Pense bête 5

Les Classes

Les classes et les objets sont des outils de programmation très puissants.

Une classe est une « classification" d'objets : Personne, Image, Animal etc.

Un objet est une instance particulière de la classe : Pierre est une instance de la classe Personne.

Les objets ont des attributs. Une instance de la classe Personne aura comme attributs : nom, taille, age etc.

Les méthodes définisent ce que les objets peuvent faire : sauter, courir etc.

Les attributs et les méthodes sont décrits dans la classe : c'est le moule des objets. Chaque objet créé sur cette classe, possède les attributs et les méthodes décrits dans la classe.

Dans un jeu vidéo, nous avons besoin de données pour décrire les personnages : par exemple le nom, le nombre de points de vie maximum, le nombre de point de vie en cours, l'image à afficher.

Pour afficher les caractéristiques d'un personnage nous devrons faire appel à une fonction du type :

```
def afficher(nom, nombre_point_max, nombre_point_courant) :
    print( nom, nombre_point_max, nombre_point_courant)
```

Avec un seul personnage, nous pouvons nous en sortir, mais si notre jeu comporte plusieurs personnages, des monstres, des obstacles etc, nous devrions créer autant de variables différentes pour chaque caractéristique de chaque personnage et autant de fonctions adaptées pour les afficher.

De plus, si nous décidons après avoir déjà bien avancé dans le programme de rajouter une caractéristique à certains personnages, il faudrait modifier tous les

personnages concernés et toutes les fonctions impactées par cette modification.

Si maintenant nous regroupons toutes les informations concernant un type de personnage, ou de monstre dans une structure particulière qui va servir de modèle et qu'ensuite nous créons un personnage ou un monstre en indiquant que ce personnage ou ce monstre est créé suivant tel ou tel modèle, il ne sera plus nécessaire de multiplier les variables, ni les fonctions permettant de manipuler les personnages ou les monstres.

Exemple :

```
#Définition de la classe Personnage
class  Personnage ():
    """Classe définissant un personnage"""
    def __init__(self)
        """Méthode définissant les attributs d'un personnage"""
        self.nom = "Maurice"
        self.nombre_point_max = 50
        self.nombre_point_courrent = 50
        self.vitesse = 10
        self.nombre_arme = 8
```

Le mot **self** représente l'objet qui sera construit à partir de cette classe.

Chaque personnage construit à partir de cette classe possèdera les attributs définis dans la classe, et ces attributs leurs seront propres.

Au départ, lors de la construction du personnage (exécution de la méthode init), tous les personnages auront des attributs ayant les mêmes valeurs, mais ensuite chaque personnage pourra modifier les valeurs de ses propres attributs.

La méthode init est appelée constructeur car c'est elle qui est exécutée **lorsqu'on crée une instance de la classe**.

Les Objets

Les objets sont des instances d'une classe.

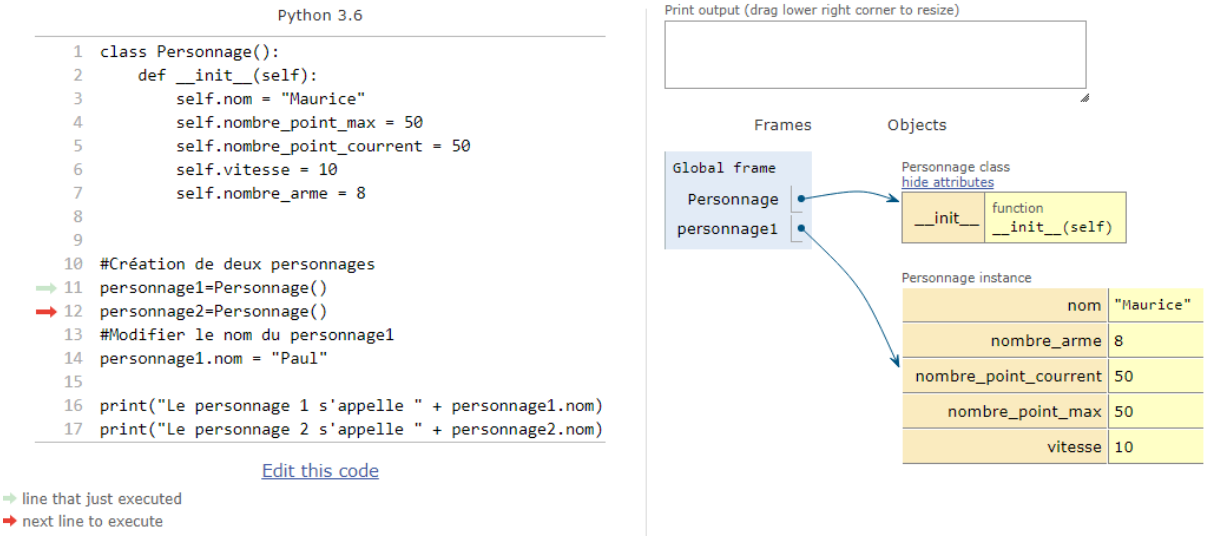
```
#Création de deux personnages
personnage1=Personnage()
personnage2=Personnage()
```

