

# Introduction to **python**

Classes

**Mauricio Sevilla**

---

email= `j.sevillam@uniandes.edu.co`

04.03.19

This is the last class of the introductory part of python, then we'll explore the intermediate level where libraries such as `numpy` and `matplotlib`.

That is why this topic is so important, it gathers together all the previous concepts and tools.

As we have mentioned during the class, `python` is a object oriented based language and today we will understand what does it mean.

The missing structure is called `classes`

```
In [1]: class animal:
        def __init__(self, name, age, specie):
            self.name=name
            self.age=age
            self.spe=specie
```

```
In [2]: A=animal( 'Scooby',2, 'Dog' )
```

```
In [3]: A.age=3
```

```
In [4]: A.age
```

```
Out[4]: 3
```

```
In [5]: Zoo=[ ]
```

```
In [6]: animals=[ 'Dog' , 'Cat' , 'Mouse' ]  
names=[ 'Scooby' , 'Tom' , 'Jerry' ]
```

```
In [7]: for i in range(len(animals)):
        Zoo.append(animal(animals[i],1,names[i]))
```

```
In [8]: print(Zoo)
```

```
[<__main__.animal object at 0x10ddf04e0>, <__main__.animal object at 0x10ddf0518>, <__main__.animal object at 0x10ddf0550>]
```

Take a look at this structure

```
In [9]: print(Zoo[0])
```

```
<__main__.animal object at 0x10ddf04e0>
```



What do you think it means?

*We saved there a Dog named Scooby and 1 year old!,*

But, What if....

```
In [10]: print(Zoo[0].name,Zoo[0].spe,Zoo[0].age)
```

Dog Scooby 1

Let us construct a different example.

*We are going to work on a problem we already did, Let us create ourselves!*

Lets have a student class.

```
In [11]: import random
random.seed(10987654321012345678910)
class student:
    def __init__(self,name,age,career,semester):
        self.name=name
        self.age=age
        self.career=career
        self.semester=semester
    def Grade(self):
        return round(random.random()*5,1)
```

```
In [12]: Me=student('Mauricio',80,'Professor',1)
```

```
In [13]: Me.Grade()
```

```
Out[13]: 2.2
```

```
In [14]: Me.Grade()
```

```
Out[14]: 0.0
```

```
In [15]: Me.Grade()
```

```
Out[15]: 3.5
```

Let us create a more complex class

```
In [16]: class animal2:
          "This class helps you to create an animal"
          def __init__(self,name,age,specie,talk):
              self.name=name
              self.age=age
              self.spe=specie
              self.speech=talk
          def talk(self):
              return self.speech
          def sit(self):
              if(self.spe=='Dog'):
                  return 'Sitted!'
              else:
                  return 'Nee'
```

```
In [17]: Zoo2=[ ]
animals=[ 'Dog' , 'Cat' , 'Mouse' ]
names=[ 'Scooby' , 'Tom' , 'Jerry' ]
ages=[1,2,3]
speeches=[ 'Woof' , 'Miau!' , 'Cheese!' ]
```

```
In [18]: for i in range(len(animals)):
          Zoo2.append( animal2( names[i] , 1 , animals[i] , speeches[i] ) )
```

```
In [19]: print(Zoo2[-1].talk())
```

Cheese!

```
In [20]: print(Zoo2[2].name)
```

Jerry

```
In [21]: print(Zoo2[2].sit())
```

Nee

```
In [22]: print(Zoo2[1].talk())
```

Miau!

```
In [23]: ?animal2
```



# Inheritance

Let see some examples of what does inheritance means while programming.

For example, my family as a class

```
In [24]: class Sevilla:
          def __init__(self, name, age):
              self.name=name
              self.age=age
          def hair(self):
              return 'Black' #We all have black hair
          def eyes(self):
              return 'Brown' #We all have Brown eyes
          def LastName(self):
              return 'Sevilla' #We all have the same last name
```

If someday i have a son/daughter, for sure he/she will have some features I do, so

```
In [25]: class SonDaughter(Sevilla):  
         def __init__(self,name):  
             self.na=name
```

```
In [26]: Me=Sevilla('Mauricio',80)
```

```
In [27]: MySon=SonDaugther('son')
```

```
In [28]: print(MySon.na)
```

son

```
In [29]: print(MySon.hair())
```

Black

```
In [30]: print(MySon.eyes())
```

Brown

```
In [31]: print(MySon.LastName())
```

Sevilla