

Exercices de POO - Terminale NSI

Exercice 1 : Identifier attributs et méthodes

Considérons le code suivant :

```
class Voiture:
    def __init__(self, marque, modele):
        self.marque = marque
        self.modele = modele
        self.vitesse = 0

    def accelerer(self, valeur):
        self.vitesse += valeur

    def freiner(self, valeur):
        self.vitesse -= valeur
```

1. Quels sont les **attributs** de la classe `Voiture` ?\
 2. Quelles sont les **méthodes** de la classe `Voiture` ?\
 3. Que devient l'attribut `vitesse` après utilisation de la méthode `accelerer` ?
-

Exercice 2 : Attributs ou méthodes ?

Indiquez si les éléments suivants sont des **attributs** ou des **méthodes** :

- `self.nom`\
 - `def ajouter_element(self, e)`\
 - `self.compteur`\
 - `def get_valeur(self)`
-

Exercice 3 : Fonction ou méthode ?

1. Expliquez la différence entre une **fonction** et une **méthode** en Python.\
2. Donnez un exemple de fonction (en dehors d'une classe) et un exemple de méthode (dans une classe).\
3. Dans le code suivant, identifiez ce qui est une fonction et ce qui est une méthode :

```
class CompteBancaire:
    def __init__(self, solde):
        self.solde = solde

    def deposer(self, montant):
        self.solde += montant

    def retirer(self, montant):
```

```
self.solde -= montant
```

```
def afficher_bonjour():  
    print("Bonjour !")
```

Exercice 4 : Compléter une classe

On souhaite modéliser une **Lampe**. Complétez la classe suivante pour qu'elle possède :

- un attribut indiquant si elle est allumée ou éteinte,\
- une méthode `allumer()`,\
- une méthode `eteindre()`.

```
class Lampe:  
    def __init__(self):  
        # à compléter  
  
    # à compléter
```

Exercice 5 : Écriture d'une classe à partir d'un énoncé

On souhaite modéliser une **montre connectée** avec les caractéristiques suivantes :

- un attribut `heure` (en chaîne de caractères, ex : "14:35"),\
- une méthode `afficher_heure()` qui affiche l'heure actuelle,\
- une méthode `changer_heure(nouvelle_heure)` qui permet de modifier l'heure.

Travail à faire : Écrire la classe `Montre` correspondant à cet énoncé.

Exercice 6 : Instanciation et utilisation

À partir de la classe suivante :

```
class Chien:  
    def __init__(self, nom):  
        self.nom = nom  
  
    def aboyer(self):  
        print("Ouaf !")
```

1. Créez un objet `mon_chien` de type `Chien` appelé "Rex".\
2. Faites aboyer le chien en utilisant son objet.\
3. Ajoutez un attribut `age` à la classe `Chien` et une méthode `vieillir()` qui augmente l'âge de 1.

Exercice 7 : Analyse d'un énoncé et modélisation

Énoncé : On souhaite représenter une **bibliothèque** contenant plusieurs **livres**.

- Un livre possède un titre et un auteur.
- Une bibliothèque possède une liste de livres.
- La bibliothèque doit pouvoir ajouter un livre et afficher la liste de ses livres.

Travail à faire :

1. Écrire les classes nécessaires (**Livre** et **Bibliothèque**).
 2. Écrire un petit programme qui crée une bibliothèque, ajoute deux livres, puis affiche la liste des livres présents.
-