Module 01

Class

Data Science Developer

Purwadhika
Startup and Coding School

Class

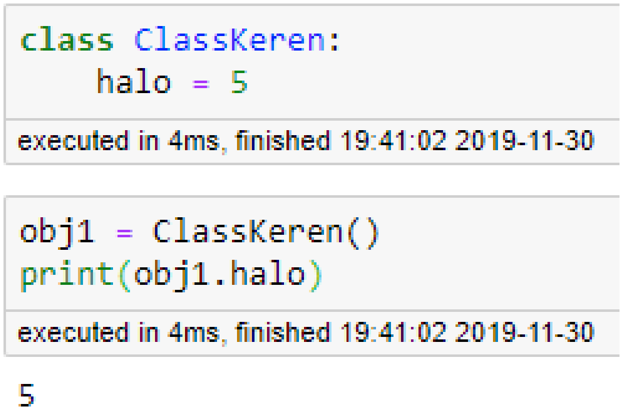
Python is an object oriented programming language.

Almost everything in Python is an object, with its properties and methods.

A Class is like an object constructor, or a "blueprint" for creating objects.

Create a Class and Object

To create a class, use the keyword class, and then we can use the class named ClassKeren to create objects:



```
class ClassKeren:
    halo = 5
```

executed in 4ms, finished 19:41:02 2019-11-30

```
obj1 = ClassKeren()
print(obj1.halo)
```

executed in 4ms, finished 19:41:02 2019-11-30

```
5
```

The __init__() Function

The examples above are classes and objects in their simplest form, and are not really useful in real life applications.

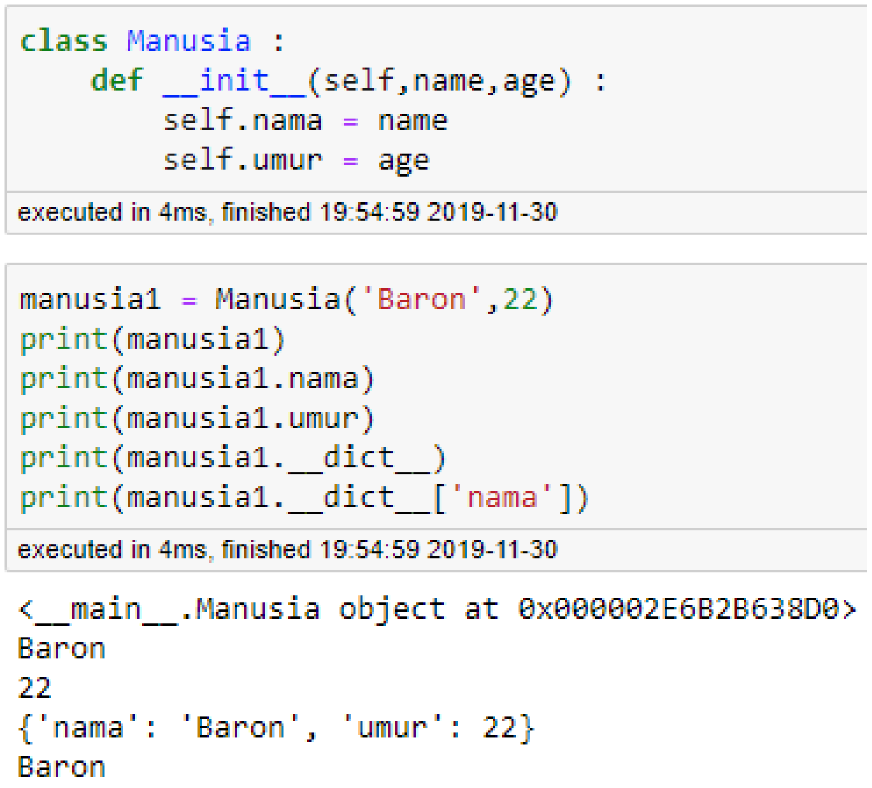
To understand the meaning of classes we have to understand the built-in __init__() function.

