

LS4 TP6

```
>>> p = Point(3,4) # crée l'objet p
>>> print(p.__doc__) # affiche la doc de la classe uniquement
Une classe Point
>>> p # affiche de quelle classe provient l'objet
<objet.Point object at 0x100722ed0>

>>> del p
>>> print(Point.nbr_de_points)
3
```

Les objets

Python est un langage orienté objet. les données sont des **objets**

```
class Nom_classe :
```

Par convention, le nom de la classe commence par une majuscule

Le corps de la classe est indenté

Pour initialiser les champs des instances, on définit une méthode et **unique**:

```
def __init__(self):
```

Le **1er paramètre** de cette methode est un parametre faisant reference à l'objet (ie à l'instance créée) et communément appelé "**self**"

Les parametres suivant sont qqconques etp ermettent en général, d'initialiser les champs de l'instance

```
class Rectangle :

    def __init__(self,h,l) :
        self.hauteur = h
        self.longueur = l

>>> r = Rectangle(10,4) (*)
>>> r.hauteur
10
>>>r.couleur
*affiche un message d'erreur*
```

`__init__` est la 1ere méthode appelée après l'instanciation de l'objet. Lorsque l'instruction (*) est exécutée, l'objet `r` est créé, une identité lui est affectée

`(>>>id(r))`, puis la méthode `init` est appelé

Remarque :

"self" ne peut être utilisée qu'à l'intérieur de la classe. C'est l'équivalent du "this" de java

On peut ensuite définir des méthodes dans la classe

```
class Rectangle :  
  
    def __init__(self,h,l) :  
        self.hauteur = h  
        self.longueur = l  
  
    def aire(self) :  
        return self.hauteur * self.largeur  
  
>>>r.aire()  
40  
  
    def ecrire(self) :  
        print("hauteur : "+ self.hauteur)
```

Les variables de classe

Une variable de classe est une variable commune à toutes les instances de la classe

```
class Rectangle :  
  
    couleur = "rouge"  
  
    def __init__(self,h,l) :  
        self.hauteur = h  
        self.longueur = l  
  
    def aire() :  
        return self.hauteur * self.largeur
```

`couleur` est une variable de la classe

Les méthodes de classe

```
def nom_class(cls): # cls = self pour les methodes  
    print("La classe est la classe Rectangle")
```

```
nom_class = classmethod(nom_class)
```

```
>>>Rectangle.nom_class()
```

```
>>>r.nom_class()
```