

□ Exercice 1 : Une application pour un élevage de chien

10 points

On souhaite créer une application de gestion d'un élevage de chiens, un chien de cet élevage sera représenté par une instanciation de la classe **Chien** suivante dont la signification des attributs est indiqué en commentaire :

```

1 class Chien :
2
3     def __init__(self ,nom,race ,sexe ,naissance):
4         # le nom, la race, le sexe et l'année de naissance du chien
5         self.nom = nom
6         self.race = race
7         self.sexe = sexe
8         self.naissance = naissance
9
10    def __str__(self):
11        return f'{{self.nom}} ({{self.sexe}}) {{self.race}} né en {{self.naissance}}'
```

1. Ecrire l'instruction permettant de créer l'objet **medor** de la classe **Chien**, de race "caniche", femelle née en 2017 et nommé "Médor" ?
2. Quel sera l'affichage produit par `print(medor)` ?
3. Ecrire une méthode **get_race** (un getter) qui renvoie la race d'un objet de type **Chien**.
4. Ecrire une méthode **set_nom** (un setter) qui permet de modifier le nom d'un objet de type **Chien**.
5. Ecrire l'instruction utilisant le méthode `set_nom` permettant de changer le nom de **medor** en "Mais-dort".
6. On suppose qu'on a importé le module **time** de Python, l'expression `time.gmtime().tm_year` renvoie alors l'année en cours. Ecrire la méthode **age**, qui renvoie l'âge approximatif d'un chien (on considère qu'un chien né en 2017 a 4 ans en 2021 sans s'occuper du jour de naissance)

□ Exercice 2 : Elève

10 points

On souhaite créer une classe **Eleve** qui modélise un élève suivant l'enseignement de spécialité NSI en terminale. Cette classe possède les attributs suivants :

- **nom** : une chaîne de caractère contenant le prénom et le nom de l'élève
- **spe2** : le nom du deuxième enseignement de spécialité suivi par l'élève
- **notes** : la liste des notes obtenues par l'élève en NSI durant l'année (cette liste est vide à la création)

1. Ecrire la classe **Eleve** et son constructeur.
2. Ecrire la méthode **ajoute_note** qui prend en argument un entier **note** et permet d'ajouter la note **note** à la liste des notes obtenu par un objet de la classe **Eleve**.
3. Compléter le programme python suivant afin que la variable **jean_untel** représente l'élève "Jean Untel" qui suit aussi l'enseignement de spécialité "SES". Et que sa liste de notes soit [12, 15, 16, 8, 11, 9, 13].

```

1     jean_untel = .....
2     liste_notes = [12, 15, 16, 8, 11, 9, 13]
3     for note in liste_notes:
4         .....
```

4. Recopier et compléter la méthode suivante de la classe **Eleve** qui permet de renvoyer la moyenne des notes qu'il a obtenu.

```

1     def moyenne(self):
2         somme_note = 0
3         for note in .....:
4             somme_note .....
5         moyenne = somme_note / .....
6         return moyenne
```

5. Corriger cette méthode de façon à ce qu'elle renvoie **None** lorsque la liste des notes de l'élève est vide.
6. Réécrire cette méthode en utilisant un parcours par indice de la liste de notes de l'élève.