A ce stade les deux personnages ont les mêmes attributs qui ont les mêmes valeurs. Par exemple, ils s'appellent tous les deux Maurice.

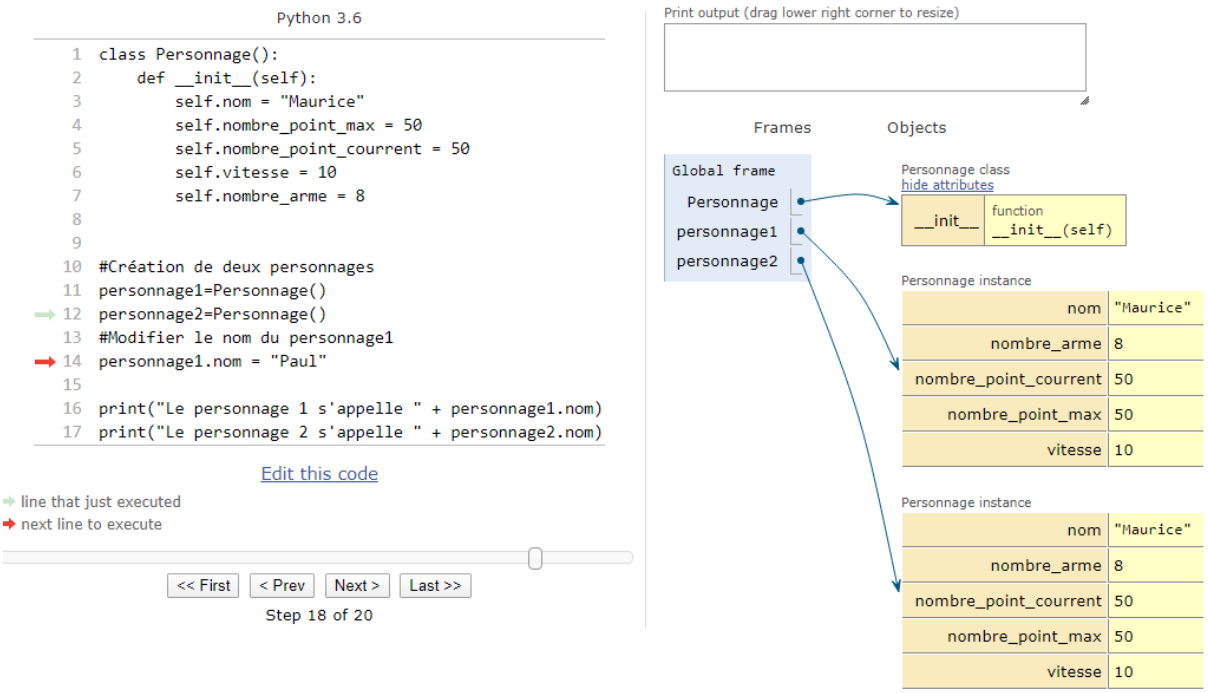
Voici une copie d'écran d'un outil permettant de visualiser ce qui se passe lors de l'exécution d'un programme python :

http://www.pythontutor.com/

Ici nous sommes à la fin de l'exécution de la ligne 11 :
personnage1=Personnage()



