- 📁 Jakarta Barat/BSD
- 📁 Kota Bandung
- 📁 Kab. Bandung
- 📁 Jawa Barat

## Hubungi Kami

Director of Sales & Partnership  
[ira@orbitventura.com](mailto:ira@orbitventura.com)  
+62 858-9187-7388

## Social Media



Orbit Future Academy



OrbitFutureAcademy



@OrbitFutureAcademyIn1



Orbit Future Academy