

# PYTHON INITIATION

---

Présenté par :  
**Xavier TABUTEAU**

# Les classes

## Les classes

- **Orienté objet : ça veut dire quoi ?**

Globalement, les langages de programmation objet implémentent le paradigme de programmation orientée objet (POO). Ce paradigme consiste en la réunion des données et des traitements associées à ces données au sein d'entités cohérentes appelées objets. Python est un langage objet composé de classes. Une classe représente un « moule » permettant de créer des objets (instances), et regroupe les attributs et méthodes communes à ces objets.

- **Les classes**

L'orienté objet facilite beaucoup dans la conception, la maintenance, la réutilisabilité des éléments (objets). Le paradigme de POO permet de tirer profit de classes parents et de classes enfants (phénomène d'héritage), etc.

Tout objet donné possède deux caractéristiques :

Son état courant (attributs)

Son comportement (méthodes)

En approche orienté objet on utilise le concept de classe, celle-ci permet de regrouper des objets de même nature.

Une classe est un moule (prototype) qui permet de définir les attributs (variables) et les méthodes (fonctions) de tous les objets de cette classe.

Les classes sont les principaux outils de la POO. Ce type de programmation permet de structurer les logiciels complexes en les organisant comme des ensembles d'objets qui interagissent entre eux et avec le monde extérieur.

# Les classes

- Exemple de classe

## Les classes

```
#Class avec COnstructor
class Velo:
    roues = 2

    def __init__(self, marque, prix, poids):
        self.marque = marque
        self.prix = prix
        self.poids = poids

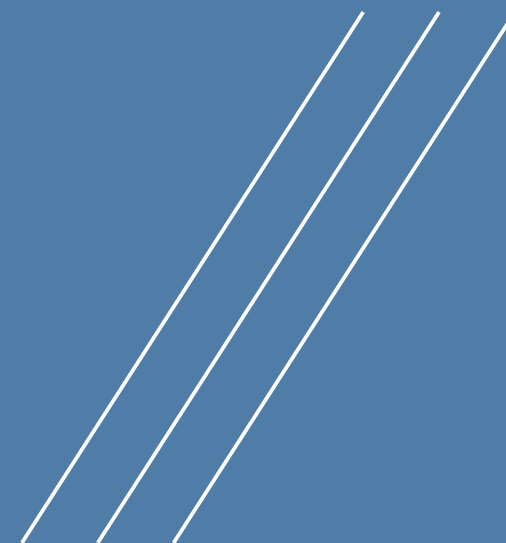
    def rouler(self):
        print("Wouh, ça roule mieux avec un vélo {} !".format(self.marque))
```

## Les classes

# Quelques remarques importantes

- Tous les attributs et méthodes des classes Python sont « publics » au sens de C++, parce que « nous sommes tous des adultes ! » (citation de Guido von Rossum, créateur de Python).
- Le constructeur d'une classe est une méthode spéciale qui s'appelle `__init__()`.
- En Python, on n'est pas tenu de déclarer tous les attributs de la classe comme d'autres langages : on peut se contenter de les initialiser dans le constructeur !
- Toutes les méthodes prennent une variable `self` comme premier argument. Cette variable est une référence à l'objet manipulé.
- Python supporte l'héritage simple et l'héritage multiple. La création d'une classe fille est relativement simple, il suffit de préciser entre parenthèses le nom de la classe mère lors de la déclaration de la classe fille.

# PYTHON INITIATION



Présenté par  
**Xavier TABUTEAU**