All classes have a function called __init__(), which is always executed when the class is being initiated.

Use the __init__() function to assign values to object properties, or other operations that are necessary to do when the object is being created:

Purwadhika
Startup and Coding School

The __init__() Function



```
class Manusia :
    def __init__(self,name,age) :
        self.nama = name
        self.umur = age
```

executed in 4ms, finished 19:54:59 2019-11-30

```
manusia1 = Manusia('Baron',22)
print(manusia1)
print(manusia1.nama)
print(manusia1.umur)
print(manusia1.__dict__)
print(manusia1.__dict__['nama'])
```

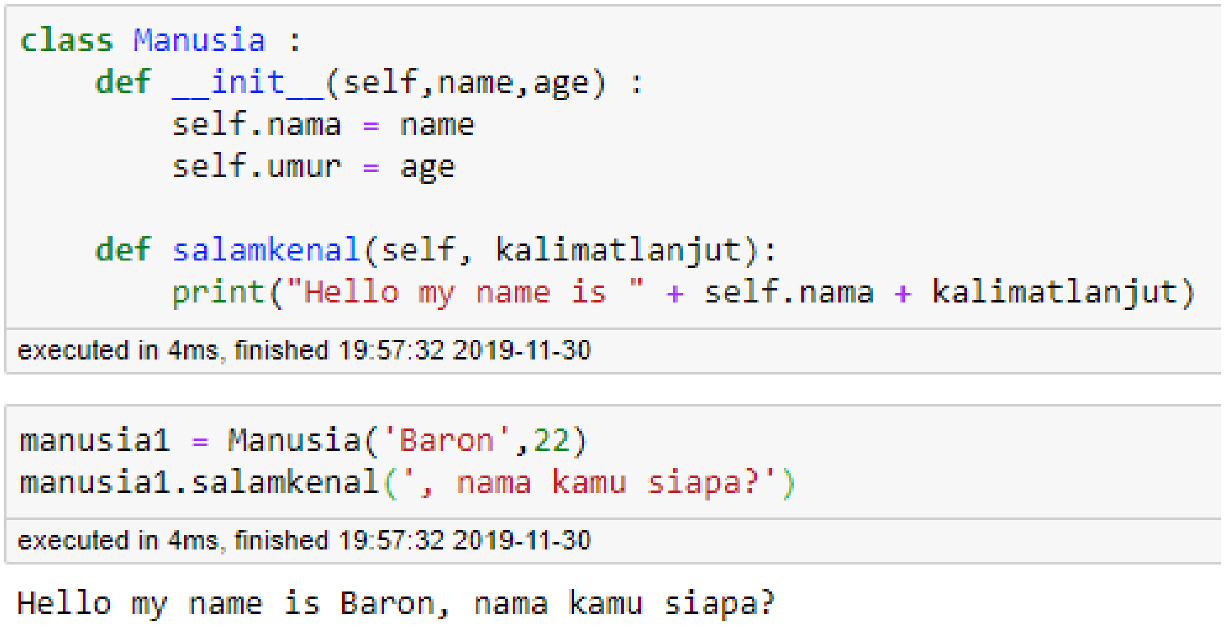
executed in 4ms, finished 19:54:59 2019-11-30

```
<__main__.Manusia object at 0x000002E6B2B638D0>
Baron
22
{'nama': 'Baron', 'umur': 22}
Baron
```

Object Methods

Objects can also contain methods. Methods in objects are functions that belong to the object.

