

TRAVAUX PRATIQUES
La programmation en Python
T.P. n°10 : POO_Classe_Méthodes_Attributs

BTS SN-IR
1 <sup>ère</sup> année
Page 1 sur 2

### Les classes :

```

1  class Voiture:                                #Définition de la classe
2      def __init__(self,marque, couleur):        #Constructeur
3          self.marque = marque                  #Attribut d'instance
4          self.couleur = couleur                #Autre attribut d'instance
5
6  voiture01 = Voiture("Porsche","Rouge")        #Création de l'objet ou instantiation de l'objet
7  voiture02 = Voiture("Lamborghini","Jaune")    #Création de l'objet ou instantiation de l'objet
8  print("la voiture est une",voiture01.marque)   #Affichage des attribut de l'objet!!!!!!
9  print("la voiture est de couleur",voiture01.couleur) #Affichage des attribut de l'objet
10 print(f"la voiture est une {voiture02.marque}, elle est de {voiture02.couleur}")
11

```

### POINTS IMPORTANTS À RETENIR

- Pour créer une classe, on utilise le mot clé class suivi du nom de la classe.
- Pour créer des attributs de classe, on utilise exactement la même syntaxe que pour déclarer une variable, en s'assurant que ces variables soient définies à l'intérieur de la classe grâce à l'indentation.
- La fonction `__init__` est une méthode qui permet de construire les objets
- N'oubliez pas le mot clé `self` qui fait référence à l'instance de l'objet
- Déclarer les attributs de l'instance à l'intérieur du constructeur `__init__`
- Attention à l'indentation qui est au même niveau que la définition de la classe pour instancier les objets

### Les méthodes et les attributs de classe :

```

1  class Voiture:
2      voiture_crees = 0                        # Attribut de classe
3      def __init__(self,marque,couleur):
4          Voiture.voiture_crees += 1          # Inrémentation à chaque instantiation
5          self.marque = marque
6          self.couleur = couleur
7
8      def affichage(self):                    # Définition de la méthode
9          print(f"La voiture est une {self.marque} de couleur {self.couleur}.")
10
11
12  maVoiture = Voiture("lamborghini","Rouge") #Instantiation de l'objet
13  taVoiture = Voiture("Porsche", "bleue")    # Création de l'objet de classe
14
15  maVoiture.affichage()                      # Appel de la méthode
16  taVoiture.affichage()
17  #Affichage de nombre d'objets créés à partir de la ttribut de classe
18  print(f"Le nombre de voitures créés est {Voiture.voiture_crees}")
19

```

TRAVAUX PRATIQUES
La programmation en Python
T.P. n°10 : POO_Classe_Méthodes_Attributs

BTS SN-IR
1 <sup>ère</sup> année
Page 2 sur 2

**Exercice 1 : Comment créer une classe Voiture qui permet de représenter une voiture qui avance et fait le plein d'essence ?**

- Créez une classe voiture avec un attribut 'essence' qui est égal à 100.
- Créez une méthode 'afficher\_reservoir' qui affiche le nombre de litres d'essence restant ('La voiture contient x litres d'essence').
- Créez une méthode 'roule' avec un paramètre km (kilomètre) qui va faire avancer la voiture et vider petit à petit le réservoir. On considère une consommation de 5L pour 100km, l'opération mathématique pour déterminer le nombre de litres d'essence nécessaire en fonction du nombre de kilomètres est donc :  

$$(km * 5) / 100$$
- Si le réservoir est vide quand on essaie de rouler, affichez la phrase : 'Vous n'avez plus d'essence, faites le plein !' et empêchez la voiture d'avancer.
- Si la jauge d'essence descend en dessous de 10L, affichez la phrase : 'Vous n'avez bientôt plus d'essence !'
- Créez une méthode 'faire\_le\_plein' qui remet le niveau d'essence à 100L et qui affiche la phrase 'Vous pouvez repartir'

**Exercice 2 : Liste de course**