

# Atelier Python : L'approche orientée objet

## Partie 1 : Enoncé exercices

### Exercice 1:

Créez une classe `Personne` avec un attribut `nom`. Instanciez deux objets `personne1` et `personne2` avec les noms "Alice" et "Bob" respectivement.

### Exercice 2:

Ajoutez une méthode `presentation` à la classe `Personne` qui affiche "Je m'appelle [nom]".

### Exercice 3:

Créez une classe `Etudiant` qui hérite de la classe `Personne`. Ajoutez un attribut `annee` à la classe `Etudiant` et une méthode `info` qui affiche le nom et l'année de l'étudiant.

### Exercice 4:

Créez une classe `Rectangle` avec des attributs `longueur` et `largeur` et une méthode `aire` qui retourne l'aire du rectangle.

### Exercice 5: (on utilisera l'importation de la bibliothèque 'math')

Créez une classe `Cercle` avec un attribut `rayon` et des méthodes `aire` et `perimetre` pour calculer respectivement l'aire et le périmètre du cercle.

### Exercice 6:

Créez une classe `Animal` avec une méthode `parler` qui affiche "Je suis un animal".

### Exercice 7:

Créez une classe `Chien` qui hérite de la classe `Animal` avec une méthode `parler` qui affiche "Je suis un chien".

### Exercice 8:

Créez une classe `Chat` qui hérite de la classe `Animal` avec une méthode `parler` qui affiche "Je suis un chat".

### Exercice 9:

Créez une classe `Vehicule` avec un attribut `vitesse` initialisé à 0 et des méthodes `accelerer` et `ralentir` qui modifient la vitesse.

### Exercice 10:

Créez une classe `Voiture` qui hérite de la classe `Vehicule` avec un attribut `modele` et une méthode `afficher\_modele` qui affiche le modèle de la voiture

## Partie 2 : Solution exercices

### Exercice 1

```
class Personne:
    def __init__(self, nom):
        self.nom = nom

personne1 = Personne("Alice")
personne2 = Personne("Bob")

print(personne1.nom) # Output: Alice
print(personne2.nom) # Output: Bob
```

### Exercice 2

```
class Personne:
    def __init__(self, nom):
        self.nom = nom
    def presentation(self):
        print("Je m'appelle", self.nom)

personne1 = Personne("Alice")
personne2 = Personne("Bob")

personne1.presentation() # Output: Je m'appelle Alice
personne2.presentation() # Output: Je m'appelle Bob
```

### Exercice 3

```
class Personne:
    def __init__(self, nom):
        self.nom = nom

    def presentation(self):
        print("Je m'appelle", self.nom)

personne1 = Personne("Alice")
personne2 = Personne("Bob")
personne1.presentation() # Output: Je m'appelle Alice
personne2.presentation() # Output: Je m'appelle Bob

class Etudiant(Personne):
    def __init__(self, nom, annee):
        super().__init__(nom)
        self.annee = annee
    def info(self):
        print("Je suis", self.nom, "et je suis en année", self.annee)

etudiant1 = Etudiant("Alice", 3)
etudiant2 = Etudiant("Bob", 2)
etudiant1.info() # Output: Je suis Alice et je suis en année 3
etudiant2.info() # Output: Je suis Bob et je suis en année 2
```

### Exercice 4

```
class Rectangle:
```

```

def __init__(self, longueur, largeur):
    self.longueur = longueur
    self.largeur = largeur
def aire(self):
    return self.longueur * self.largeur
rectangle1 = Rectangle(5, 3)
rectangle2 = Rectangle(4, 6)
print(rectangle1.aire()) # Output: 15
print(rectangle2.aire()) # Output: 24

```

#### Exercice 5

```

import math
class Cercle:
    def __init__(self, rayon):
        self.rayon = rayon
    def aire(self):
        return math.pi * self.rayon ** 2
    def perimetre(self):
        return 2 * math.pi * self.rayon
cercle1 = Cercle(3)
cercle2 = Cercle(5)
print(cercle1.aire()) # Output: 28.274333882308138
print(cercle1.perimetre()) # Output: 18.84955592153876
print(cercle2.aire()) # Output: 78.53981633974483
print(cercle2.perimetre()) # Output: 31.41592653589793

```

#### Exercice 6

```

class Animal:
    def parler(self):
        print("Je suis un animal")
animal = Animal()
animal.parler() # Output: Je suis un animal

```

#### Exercice 7

```

class Animal:
    def parler(self):
        print("Je suis un animal")
animal = Animal()

```

```
animal.parler() # Output: Je suis un animal

class Chien(Animal):
    def parler(self):
        print("Je suis un chien")
chien = Chien()
chien.parler() # Output: Je suis un chien
```

#### Exercice 8

```
class Animal:
    def parler(self):
        print("Je suis un animal")
animal = Animal()
animal.parler() # Output: Je suis un animal

class Chat(Animal):
    def parler(self):
        print("Je suis un chat")
chat = Chat()
chat.parler() # Output: Je suis un chat
```

#### Exercice 9

```
class Vehicule:
    def __init__(self):
        self.vitesse = 0
    def accelerer(self, increment):
        self.vitesse += increment
    def ralentir(self, decrement):
        self.vitesse -= decrement
voiture = Vehicule()
print(voiture.vitesse) # Output: 0
voiture.accelerer(50)
print(voiture.vitesse) # Output: 50
voiture.ralentir(20)
print(voiture.vitesse) # Output: 30
```

#### Exercice 10

```
class Vehicule:
    def __init__(self):
        self.vitesse = 0
    def accelerer(self, increment):
        self.vitesse += increment
    def ralentir(self, decrement):
        self.vitesse -= decrement
```

```
voiture = Vehicule()

class Voiture(Vehicule):
    def __init__(self, modele):
        super().__init__()
        self.modele = modele
    def afficher_modele(self):
        print("Modèle de la voiture:", self.modele)

voiture = Voiture("Toyota")
voiture.afficher_modele() # Output: Modèle de la voiture: Toyota
voiture.accelerer(20)
print(voiture.vitesse) # Output: 20
```