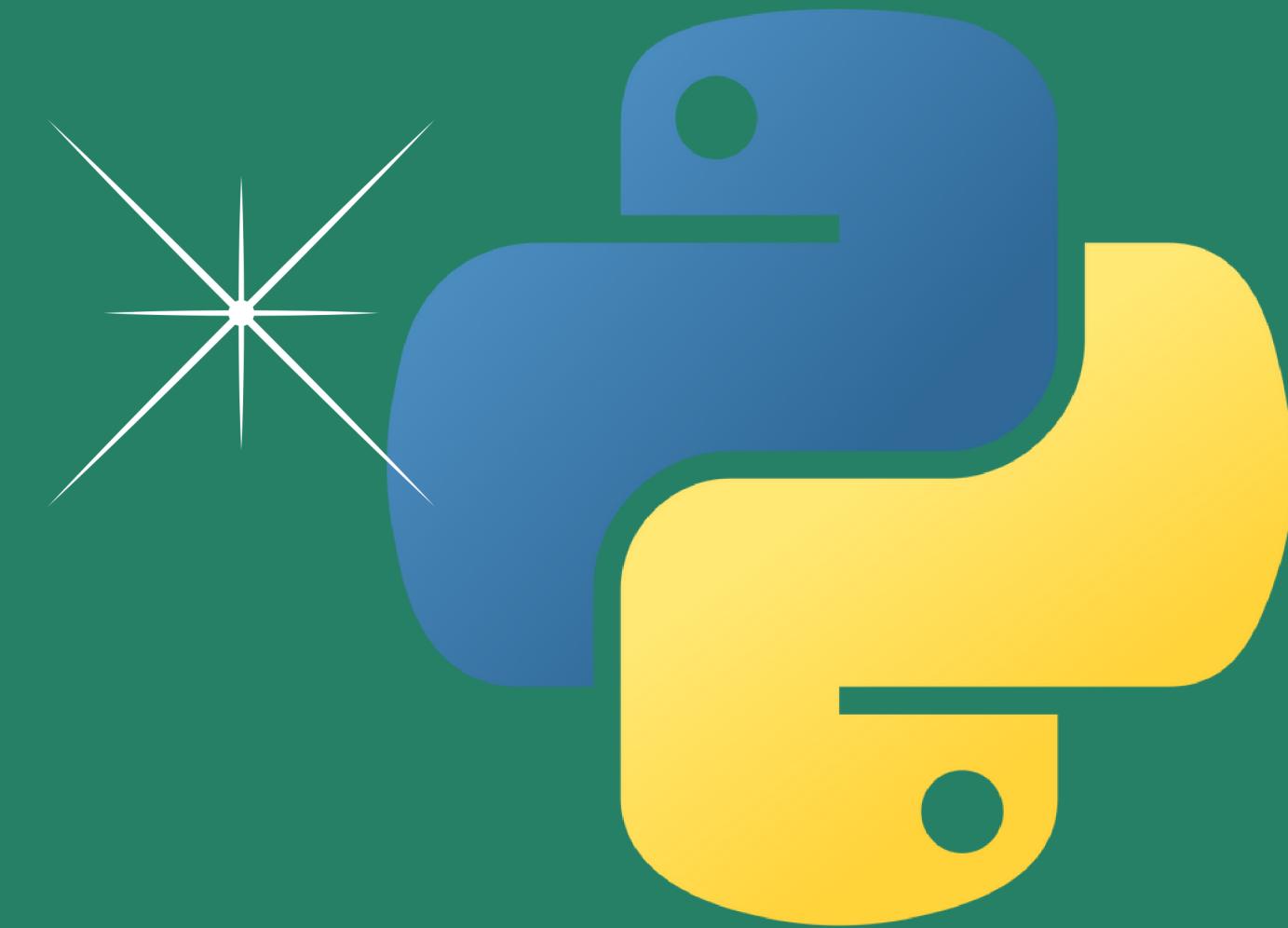




KAMPUS
MERDEKA X
MYEDUSOLVE

PYTHON → WEEK #6

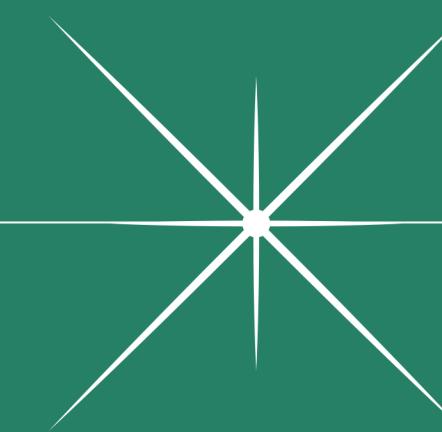
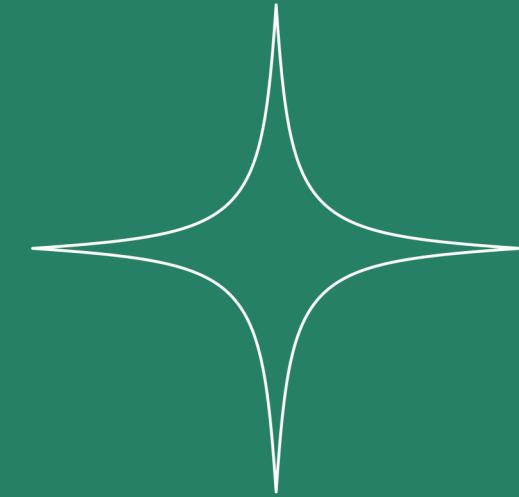


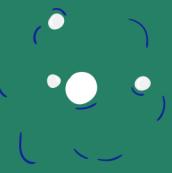
Team 3 - Data Science A



INTRODUCING OUR TEAM

- MUHAMMAD REZA ABDILLAH
- AUDRIC LYSANDER
- ALEX MARIO SIMANJUNTAK
- YEMIMA SIPAYUNG
- LATIFAH SINTA NUGRAHENI





OUTLINE



CONTROL FLOW



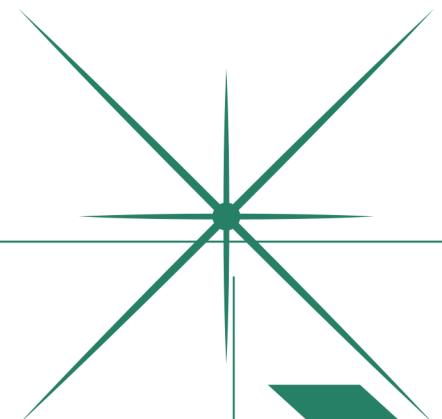
BRANCHING CONDITIONAL STATEMENT