Let us create a method in the Manusia class:



```python
class Manusia :
    def __init__(self,name,age) :
        self.nama = name
        self.umur = age

    def salamkenal(self, kalimatlanjut):
        print("Hello my name is " + self.nama + kalimatlanjut)
```

executed in 4ms, finished 19:57:32 2019-11-30

```python
manusia1 = Manusia('Baron',22)
manusia1.salamkenal(', nama kamu siapa?')
```

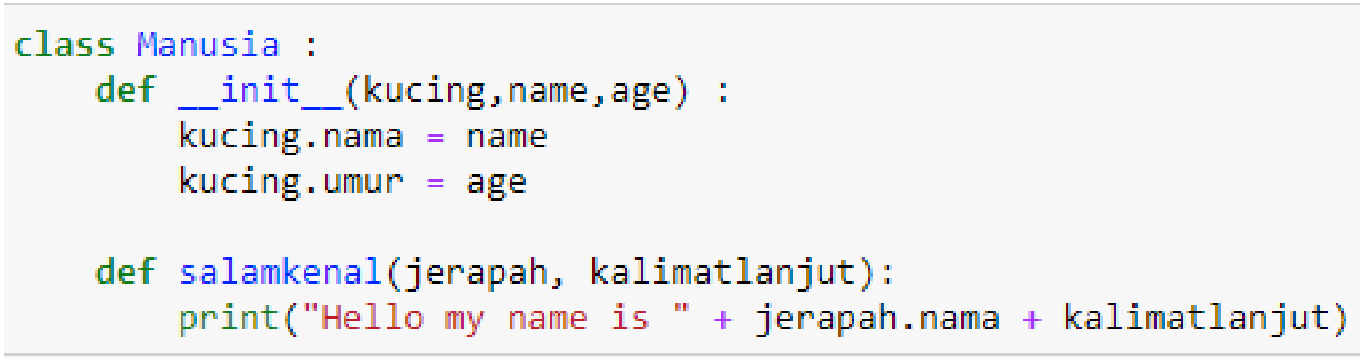
executed in 4ms, finished 19:57:32 2019-11-30

```
Hello my name is Baron, nama kamu siapa?
```

The self Parameter

The self parameter is a reference to the current instance of the class, and is used to access variables that belongs to the class.

It does not have to be named self , you can call it whatever you like, but it has to be the first parameter of any function in the class:



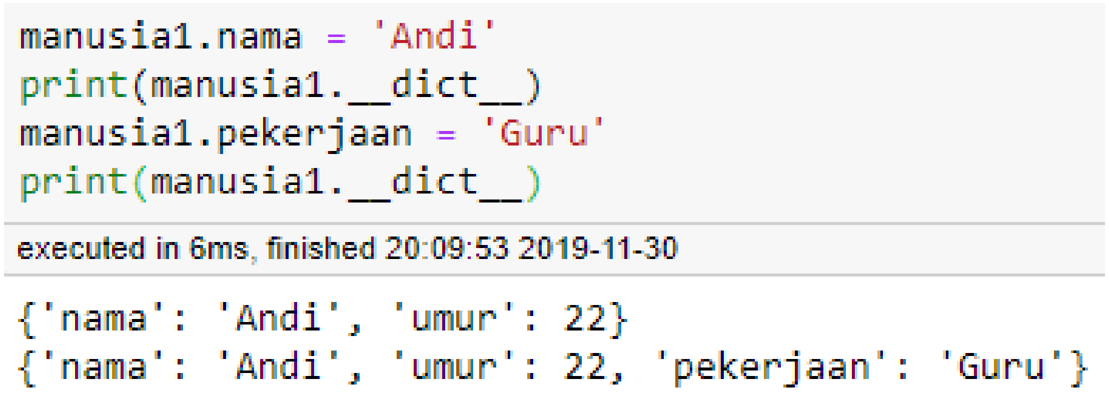
```python
class Manusia :
    def __init__(kucing,name,age) :
        kucing.nama = name
        kucing.umur = age

    def salamkenal(jerapah, kalimatlanjut):
        print("Hello my name is " + jerapah.nama + kalimatlanjut)
```

Purwadhika
Startup and Coding School

Modify Object Properties

You can modify properties on objects like this, or even add a new one :



