PYTHON

Présenté par :

Xavier TABUTEAU

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

- **Attributs de classe**

Une classe peut également avoir des attributs. Pour cela, il suffit de les déclarer dans le corps de la classe. Les attributs de classe sont accessibles depuis la classe elle-même et sont partagés par tous les objets. Si un objet modifie un attribut de classe, cette modification est visible de tous les autres objets. Les attributs de classe sont le plus souvent utilisés pour représenter des constantes.

- **Méthode de classe**

Tout comme il est possible de déclarer des attributs de classe, il est également possible de déclarer des méthodes de classe. Pour cela, on utilise le décorateur @classmethod. Comme une méthode de classe appartient à une classe, le premier paramètre correspond à la classe. Par convention, on appelle ce paramètre cls pour préciser qu'il s'agit de la classe et pour le distinguer de self.

- **Méthode statique**

Une méthode statique est une méthode qui appartient à la classe mais qui n'a pas besoin de s'exécuter dans le contexte d'une classe. Autrement dit, c'est une méthode qui ne doit pas prendre le paramètre cls comme premier paramètre. Pour déclarer une méthode statique, on utilise le décorateur @staticmethod. Les méthodes statiques sont des méthodes utilitaires très proches des fonctions mais que l'on souhaite déclarer dans le corps d'une classe.

Les classes

- **Exemple de classe**

```python
#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))
```

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__().

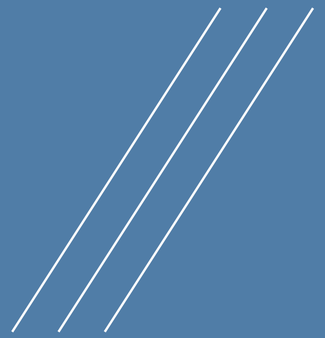
- En Python, on n'est pas tenu de déclarer tous les attributs de la classe comme d'autres langages : on peut se contenter de les initialiser dans le constructeur !

- 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.

classes.py
ex_cercle_cylindre
ex_compte_bancaire
ex_compte_epargne
ex_jeu_de_cartes
ex_jeu_a_et_b

PYTHON



Présenté par

Xavier TABUTEAU