LOOP



FILE HANDLING



CODE DOCUMENTATION AND STRUCTURE



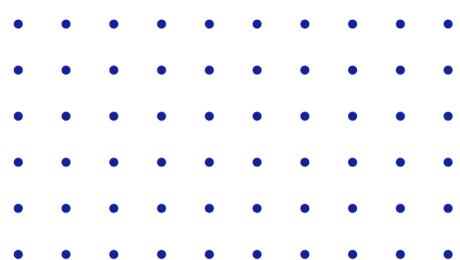
TYPE VARIABLES IN PYTHON



FUNCTION



MODULE

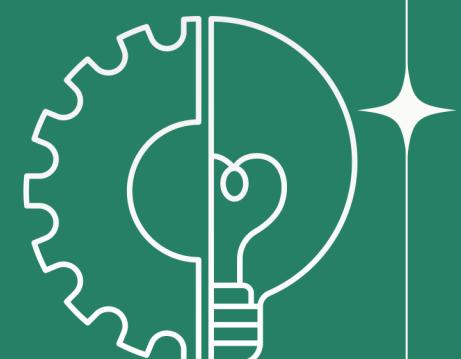


BRANCHING CONDITIONAL STATEMENT

Kondisi **IF**, digunakan untuk mengantisipasi kondisi yang terjadi saat jalanya program dan menentukan tindakan apa yang akan diambil sesuai dengan statement yang kita set.

Pada python ada beberapa statement diantaranya adalah **if**, **else** dan **elif**.