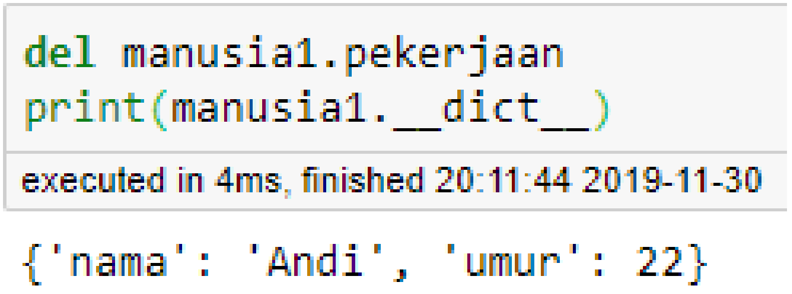
```
manusia1.nama = 'Andi'
print(manusia1.__dict__)
manusia1.pekerjaan = 'Guru'
print(manusia1.__dict__)
```

```
executed in 6ms, finished 20:09:53 2019-11-30
```

```
{'nama': 'Andi', 'umur': 22}
{'nama': 'Andi', 'umur': 22, 'pekerjaan': 'Guru'}
```

Delete Object Properties

You can delete properties on objects by using the del keyword:



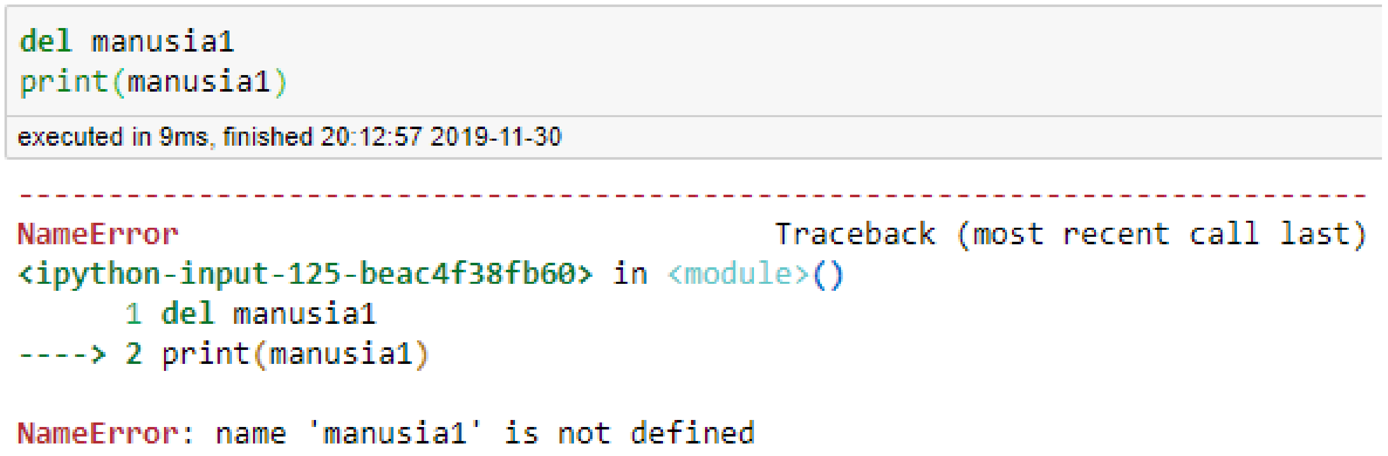
```
del manusia1.pekerjaan
print(manusia1.__dict__)
```

executed in 4ms, finished 20:11:44 2019-11-30

```
{'nama': 'Andi', 'umur': 22}
```

Delete Objects

You can delete objects by using the del keyword:



```
del manusia1
print(manusia1)
```

executed in 9ms, finished 20:12:57 2019-11-30

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-125-beac4f38fb60> in <module>()
      1 del manusia1
----> 2 print(manusia1)

NameError: name 'manusia1' is not defined
```