

Ex 1: dictionnaire: créer et affecter un couple cle:valeur

1. Créer un dictionnaire D qui contient les noms et âges de trois personnes. Utiliser le prénom pour clé et l'âge comme valeur. Placer directement les valeurs dans le dictionnaire, à sa création. Utiliser les informations suivantes:

```
1. nom = ["Alice", "Bob", "Charlie"]
2. age = [30, 25, 35]
3. D = {
```

2. Faire la même chose qu'au 1., mais cette fois, après avoir créé un dictionnaire D, vide. Compléter:

```
1. D = {}
2. ..
3. ..
4. ..
```

3. accéder et afficher leur âge. Utiliser print

```
> print(D[..  
>  
>
```

4. Ajoute "David": 40, puis change l'âge de "Alice" à 32:

```
> ..
```

Ex 2: dictionnaire: modifier une valeur

Le script suivant va modifier le score du joueur 1, en ajoutant 1 à sa valeur, stockée dans le dictionnaire:

```
1. scores = {"joueur1":0, "joueur2":0}
2. scores["joueur1"] += 1
```

1. Quel est le contenu de scores après ces 2 lignes?
2. Compléter programme suivant, qui ajoute +1 au joueur qui remporte la manche:

```
1. scores = {"joueur1":0, "joueur2":0}
2. while True:
3.     gagnant = input("Quel joueur remporte la manche?")
4.     if .. ..
5.         .. ..
6.     else:
7.         .. ..
```

Ex 3: Nombre d'occurrences

1. Le programme suivant va compter le nombre de caractères "s" dans le mot "assassins"

```
1. lettre = "s"
2. mot = "assassins"
3. score = 0
4. for c in mot:
5.     if c == lettre:
6.         score = score + 1
```

Que vaut la variable score à la fin du programme?

2. Compléter la fonction qui retourne le nombre d'occurrences de `lettre` dans `mot`:

```
1. def compter_lettre(mot, lettre):
2.     score = 0
3.     for c in mot:
4.         ..
5.         ..
6.     return ..
```

2. Quelle instruction, utilisant la fonction `compter_lettre` va retourner le nombre caractères "s" dans le mot "assassins"?

>

Ex 4: dictionnaire: Nombre d'occurrences

1. Compléter la fonction qui retourne un dictionnaire contenant le nombre d'occurrences de chaque lettre dans une chaîne.

```
1. def compter_lettres(chaine):
2.     dico = {}
3.     for lettre in chaine:
4.         if lettre in dico:
5.             ...
6.         else:
7.             ...
8.     return dico
```

2. Que retourne l'instruction suivante?

```
> print(compter_lettres("bonjour"))
```

Ex 3: Données en table

On considère les données des étudiants mises dans une liste de dictionnaires:

```
1. etudiants = [  
2.     {"nom": "Alice", "age": 21, "ville": "Paris"},  
3.     {"nom": "Bob", "age": 22, "ville": "Lyon"},  
4.     {"nom": "Charlie", "age": 23, "ville": "Marseille"}  
5. ]
```

1. Compléter le script qui donne le nom de tous les étudiants

```
1. for etudiant in etudiants:  
2.     print(etudiant[...])
```

2. afficher l'âge à l'aide d'une boucle for:

```
1. for etudiant in etudiants:  
2.     ...
```

3. Afficher le nom des étudiants qui habitent Lyon (s'il y en a):

```
1. for e in etudiants:  
2.     if e["ville"] == "Lyon":  
3.         print(e["nom"])
```

4. Affiche toutes les personnes avec leur âge. On utilisera l'expression formatée

```
> print("{} : {} ans".format(nom, age))  
  
1.  
2.  
3.
```

5. Comment est modifiée la liste etudiants avec l'instruction suivante?

```
> etudiants[0]["age"] = 22
```

6. Le script suivant permet de calculer l'âge moyen des étudiants. Complétez le:

```
1. total_age = 0  
2. for e in etudiants:  
3.     total_age += e["age"]  
4.  
5. moyenne = total_age / len(etudiants)  
6. print("Moyenne d'âge :", moyenne)
```