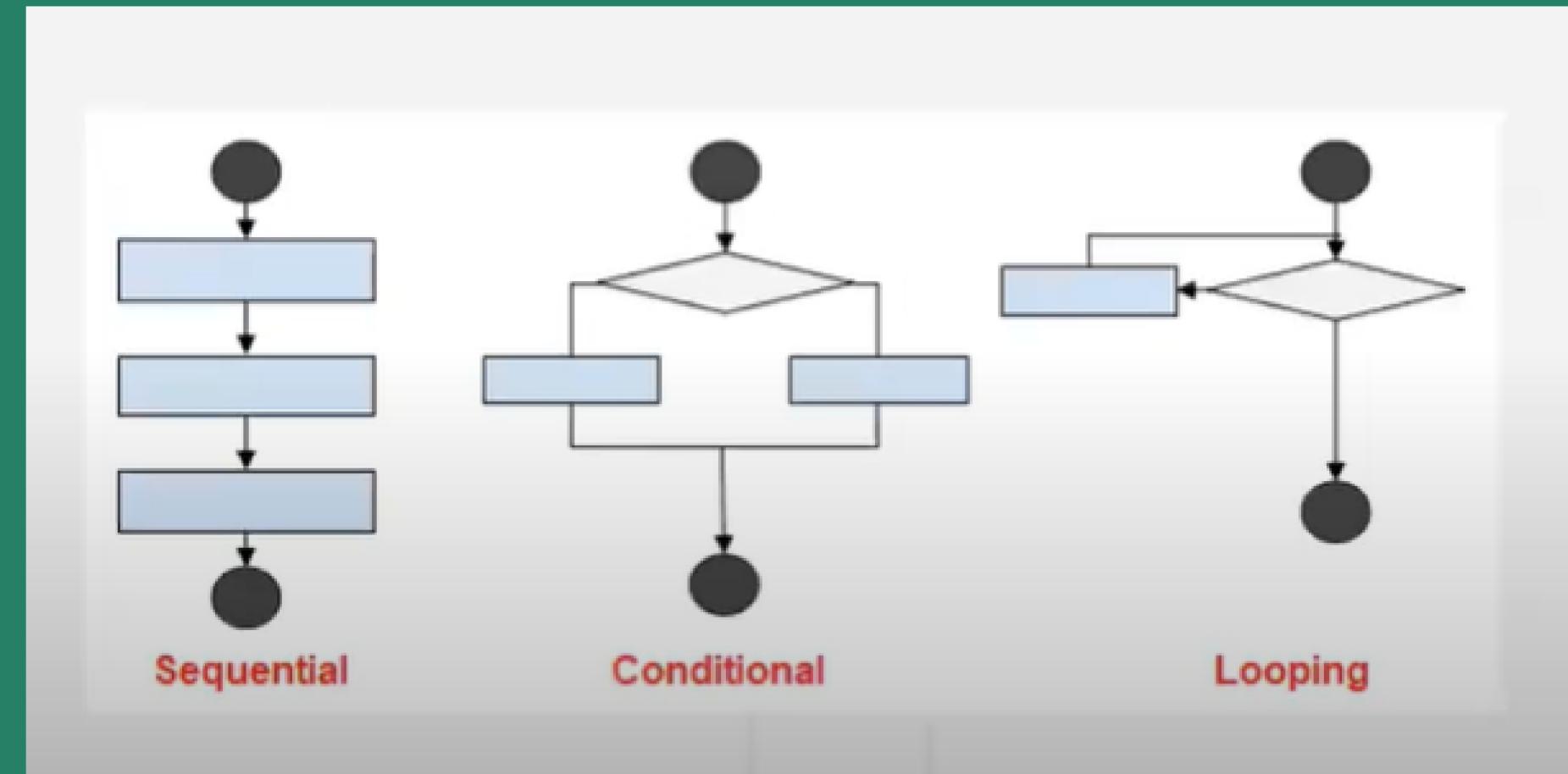
IF digunakan untuk mengeksekusi code jika kondisi bernilai benar, jika kondisi bernilai salah False maka statement tidak akan di-eksekusi.



CONTROL FLOW



Control Flow (Flow of Control) adalah suatu conditional statement yang berguna untuk melakukan perintah yang mengontrol alur dari sebuah program berdasarkan kondisi tertentu ketika program tersebut berjalan.



LOOP



Loop dalam pemrograman adalah code atau instruksi yang dieksekusi secara berulang-ulang (di-iterasi) sampai suatu kondisi tertentu pada tiap sequence terpenuhi.

Loop berbeda dengan **branching**, perbedaannya loop akan melakukan iterasi berulang-ulang selama kondisi tertentu masih terpenuhi, sedangkan branching hanya sekali saja.

