Dictionnaires

Exercices

Exercice 0 *QCM – vérification des prérequis* Pour chaque question, une seule réponse est correcte.

1. Quelle structure Python associe des clés à des valeurs ?

— A. Une liste

— B. Un tuple

— C. Un dictionnaire

— D. Un ensemble (set)

2. Comment accéder à la valeur associée à la clé `'nom'` dans `d = {'nom': 'Alice', 'age': 17}` ?

— A. `d[0]`

— B. `d['nom']`

— C. `d.nom`

— D. `d(nom)`

3. Que se passe-t-il si on écrit `d['note'] = 15` avec `d = {'nom': 'Alice'}` ?

— A. Erreur car la clé `'note'` n'existe pas

— B. La paire `'note': 15` est ajoutée au dictionnaire

— C. Le dictionnaire est remplacé par `{'note': 15}`

— D. Rien ne se passe

4. Quels types Python peuvent servir de clé dans un dictionnaire ?

— A. Uniquement les chaînes de caractères

— B. Uniquement les entiers

— C. Tout type immuable (int, str, tuple, etc.)

— D. N'importe quel type, y compris les listes

Exercice 1 *Manipulation d'un dictionnaire* Soit le dictionnaire :

```
1 d = {'nom': 'Dupuis', 'prenom': 'Jacque', 'age': 30}
```

— Corriger l'erreur dans le prénom `'Jacques'` à l'aide d'une instruction.

— Afficher la liste des clés du dictionnaire.

— Afficher la liste des valeurs du dictionnaire.

— Afficher la liste des paires clé/valeur du dictionnaire.

— Écrire une instruction qui affiche la phrase « Jacques Dupuis a 30 ans ».

Exercice 2 *Dictionnaire d'élèves*

1. Créer un dictionnaire `eleves` associant des prénoms et des âges (pour environ cinq personnes).
2. Donner deux façons différentes d'itérer sur le dictionnaire et d'afficher les paires sous le format `"prenom a age ans"`.

Exercice 3 *Fréquence des mots* Écrire une fonction qui prend en entrée une chaîne de caractères et retourne un dictionnaire indiquant, pour chaque mot, sa fréquence dans la chaîne fournie.

Exercice 4 *Ventes* On dispose d'un dictionnaire associant à des noms de commerciaux d'une société le nombre de ventes qu'ils ont réalisées. Par exemple :

```
ventes = {"Dupont": 14, "Hervy": 19, "Geoffroy": 15, "Layec": 21}
```

1. Écrire une fonction qui prend en entrée un tel dictionnaire et renvoie le nombre total de ventes dans la société.
2. Écrire une fonction qui prend en entrée un tel dictionnaire et renvoie le nom du vendeur ayant réalisé le plus de ventes. Si plusieurs vendeurs sont ex-aequo sur ce critère, la fonction devra retourner le nom de l'un d'entre eux.

Exercice 5 *Conversion dictionnaire/liste*

1. Écrire une fonction `dico_vers_liste` qui convertit un dictionnaire en une liste de tuples (sans utiliser la fonction `items()`).
2. Écrire une fonction `liste_vers_dico` qui convertit une liste de tuples en dictionnaire.

Exercice 6 *Étudiants* Écrire une fonction qui prend en entrée une chaîne de caractères comprenant, sur chaque ligne, trois champs séparés par des caractères `';'` (un numéro d'étudiant, un nom et un prénom) et renvoie un dictionnaire dont les clés sont les numéros d'étudiants et les valeurs sont, pour chaque numéro d'étudiant, une chaîne correspondant à la concaténation des prénom et nom de la personne. On pourra tester la fonction avec la chaîne suivante :

```
chaine_etudiants = """213615200;BESNIER;JEAN
213565488;DUPOND;MARC
214665555;DURAND;JULIE"""
```

Exercice 7 *Meilleures moyennes* Écrire une fonction qui prend en entrée un dictionnaire associant à un nom une liste de notes et qui renvoie la liste des noms des personnes qui ont la moyenne la plus élevée (s'il y a des ex-aequo, cette liste contiendra plusieurs éléments, sinon, elle n'en contiendra qu'un) et la moyenne correspondante. On pourra utiliser le dictionnaire suivant pour tester :

```
notes = {"Tom": [8, 10, 12], "Mila": [10, 9], "Alex": [], "Lina": [12, 10, 8]}
```

Exercice 8 *Salles du lycée* Le dictionnaire suivant décrit quelques salles d'un lycée. La clé est le nom de la salle ; la valeur est un tuple qui contient, dans cet ordre, le bâtiment, le nombre de places et un booléen qui indique si la salle a un vidéoprojecteur.

```python
salles = {
    "A101": ("A", 30, True),
    "A102": ("A", 36, False),
    "B12": ("B", 24, True),
    "B14": ("B", 35, True),
    "C3": ("C", 18, False),
    "Amphi": ("C", 120, True),
}
```

Par exemple, `salles["B14"]` vaut `('B', 35, True)`.

1. Écrire un programme qui affiche la liste des bâtiments, chacun une seule fois.
2. Une classe de 35 élèves doit voir une vidéo. Écrire un programme qui affiche le nom de chaque salle qui a au moins 35 places **et** un vidéoprojecteur.
3. Écrire une fonction `salles_batiment(batiment)` qui renvoie la liste des noms des salles situées dans ce bâtiment. Par exemple, `salles_batiment("B")` renvoie `["B12", "B14"]`.
4. Écrire une fonction `places_batiment(batiment)` qui renvoie le nombre total de places des salles de ce bâtiment. Par exemple, `places_batiment("A")` renvoie `66`.

Exercice 9 *Le foyer des lycéens* À la récréation, le foyer des lycéens vend quelques en-cas. Les tarifs sont rangés dans un dictionnaire `tarifs` : chaque clé est le nom d'un produit (`str`), et la valeur associée son prix **en centimes** (`int`). Compter en centimes évite les erreurs d'arrondi des flottants : `0.1 + 0.2` ne vaut pas exactement `0.3`, alors que `10 + 20` vaut exactement `30`.

1. Écrire l'instruction qui crée le dictionnaire `tarifs` correspondant à cette table :

Produit	Prix (centimes)
Croissant	110
Pain au chocolat	120
Jus d'orange	150
Compote	80
Barre de céréales	90
Eau	50

2. Écrire une fonction `en_vente(produit, tarifs)` qui renvoie `True` si `produit` figure dans `tarifs`, et `False` sinon.
3. Écrire une fonction `prix_moyen(tarifs)` qui renvoie le prix moyen, en centimes, des produits vendus (on suppose `tarifs` non vide). Avec la table ci-dessus, elle renvoie `100.0`.
4. Écrire une fonction `entre(mini, maxi, tarifs)` qui renvoie la liste des produits dont le prix est compris entre `mini` et `maxi`, bornes incluses. Par exemple, `entre(85, 130, tarifs)` renvoie `['Croissant', 'Pain au chocolat', 'Barre de céréales']`.
5. Une commande est un dictionnaire dont les clés sont des noms de produits et les valeurs des nombres d'exemplaires. Écrire le dictionnaire `commande` d'un élève qui demande deux croissants, un jus d'orange et trois compotes.
6. Écrire une fonction `commande_possible(commande, tarifs)` qui renvoie `True` si tous les produits de la commande sont en vente, et `False` sinon.

7. Écrire une fonction `montant(commande, tarifs)` qui renvoie le prix total de la commande, en centimes (on suppose tous les produits en vente). Pour la commande de la question 5, elle renvoie `610`, soit 6,10 €.