Révisions Python (fonctions et dictionnaires)

Python

Exercice 1 *QCM*

Pour chaque question, une seule réponse est correcte.

1. Que renvoie la fonction suivante pour `mystere(3, 5)` ?

```python
def mystere(a, b=10):
    return a + b
```

   A. 8

   B. 13

   C. 15

   D. Une erreur

2. Quel est le résultat de ce code ?

```python
x = 10
def g():
    x = 5
    return x
print(f"{g()} {x}")
```

   A. `5 5`

   B. `5 10`

   C. `10 10`

   D. `10 5`

3. Quelle est la complexité de la recherche d'une clé dans un dictionnaire Python ?

   A. $O(n)$

   B. $O(\log n)$

   C. $O(1)$ en moyenne

   D. $O(n^2)$

4. Que produit le code suivant ?

```python
d = {'a': 1, 'b': 2, 'c': 3}
d['b'] = 5
d['d'] = 4
print(len(d))
```

A. 3

B. 4

C. 5

D. Une erreur

Exercice 2 *Fonctions avec assertions et paramètres*

a) Écrire une fonction `moyenne(L)` qui calcule la moyenne d'une liste de nombres. Ajouter une assertion vérifiant que la liste n'est pas vide et une docstring avec un exemple.

b) Écrire une fonction `saluer(prenom, formule="Bonjour")` qui renvoie une chaîne de la forme `"Bonjour Marie"`. Que renvoient les appels suivants ?

```
saluer("Alice")
saluer("Bob", "Salut")
saluer(formule="Bonsoir", prenom="Clara")
```

c) Écrire une fonction `clamp(x, mini=0, maxi=100)` qui renvoie `x` si `mini <= x <= maxi`, sinon `mini` ou `maxi` selon le cas. Ajouter une assertion vérifiant que `mini <= maxi`.

d) Écrire une fonction `presentation(*args)` qui accepte un nombre variable de prénoms et renvoie `"Bonjour Alice, Bob et Clara"` (avec une virgule entre les prénoms sauf avant le dernier, précédé de « et »).

Exercice 3 *Dictionnaires – manipulation de base*

On donne le dictionnaire suivant :

```
eleves = {
    "Alice": {"age": 17, "moyenne": 14.5},
    "Bob": {"age": 16, "moyenne": 12.0},
    "Clara": {"age": 17, "moyenne": 16.5},
    "David": {"age": 16, "moyenne": 9.5}
}
```

a) Écrire une fonction `moyenne_classe(eleves)` qui calcule la moyenne générale de la classe.

b) Écrire une fonction `meilleur_eleve(eleves)` qui renvoie le nom de l'élève ayant la meilleure moyenne.

c) Écrire une fonction `eleves_par_age(eleves, age)` qui renvoie la liste des noms des élèves ayant l'âge donné.

d) Ajouter l'élève « Emma », 17 ans, moyenne 15,0 au dictionnaire et recalculer la moyenne de la classe.

Exercice 4 *Compréhensions de dictionnaires*

a) Créer par compréhension un dictionnaire associant à chaque entier de 1 à 10 son carré.

b) À partir du dictionnaire `{"a": 1, "b": 2, "c": 3, "d": 4}`, créer par compréhension un nouveau dictionnaire ne contenant que les paires dont la valeur est paire.

c) Écrire une fonction `inverser(d)` qui renvoie un dictionnaire où les clés et valeurs sont échangées. Que se passe-t-il si deux clés ont la même valeur ?

Exercice 5 *Comptage et fréquences*

a) Écrire une fonction `compter_lettres(texte)` qui renvoie un dictionnaire associant à chaque lettre (en minuscules) son nombre d'occurrences dans le texte.

b) Écrire une fonction `lettre_frequente(texte)` qui renvoie la lettre la plus fréquente.

c) Écrire une fonction `compter_mots(phrase)` qui renvoie un dictionnaire associant à chaque mot son nombre d'occurrences.

Exercice 6 *Fonctions et dictionnaires combinés*

a) Écrire une fonction `statistiques(L)` qui prend une liste de nombres et renvoie un dictionnaire contenant les clés `"min"`, `"max"`, `"moyenne"` et `"nb_elements"` avec les valeurs correspondantes.

b) Écrire une fonction `fusionner(d1, d2)` qui prend deux dictionnaires et renvoie un nouveau dictionnaire contenant toutes les paires clé-valeur des deux. Si une clé existe dans les deux, la valeur de `d2` est conservée. Ne pas modifier `d1` ni `d2`.

c) Écrire une fonction `appliquer_remise(produits, remise=0.1)` qui prend un dictionnaire `{nom: prix}` et renvoie un nouveau dictionnaire avec les prix réduits du pourcentage indiqué. La fonction ne doit pas modifier le dictionnaire original.

Exercice 7 *Synthèse – carnet de contacts*

On souhaite créer un carnet de contacts sous forme de dictionnaire de dictionnaires.

a) Écrire une fonction `ajouter_contact(carnet, nom, telephone, email="")` qui ajoute un contact au carnet. Si le contact existe déjà, ses informations sont mises à jour.

b) Écrire une fonction `rechercher(carnet, nom)` qui renvoie les informations du contact ou un message indiquant qu'il n'existe pas (en utilisant `.get()`).

c) Écrire une fonction `supprimer_contact(carnet, nom)` qui supprime un contact s'il existe, et affiche un message sinon.

d) Écrire une fonction `exporter(carnet)` qui renvoie une liste de chaînes formatées `"Nom : tel - email"`, triée par nom.

e) Tester toutes ces fonctions avec un carnet contenant au moins quatre contacts.