LOOP ADA 3 BAGIAN PENTING

SEQUENCE

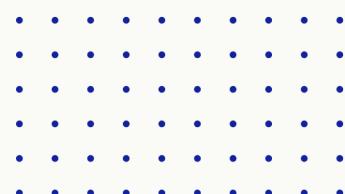
nilai yang bersifat iterable

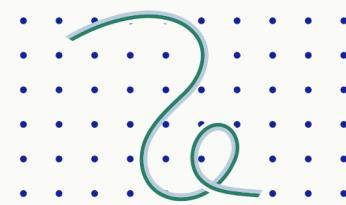
VALUES

setiap item yang iterasi dari sequence

CODE LOCK

statement tertentu yang dieksekusi secara berulang





LOOP

■ FOR LOOP

for loop adalah loop yang sering digunakan untuk iterasi sebuah data secara berulang-ulang. Untuk mengetahui urutan iterasi pada list dapat menggunakan fungsi `enumerate()`.



```
# contoh buat list baru
list1 = ["bambang", "pamungkas", "firman", "utina", "budi", "sudarsono"]
list2 = [1,2,4,5,6,7,8,9,10]
list3 = [1,3,4,"string", True]

# contoh for loop, i adalah "alias" untuk iterasi item
for i in list3:
    print(i)
```

```
# contoh for Loop dengan operasi
for num in list2:
    print("mulai putaran")
    print(num, "dibagi 2 adalah", num/2)
    print(num, "dikali 2 adalah", num*2)
    print("end putaran \n")
```

```
# contoh for Loop dengan enumerate
for i,v in enumerate(list1):
    print(f"index : {i} || values : {v}")
```





LOOP

■ WHILE LOOP

while loop adalah loop yang akan mengulang suatu statement berkali-kali selama kondisi true terpenuhi.

```
# contoh while Loop harus tentukan batasnya dulu, jika tidak maka akan diulang tanpa batas
n=0 # set initial number
while(n < 9): #9 batasnya
    print("Loop ke:", n)
    n = n+1 # 1 increament
print("Loop Selesai")
```

■ BREAK

mengakhiri loop saat ini bahkan jika kondisi while benar dan melanjutkan eksekusi pada pernyataan berikutnya.

■ PASS

Untuk loop tidak boleh kosong, tetapi jika Anda karena alasan tertentu memiliki loop for tanpa konten, masukkan pernyataan pass untuk menghindari kesalahan.

■ CONTINUE

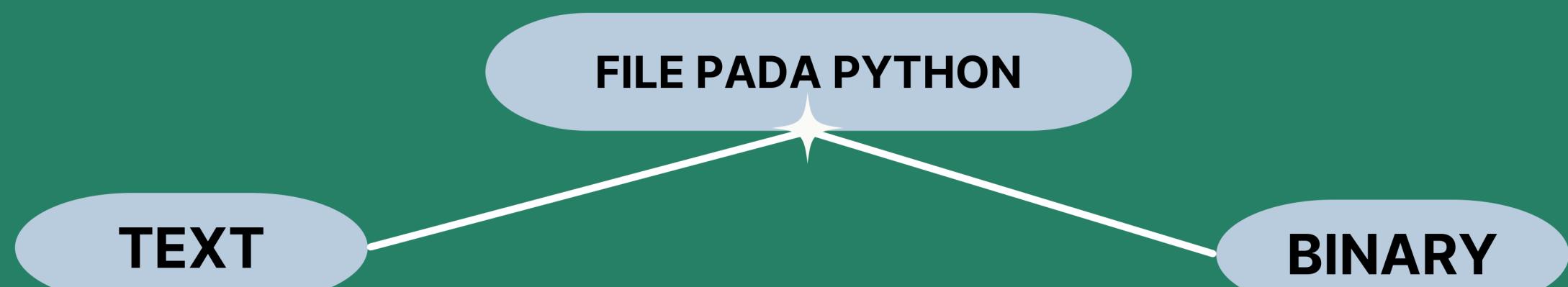
hentikan iterasi saat ini dan lanjutkan dengan iterasi berikutnya.

■ NESTED LOOPS

loop di dalam loop. "Loop dalam" akan dieksekusi satu kali untuk setiap iterasi "loop luar"

FILE HANDLING

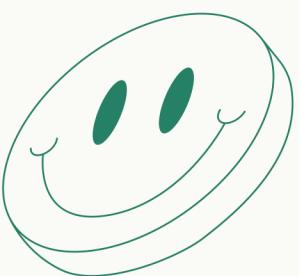
Membaca dan menulis file adalah teknik dasar yang harus dipahami dalam pemrograman Python, karena banyak digunakan untuk pengolahan dan melakukan processing terhadap file.