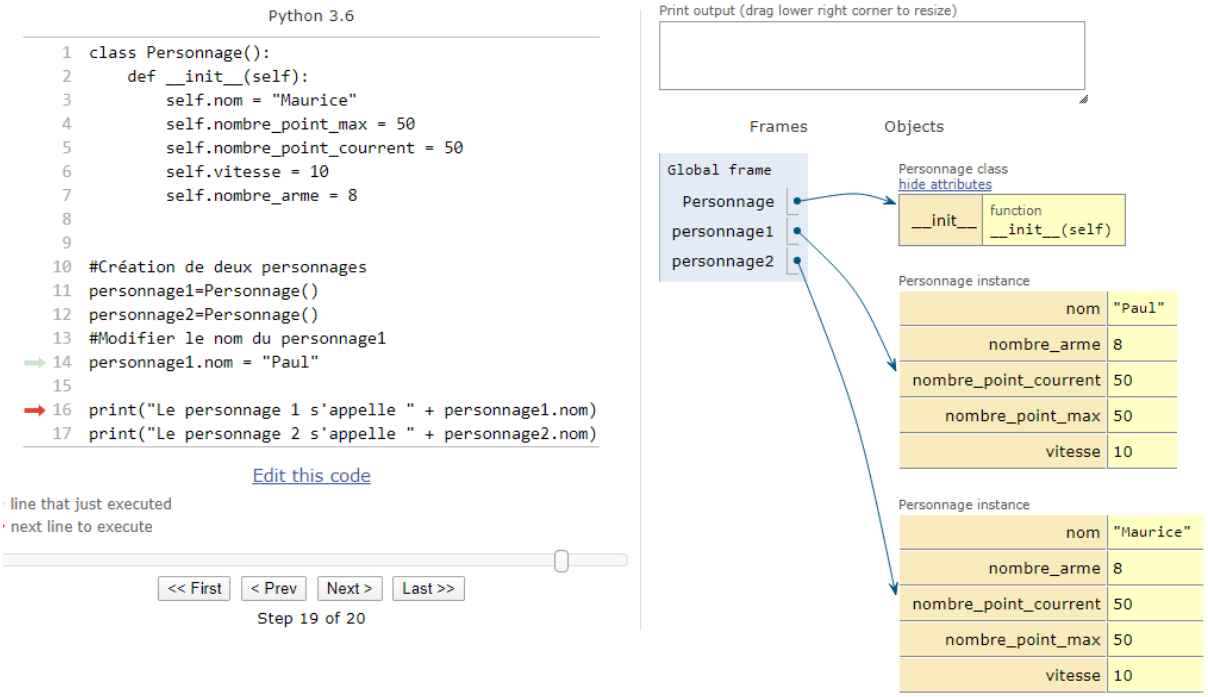
Après l'exécution de la ligne 12, nous avons en mémoire :



Si nous écrivons :

personnage1.nom = "Paul"

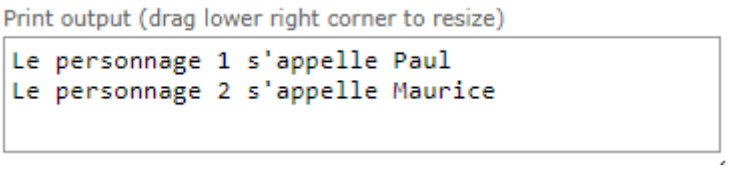
nous changeons la valeur de l'attribut nom, uniquement dans le personnage1.



Pour afficher le nom des deux personnages nous pouvons écrire :

```
print("Le personnage 1 s'appelle " + personnage1.nom)
print("Le personnage 2 s'appelle " + personnage2.nom)
```

L'exécution de ces deux lignes donnent :



Les méthodes

Ajoutons une méthode à notre classe Personnage, permettant d'afficher la valeur de tous les attributs d'un personnage.

```
class  Personnage ():
    """Classe définissant un personnage"""
    def __init__(self)
        """Méthode définissant les attributs d'un personnage"""
        self.nom = "Maurice"
        self.nombre_point_max = 50
        self.nombre_point_courrent = 50
        self.vitesse = 10
        self.nombre_arme = 8


    def affiche_attribut(self) :
        print("Nom : ", self.nom)
        print("Nombre de points max : ",self.nombre_point_max)
        print("Nombre de points courrent : " ,
self.nombre_point_courrent)
        print("Vitesse : ",vitesse)
        print("Nombre d'armes : ",self.nombre_arme)
```

```
#Création de deux personnages
personnage1=Personnage()
personnage2=Personnage()
# Changer le nom du personnage 1
personnage1.nom = "Paul"
#Afficher la valeur des attributs des deux personnages
print("personnage 1 :")
print(personnage1.affiche_attribut())
print("personnage 2 :")
print(personnage2.affiche_attribut()
```

Le paramètre **self** de la méthode **affiche_attribut**, fournit à cette méthode la **référence** à l'objet personnage sur laquelle elle doit travailler. Elle affichera les attributs de l'objet dont elle possède la référence.

Cette référence est fournie lors de l'appel de la méthode :

personnage1.affiche_attribut() fournit à la méthode, la référence à l'objet personnage1. Grace à cette référence, la méhode retrouve les valeurs des attributs de personnage1.