

Nom:

Prénom:

Exercice 1 ►**Compréhension et Vocabulaire:** Un développeur utilise un dictionnaire pour stocker l'inventaire d'un joueur :

```
inventaire = {"potions": 5, "pieces": 150, "epee": 1}
```

1. **Type et Terminologie** : Quel est le type de la variable `inventaire` ? Dans le couple `"pieces": 150`, identifiez la clé et la valeur.
2. **Accès** : Quelle instruction permet d'afficher le nombre de "potions" ?
3. **Mise à jour** : Le joueur achète une potion supplémentaire. Écrivez l'instruction qui met à jour la quantité de potions à 6.
4. **Ajout** : Le joueur ramasse un nouvel objet : "bouclier" (quantité 1). Écrivez l'instruction permettant d'ajouter cet élément au dictionnaire.

Exercice 2 ►**Parcours et Filtrage:** On considère un dictionnaire de produits et leurs prix en euros :

```
catalogue = {"Clavier": 45, "Souris": 25, "Ecran": 180, "Tapis": 15}
```

Complétez le script Python suivant pour qu'il affiche uniquement les **noms** des produits dont le prix est **inférieur à 30 euros**.

```
for produit, prix in catalogue.....:
    if ..... < 30:
        print(.....)
```

Exercice 3 ►**Algorithme de Fréquences**

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

```
texte = "nsi"
fréquences = {}

for caractère in texte:
    if caractère in ....:
        fréquences[caractère] = fréquences[caractère] + 1
    else:
        fréquences[caractère] = .....

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

Exercice 4 ►**Structures Complexes : Bibliothèque:** On utilise une liste de dictionnaires pour gérer des livres :

```
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.

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