File yang berisi teks. Setiap baris teks memiliki EOL (End of Line).
Contoh: TXT, MD, CSV, JSON, dsb.

File yang bukan teks, hanya bisa diproses oleh program tertentu yang memahami strukturnya.
Contoh: EXE, JPG, MKV, M4A, 3GP, dsb.





MEMBUKA, MEMBACA DAN MENULIS FILE



Untuk membuka file, Python sudah menyediakan fungsi **open()**, yang memiliki dua parameter yaitu nama file dan mode. Contoh : `file = open("/path/file.txt", mode = "r")`

Mode	Keterangan
"r"	hanya baca saja
"w"	akses untuk menulis file, jika file sudah ada, maka file akan di-replace dan diganti dengan yang baru ditulis
"a"	digunakan untuk append atau menambah data ke file, artinya jika sudah ada data dalam file, maka akan ditambahkan dan tidak di-replace
"r+"	digunakan untuk membaca sekaligus menulis data ke file

INPUT DAN OUTPUT OPERATIONS

■ OPEN FILE

untuk membuka file, ia mengembalikan sesuatu yang disebut objek file.

```
my_file = open(filename, filemode)
```

■ CLOSE FILE

dalam beberapa kasus, karena buffering, perubahan yang dibuat pada file mungkin tidak ditampilkan hingga Anda menutup file.

```
f = open("demofile.txt" , "r")
print(f.read())
f.close
```

INPUT DAN OUTPUT OPERATIONS



READ AND WRITE

Operation	Syntax	Description
Read Only	r	Open text file for reading only.
Read and Write	r+	Open the file for reading and writing.
Write Only	w	Open the file for writing.
Write and Read	w+	Open the file for reading and writing. Unlike "r+" it doesn't raise an I/O error if file doesn't exist.
Append Only	a	Open the file for writing and creates new file if it doesn't exist. All additions are made at the end of the file and no existing data can be modified.
Append and Read	a+	Open the file for reading and writing and creates new file if it doesn't exist. All additions are made at the end of the file and no existing data can be modified.

CODE DOCUMENTATION AND STRUCTURE

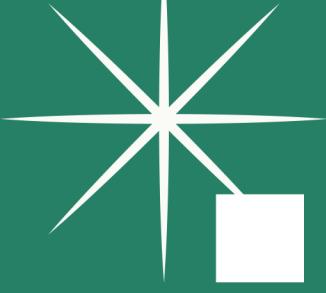
DOCUMENT CODE SEGMENT

- Identiation
- Comment
- Document String

FUNCTION DEFINITION

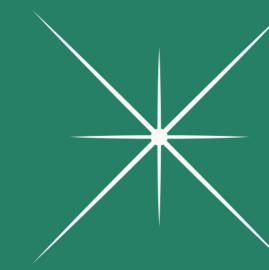
- Call Signature
- Default Value
- Return Function Value
- Define Function





■ Indentation

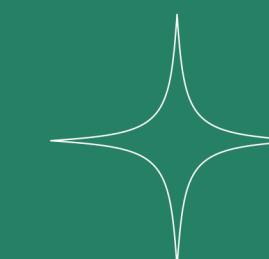
Indentasi mengacu pada spasi di awal baris kode. Di mana dalam bahasa pemrograman lain lekukan dalam kode hanya untuk keterbacaan, lekukan dalam Python sangat penting.



```
if 5 > 2:  
    print("Five is greater than two!")
```

■ Comment

Komentar digunakan selama debugging untuk mengidentifikasi masalah dan menjelaskan kode. Komentar dimulai dengan karakter hash (#). Komentar dapat digunakan untuk menjelaskan kode Python, membuat kode lebih mudah dibaca, dan mencegah eksekusi saat menguji kode.

