

Nom:

Prénom:

Exercice 1 : Indiquer ce qui est affiché par les instructions suivantes :

</> Code Python

```
dico = {'nom': 'Dupont', 'age': 30, 'ville': 'Paris', 'enfants': ['Alice', 'Bob']}
```

Instruction	Affichage	Instruction	Affichage
dico['nom']		len(dico)	
dico['age']		dico['enfants'][0]	
dico['ville']		dico['enfants']	

Exercice 2 : On considère le dictionnaire suivant :

</> Code Python

```
dico = {'a': 1, 'b': 2, 'c': 3}
```

Ecrire les instructions permettant

1. d'ajouter une nouvelle entrée 'd' avec la valeur 4.
2. de modifier la valeur associée à la clé 'b' pour qu'elle soit égale à 5.
3. de récupérer la valeur associée à la clé 'c'.
4. de vérifier si la clé 'a' est présente dans le dictionnaire.
5. de supprimer l'entrée associée à la clé 'a'.

Exercice 3 : On considère la liste suivante :

</> Code Python

```
L = ['Alice', 'Bob', 'Charlie', 'David']
```

Ecrire une fonction longueur_mot(L) qui renvoie un dictionnaire dont :

- les clés sont les éléments de la liste L
- les valeurs sont les longueurs des éléments de la liste L

</> Code Python

```
def longueur_mot(L):
```



```
assert longueur_mot(L) == {'Alice': 5, 'Bob': 3, 'Charlie': 7, 'David': 5}
```

Exercice 4 :

Dans cet exercice, on gère une petite guilde de guerriers. Le dictionnaire guilde contient le nom des joueurs en clés et leur niveau en valeurs.

Code Python

```
gilde = {
    "Aethelgard": 42,
    "Slythe": 15,
    "Valeria": 38,
    "Groggar": 10,
    "Lyra": 25
}
```

1. Ecrire une fonction `liste_de_noms(guild)` qui renvoie la liste des noms des joueurs.

Code Python

```
def liste_de_noms(guild):
```

```
assert liste_de_noms(guild) == ["Aethelgard", "Slythe", "Valeria",  
                                "Grognar", "Lyra"]
```

2. Ecrire une fonction `niveau_moyen(guild)` qui renvoie le niveau moyen des joueurs de la guild.

Code Python

```
def niveau_moyen(guild):
```

```
assert niveau_moyen(guild) == 26.0
```

3. Ecrire une fonction `listes_elite(guild)` qui renvoie un dictionnaire contenant uniquement les joueurs de niveau supérieur ou égal à 30.

Code Python

```
def listes_elite(guild):
```

```
assert listes_elite(guild) == {"Aethelgard": 42, "Valeria": 38}
```

Exercice 5 :**Algorithme de Fréquences**

Cet algorithme classique permet de compter les occurrences de chaque caractère dans une chaîne. Complétez les zones vides.

</> Code Python

```

texte = "nsi"
frequences = {}

for caractere in texte:
    if caractere in .....:
        frequences[caractere] = frequences[caractere] + 1
    else:
        frequences[caractere] = .....

print(frequences) # Doit afficher {'n': 1, 's': 1, 'i': 1}

```

Exercice 6 :

Structures Complexes : Bibliothèque : On utilise une liste de dictionnaires pour gérer des livres :

</> Code Python

```

biblio = [
    {"titre": "1984", "auteur": "Orwell", "disponible": True},
    {"titre": "Dune", "auteur": "Herbert", "disponible": False},
    {"titre": "Fondation", "auteur": "Asimov", "disponible": True}
]

```

1. Quelle instruction permet d'accéder au dictionnaire du livre "Dune" ?
2. Que renvoie l'instruction `biblio[2]["auteur"]` ?
3. **Écriture de fonction :** Complétez la fonction ci-dessous qui renvoie True si un livre est disponible, et False sinon.

</> Code Python

```

def est_dispo(titre_recherche, table_livres):
    for livre in table_livres:
        if livre["titre"] == .....:
            return .....["disponible"]
    return "Livre non reference"

```