

STMIK YMI TEGAL



Tugas Algoritma Dan Pemrograman 1

Galih Anggoro Prasetya

22205018

Teknik Informatika

1. Mengenai type hint python?

Jawab :

Ini berarti kita perlu mendeklarasikan tipe variabel, parameter, dan nilai kembalian fungsi di awal. Jenis yang telah ditentukan memungkinkan kompiler untuk memeriksa kode sebelum mengkompilasi dan menjalankan program.

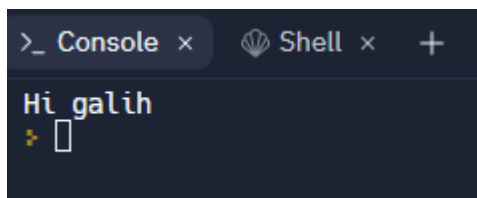
Python menggunakan pengetikan dinamis, di mana variabel , parameter, dan nilai kembalian suatu fungsi dapat berupa tipe apa saja. Juga, jenis variabel dapat berubah saat program berjalan.

Umumnya, pengetikan dinamis memudahkan untuk memprogram dan menyebabkan kesalahan tak terduga yang hanya dapat Anda temukan hingga program berjalan.

Penggunaan type hint python :

```
def say_hi(name):  
    return f'Hi {name}'  
  
greeting = say_hi('John')  
print(greeting)
```

Hasil :



The screenshot shows a terminal window with two tabs: 'Console' and 'Shell'. The 'Console' tab is active, and it displays the output 'Hi_galih' on the first line, followed by a prompt character '➤' and a cursor on the second line.

2. Mengenai args python?

Jawab :

Sintaks khusus `*args` dalam definisi fungsi dalam python digunakan untuk meneruskan sejumlah variabel argumen ke suatu fungsi. Ini digunakan untuk melewati daftar argumen panjang variabel tanpa kata kunci.

- Sintaksnya adalah menggunakan simbol `*` untuk menerima sejumlah variabel argumen; menurut konvensi, sering digunakan dengan kata `args`.
- Apa yang `*args` memungkinkan Anda lakukan adalah mengambil lebih banyak argumen daripada jumlah argumen formal yang Anda tetapkan sebelumnya. Dengan `*args`, sejumlah argumen tambahan dapat ditempelkan ke parameter formal Anda saat ini (termasuk nol argumen tambahan)
- Misalnya, kita ingin membuat fungsi perkalian yang mengambil sejumlah argumen dan dapat mengalikan semuanya. Itu dapat dilakukan dengan menggunakan `*args`.
- Menggunakan `*`, variabel yang kita kaitkan dengan `*` menjadi arti yang dapat diubah, Anda dapat melakukan hal-hal seperti mengulangnya, menjalankan beberapa fungsi tingkat tinggi seperti peta dan filter, dll.

Penggunaan args python :

```
def myFun(*argv):  
    for arg in argv:  
        print(arg)  
  
myFun('Hello', 'Welcome', 'to', 'GeeksforGeeks')
```

Hasil :

```
Hello  
Welcome  
to  
GeeksforGeeks
```

3. Mengenai kwargs python?

Jawab :

Sintaks khusus `**kwargs` dalam definisi fungsi dalam python digunakan untuk meneruskan daftar argumen panjang variabel dengan kata kunci. Kami menggunakan nama `kwargs` dengan bintang ganda. Alasannya adalah bahwa bintang ganda memungkinkan kita untuk melewati argumen kata kunci (dan berapa pun jumlahnya).

- Argumen kata kunci adalah tempat Anda memberikan nama ke variabel saat Anda meneruskannya ke dalam fungsi.
- Orang dapat menganggap `kwargs` sebagai kamus yang memetakan setiap kata kunci ke nilai yang kita lewati di sampingnya. Itulah sebabnya ketika kami mengulangi `kwargs` tampaknya tidak ada urutan di mana mereka dicetak.

Penggunaan `kwargs` python :

```
def myFun(**kwargs):  
    for key, value in kwargs.items():  
        print("%s == %s" % (key, value))  
  
# Driver code  
myFun(first='Geeks', mid='for', last='Geeks')
```

Hasil :

```
first == Geeks  
mid == for  
last == Geeks
```

4. Mengenai lambda python?

Jawab :

Lambda Python adalah kata kunci yang digunakan untuk menangani fungsi anonim atau fungsi yang tidak memiliki nama. Fungsi yang tidak memiliki nama (anonym) adalah fungsi kecil dan terbatas yang memiliki tidak lebih dari satu baris. Sama seperti fungsi normal, fungsi Lambda dapat memiliki banyak argumen dengan satu ekspresi.

Penggunaan lambda python :

```
# perbedaan antara def() dengan lambda().  
def kubus(i):  
    return i*i*i
```

```
lambda_kubus = lambda i: i*i*i
```

```
# pemanggilan def() dan lambda()  
# mencetak hasil pada def()  
print(kubus(10))
```

```
# mencetak hasil pada lambda()  
print(lambda_kubus(10))
```

Output:

```
1000  
1000
```


5. OOP pada python ?

Jawab :

Konsep OOP merupakan penulisan pada pemrograman berorientasi objek yang di mana ketika kita menulis kode bergaya OOP kita harus menggunakan class terlebih dahulu. Ibarat class adalah cetakan yang di mana kita harus di inisialisasi dulu.

Dan jika kita menjalankan sebuah kode OOP kita harus menggunakan method yang di mana method ini adalah function pada pemrograman OOP. Terakhir dengan kita menggunakan dan belajar OOP sangat wajib sekali kalo kita menggunakan teknologi framework nantinya.

Penggunaan OOP python :

```
1 ▼ class Mahasiswa:
2 ▼     def __init__(self):
3         print("===DATA MAHASISWA===")
4
5
6 ▼     def dataMahasiswa(self):
7         print("""
8 Nama saya adalah : galih anggoro prasetya
9 Umur saya : 18 tahun
10 Jurusan : Teknik Informatika
11         """)
12
13 ▼ if __name__ == '__main__':
14     RUN = Mahasiswa()
15     RUN.dataMahasiswa()
```

Hasil :

```
===DATA MAHASISWA===  
  
Nama saya adalah : galih anggoro prasetya  
Umur saya : 18 tahun  
Jurusan : Teknik Informatika  
  
❖ □
```

Selesai