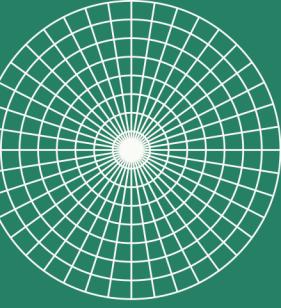
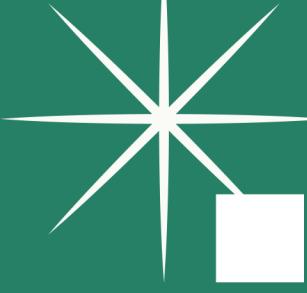


```
# ini adalah komentar  
print("Hello, World!")
```

■ Document String

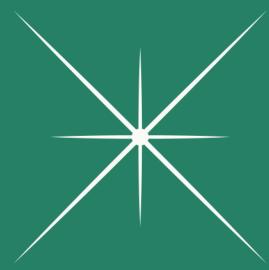
menyediakan cara yang nyaman untuk mengaitkan dokumentasi dengan modul, fungsi, kelas, dan metode Python.

```
def my_function():  
    """Demonstrates triple double quotes docstrings and does nothing really."  
    return None  
    print("Using doc_:")  
    print(my_function.__doc__)  
    print("Using help:")  
    help(my_function)
```



■ Call Signature

Dibutuhkan callable sebagai parameter dan mengembalikan anotasi. Ini menimbulkan kesalahan nilai jika tidak ada tanda tangan yang diberikan. Jika objek tipe tidak valid diberikan maka itu melempar Kesalahan Tipe.



■ Default Value

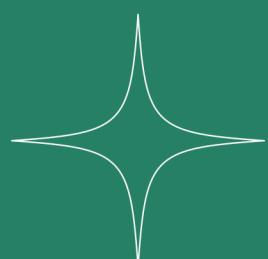
Untuk menentukan nilai default untuk parameter, Anda menggunakan yang berikut:

```
def function_name(param1, param2=value2, param3=value, ...):
```

```
def greet(name, message='Hi'):  
    return f'{message}{name}'  
greeting = greet('John', 'Hello')  
print(greeting)
```

■ Define Function

suatu fungsi dengan kata kunci def, lalu menulis pengenal fungsi (nama) diikuti dengan tanda kurung dan titik dua.



```
def my_function():  
    print("Hello from a function")
```

TYPE VARIABLES IN PYTHON

Global Variable

Variabel yang dapat diakses dari semua fungsi.

Local Variable

Variabel yang hanya dapat diakses di dalam fungsi yang ada di dalamnya.

Built in Variables

Variabel yang sudah secara default ada di dalam python.



TYPE VARIABLES IN PYTHON

Dalam python, urutan mengakses variabel dikenal sebagai L.GB (Lokal, Global, Built-in). Jadi program python mulai mencari variabel lokal terlebih dahulu, jika variabel ada, maka itu yang digunakan, jika tidak ada, maka python akan melanjutkan pencarian hingga ke global.



FUNCTION

Fungsi adalah kumpulan perintah kode yang kita kelompokkan bersama sehingga kita dapat mengingatnya berkali-kali.

Dengan fungsi, kita dapat memecah program besar yang kita verifikasi, menjadi bagian-bagian kecil dengan tugas masing-masing Penulisan fungsi dimulai dengan kata kunci titik, kemudian diikuti dengan isi parameter, ditandai dengan tanda kurung () dan titik dua

MODULE

MODULS

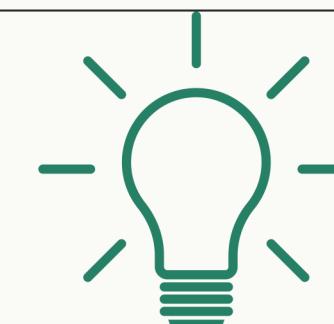
Modul adalah file yang berisi kode untuk fungsi, kelas, dan variabel yang disimpan dalam satu file dengan ekstensi py dan dapat diimpor berulang kali

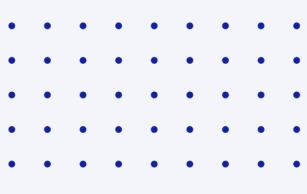
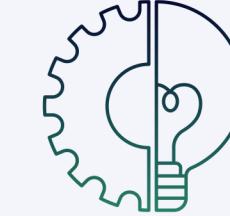
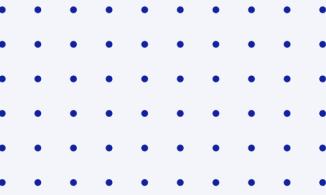
RANDOM MODULS

Modul Python bawaan yang digunakan untuk menghasilkan angka acak. Modul ini dapat digunakan untuk melakukan tindakan acak seperti menghasilkan angka acak, mencetak nilai acak.

MATH MODULS

Modul ini menyajikan fungsi matematika yang umum diperlukan serta bilangan Buge dari fungsi matematika lainnya





Thank You