

Lab 3.4.6

Lab – Python Classes Review

Tujuan :

Part 1 : Jalankan DEVASC VM

Part 2 : Review Function, Method dan Class

Part 3 : Membuat Function

Part 4 : Membuat class dengan method

Part 5 : Review script class circleClass.py

Resource :

- 1 PC
- Virtual Box / VM Ware
- DEVASC Virtual Machine

Langkah-langkah :

Part 1 : Jalankan DEVASC VM

Part 2 : Review Function, Methods, dan Class

Langkah 1 : Function

Fungsi adalah bagian dari program yang dapat digunakan ulang. Hal ini bisa dicapai dengan memberi nama pada blok statemen, kemudian nama ini dapat dipanggil di manapun dalam program.

Fungsi dalam Python didefinisikan menggunakan kata kunci def. Setelah def ada nama pengenalan fungsi diikuti dengan parameter yang diapit oleh tanda kurung dan diakhiri dengan tanda titik dua :. Baris berikutnya berupa blok fungsi yang akan dijalankan jika fungsi dipanggil.

```
# Define the function
def functionName:
    ...blocks of code...
# Call the function
functionName()
```

Langkah 2 : Method

Method tidak dapat dipanggil dengan sendirinya. Method Ini tergantung pada objek di mana ia didefinisikan. Dalam contoh berikut, className dideklarasikan dan tiga method telah ditentukan.

```
# Define the class
class className
```

```

# Define a method
def method1Name
...blocks of code
# Define another method
def method2Name
...blocks of code
# Define yet another method
def method3Name
...blocks of code
# Instantiate the class
myClass = className()
# Call the instantiation and associated methods
myClass.method1Name()
myClass.method2Name()
myClass.method3Name()

```

Part 3 : Function

- a. Masuk ke dalam directory **~/labs/devnet-src/python**
- b. Buka file baru dan simpan dengan nama : **myCity.py** . Letakkan di dalam directory **~/labs/devnet-src/python**
- c. Definisikan function **myCity** dengan argument/nilai inputan city, yang akan digunakan untuk nama city. Ketika fungsi dipanggil dengan nama kota tertentu, maka function akan mencetak pernyataan yang menyertakan nama kota tersebut.

```

def myCity(city):
    print("I live in " + city + ".")

```

- d. Panggil function **myCity**, dengan menggunakan nilai inputan yang berbeda,

```

myCity("Austin")
myCity("Tokyo")
myCity("Salzburg")

```

- e. Simpan dan run file **myCity.py**

```

devasc@labvm:~/labs/devnet-src/python$ python3 myCity.py
I live in Austin.
I live in Tokyo.
I live in Salzburg.

```

```
devasc@labvm:~/labs/devnet-src/python$
```

Part 4 : Mendefinisikan class dengan method

Langkah 1 : Mendefinisikan dan kemudian inisiasikan class dengan method `__init__()`

- Open new file dan simpan file dengan nama : **myLocation.py**
- Definisikan class dengan nama Location . Catatan : jika bekerja dengan menggunakan VS Code maka untuk script yang ada di line 2 harus berada di space seperti di bawah ini :

```
class Location:
    |<-- cursor should now be here
```

- Kemudian, inisialisasikan `__init__()` function.

```
def __init__(self, name, country):
    self.name = name
    self.country = country
```

```
|<-- cursor should now be here
```

- Tambahkan script dibawah ini :

```
loc = Location("Your_Name", "Your_Country")
```

- Tambahkan script dibawah ini :

```
print(loc.name)
print(loc.country)
```

- Tambahkan script dibawah ini :

```
print(type(loc))
```

- Jalankan script diatas dengan perintah dibawah ini :

```
devasc@labvm:~/labs/devnet-src/python$ python3 myLocation.py
Your_Name
Your_Country
<class '__main__.Location'>
devasc@labvm:~/labs/devnet-src/python$
```

langkah 2 : Tambahkan method di class Location

- Hapus kode yang dimulai dengan instantiation dari kelas loc. Pastikan Script myLocation.py seperti pada code dibawah ini :

```
class Location:
```

```
def __init__(self, name, country):
    self.name = name
    self.country = country
```

- b. Tambahkan script dibawah ini :

```
self.country = country
|<--Your cursor should be here
```

- c. Tambahkan script dibawah ini :

```
def myLocation(self):
    print("Hi, my name is " + self.name + " and I live in
" +
self.country + ".")
```

- d. Tekan tombol Enter dua kali dan backspace dua kali.
e. Simpan dan jalankan skrip Anda untuk memastikan tidak ada kesalahan.

Langkah 3 :Inisiasi class Location beberapa kali dan panggil method myLocation

- a. Tambahkan code di bawah ini ke dalam file **myLocation.py**

```
# First instantiation of the class Location
loc1 = Location("Tomas", "Portugal")
# Call a method from the instantiated class
loc1.myLocation()
```

- b. Save dan run scriptnya

```
devasc@labvm:~/labs/devnet-src/python$ python3 myLocation.py
Hi, my name is Tomas and I live in Portugal.
devasc@labvm:~/labs/devnet-src/python$
```

- c. Tambahkan script selanjutnya :

```
loc2 = Location("Ying", "China")
loc3 = Location("Amare", "Kenya")
loc2.myLocation()
loc3.myLocation()
your_loc = Location("Your_Name", "Your_Country")
your_loc.myLocation()
```

- d. Save dan run

```
devasc@labvm:~/labs/devnet-src/python$ python3 myLocation.py
Hi, my name is Tomas and I live in Portugal.
```

Hi, my name is Ying and I live in China.

Hi, my name is Amare and I live in Kenya.

Hi, my name is Your_Name and I live in Your_Country.

devasc@labvm:~/labs/devnet-src/python\$

Langkah 4 : Review script myLocation.py seperti dibawah ini :

```
# Define a class with variables for **name** and **country**.
# Then define a method that belongs to the class. The method's
# purpose is to print a sentence that uses the variables.
class Location:
    def __init__(self, name, country):
        self.name = name
        self.country = country

    def myLocation(self):
        print("Hi, my name is " + self.name + " and I live in " +
self.country + ".")

# First instantiation of the Location class
loc1 = Location("Tomas", "Portugal")
# Call a method from the instantiated class
loc1.myLocation()

# Three more instantiations and method calls for the Location class
loc2 = Location("Ying", "China")
loc3 = Location("Amare", "Kenya")
loc2.myLocation()
loc3.myLocation()
your_loc = Location("Your_Name", "Your_Country")
your_loc.myLocation()
```

Part 5 : Review the circleClass.py Script

Given a radius value, print the circumference of a circle.

Formula for a circumference is $c = \pi * 2 * \text{radius}$

class Circle:

```
def __init__(self, radius):
    self.radius = radius

def circumference(self):
    pi = 3.14
    circumferenceValue = pi * self.radius * 2
    return circumferenceValue

def printCircumference(self):
    myCircumference = self.circumference()
    print ("Circumference of a circle with a radius of " + str(self.radius)
+ " is " + str(myCircumference))

# First instantiation of the Circle class.
circle1 = Circle(2)
# Call the printCircumference for the instantiated circle1 class.
circle1.printCircumference()

# Two more instantiations and method calls for the Circle class.
circle2 = Circle(5)
circle2.printCircumference()

circle3 = Circle(7)
circle3.printCircumference()
```