# 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 `class` es

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

In [2]:
```
animalito=animal('Comino',2,'Dog')
animalito.spe
```

Out[2]:
```
'Dog'
```

In [3]:
```
A=animal('Scooby',2,'Dog')
```

In [4]:
```
A.age=3
```

In [5]:
```
A.age
```

Out[5]: 3

In [6]:
```
Zoo=[]
```

In [7]:
```
animals=['Dog','Cat','Mouse']
names=['Scooby','Tom','Jerry']
```

In [8]:
```
for i in range(len(animals)):
    Zoo.append(animal(names[i],1,animals[i]))
```

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

```
[<__main__.animal object at 0x1040846a0>, <__main__.animal object at 0x1040846
d8>, <__main__.animal object at 0x104084710>]
```

Take a look at this structure

In [10]:
```
print(Zoo[0])
```

```
<__main__.animal object at 0x1040846a0>
```

What do you think it means?

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

But, What if....

In [11]:
```
print(Zoo[0].name,Zoo[0].spe,Zoo[0].age)
```

```
Scooby Dog 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 [12]:
```
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 [13]:
```
Me=student('Mauricio',80,'Professor',1)
```

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

Out[14]: 2.2

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

Out[15]: 0.0

In [16]:
```
Me.Grade()
```

Out[16]: 3.5

Let us create a more complex class

In [17]:
```
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 [18]:
```
Zoo2=[]
animals=['Dog','Cat','Mouse']
names=['Scooby','Tom','Jerry']
ages=[1,2,3]
speeches=['Woof','Miau!','Cheese!']
```

In [19]:
```
for i in range(len(animals)):
    Zoo2.append(animal2(names[i],1,animals[i],speeches[i]))
```

In [20]:
```
print(Zoo2[-1].talk())
```

```
Cheese!
```

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

```
Jerry
```

In [22]:
```
print(Zoo2[2].sit())
```

```
Nee
```

In [23]:
```
print(Zoo2[1].talk())
```

```
Miau!
```

In [24]:
```
?animal2
```

# Inheritance

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

For example, my family as a class

In [25]:
```
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 [26]:
```
class SonDaugther(Sevilla):
    def __init__(self,name):
        self.na=name
```

In [27]:
```
Me=Sevilla('Mauricio',80)
```

In [28]:
```
MySon=SonDaugther('son')
```

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

```
son
```

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

```
Black
```

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

```
Brown
```

In [32]:
```
print(MySon.LastName())
```

```
Sevilla
```

In [33]:
```
class SonDaugther2(Sevilla):
    def __init__(self,name):
        Sevilla.__init__(self,'Nombre',10)
        self.na=name
```

In [34]:
```
MySon2=SonDaugther2('name')
```

In [35]:
```
MySon2.age
```

Out[35]:
```
10
```