

# Full stack web development using python

## `__init__()` method



Saurabh Shukla (MySirG)

## Agenda

- ① `__init__()` method
- ② Defining `__init__()` method
- ③ Job of `__init__()` method
- ④ Providing Arguments to `__init__()`

## \_\_init\_\_( ) method

- You can define \_\_init\_\_( ) method in the class ( that is , it is optional)
- Similar to the concept of constructor in C++ or Java
- \_\_init\_\_( ) method invokes implicitly everytime when an instance object is created.
- Therefore, \_\_init\_\_( ) method is the first method runs for an object , just after the object creation.

## Defining \_\_init\_\_() method

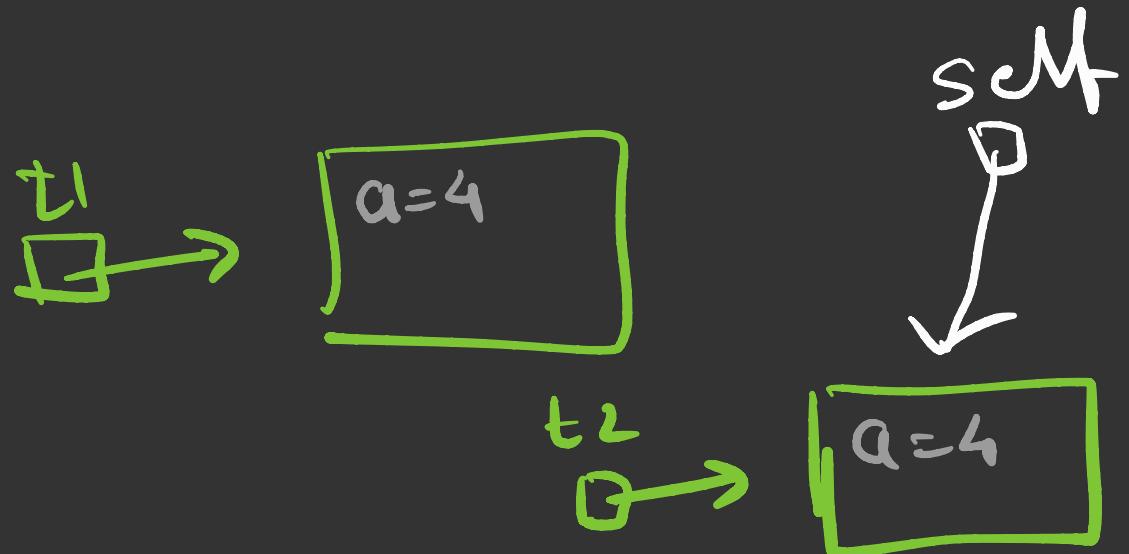
class Test :

def \_\_init\_\_(self):

- It is mandatory to take at least one positional argument in \_\_init\_\_() method
- You can give any name to the first argument of \_\_init\_\_() method but self is recommended.

```
class Test:  
    def __init__(self):  
        a=5 # local variable  
        self.a=4 # instance object variable
```

```
t1 = Test()      # __init__(t1)  
t2 = Test()      # __init__(t2)
```



## Job of `--init--()`

Just because you are defining `--init--()` method, you can write any code in it,

but

- you should write code in `--init--()` which you want to run immediately after object creation.
- ideally you should initialize instance object variables.

# Providing Arguments to `__init__()` method

class Test:

def \_\_init\_\_(self, a, b):

instance object  
variable

self.a = a

self.b = b

formal argument  
(local variables)

t1 = Test() # \_\_init\_\_(t1)

wrong  
(error)

t1 = Test(3, 4) # \_\_init\_\_(t1, 3, 4)

t2 = Test(5, 7) # \_\_init\_\_(t2, 5, 7)

t2 →

a=5  
b=7

t1 →

a=3  
b=4