

Types construits



Exercice 1 (Vrai/Faux)

Pour chacune des affirmations suivantes, dire si elle est vraie ou fausse. Justifier si possible la réponse.

1. Le n-uplet $(1, 2, (3, 4), 5)$ a pour longueur 5.
2. Si $t = 'a', 'd', 'c'$, l'instruction $t[1] = 'b'$ provoque une erreur.
3. l'expression $3*(1, 2)$ a pour valeur $(3, 6)$.
4. Soit L la liste $[1, 2, 3]$. Après l'instruction $L.append([4, 5])$, la liste L a pour valeur $[1, 2, 3, 4, 5]$.
5. Après l'instruction :

```
L=[3*x for x in [1,2,3] if x%2!=0]
```

la liste L a pour valeur $[3, 9]$.

6. Soit `nombres` une liste de nombres. Pour modifier le dernier élément, on écrit :
`nombres(len(nombres)) = ...`
7. Soit d un dictionnaire vide. L'instruction $v = d['un']$ provoque une erreur.
8. Soit d un dictionnaire. L'instruction $d[(10, 15)] = 5$ provoque une erreur.
9. Soit d un dictionnaire. L'instruction $d[[10, 15]] = 5$ provoque une erreur.

Exercice 2 (QCM)

Pour chaque question, une seule réponse est valable. Indiquer laquelle.

1. On considère le n-uplet $t = (3, 5, 1)$. Qu'obtient-on après l'instruction $t[1] = 4$?
 - (a) La valeur de t est $t = (4, 5, 1)$.
 - (b) La valeur de t est $t = (3, 4, 5)$.
 - (c) La valeur de t est $t = (3, 5, 4)$.
 - (d) Une erreur
2. Après l'instruction $a = 0, 2+0, 3==0, 5$, qu'obtient-on ?
 - (a) a vaut `False` (à cause des flottants).
 - (b) a vaut `True` (car l'égalité est bien vraie dans Python).
 - (c) a vaut $(0, 2, False, 5)$.
 - (d) Une erreur
3. Soit la liste $L = [15, 17, 12, 23]$. Après l'instruction $L[2] = 25$, la liste L vaut :
 - (a) $[15, 17, 25, 23]$
 - (b) $[15, 25, 12, 23]$
 - (c) $[15, 17, 25, 12, 23]$
 - (d) $[15, 25, 17, 12, 23]$
4. On dispose de la liste $L = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]$. La valeur de $L[1][2]$ est :

- (a) 2 (b) 4 (c) 6 (d) 8

5. Après l'instruction `L=[[i,i+1] for i in range(2)]`, la valeur de L est :

- (a) `[[0,1],[1,2]]` (c) `[0,1,1,2]`
(b) `[[1,2],[2,3]]` (d) `[1,2,2,3]`

6. On construit une matrice avec le code suivant :

```
matrice = [3*[0] for i in range(3)]
for i in range(3):
    matrice[i][i]=i+1
    matrice[0][i]=matrice[0][i]+i+1
    matrice[i][2]=matrice[i][2]+i+1
```

Quel est le résultat obtenu ?

- (a) `[[1,2,4],[0,2,2],[0,0,6]]` (c) `[[1,2,4],[0,2,4],[0,0,6]]`
(b) `[[2,2,4],[0,2,2],[0,0,4]]` (d) `[[2,2,4],[0,2,2],[0,0,6]]`

7. On considère le code Python suivant :

```
def f():
    liste1.append(2)
    liste2=liste1+[3]
liste1=[0,1]
liste2=[0,1]
f()
```

Quel est le résultat obtenu ?

- (a) La liste `liste1` vaut `[0,1,2]` et la liste `liste2` vaut `[0,1,2,3]`.
(b) La liste `liste1` vaut `[0,1,2]` et la liste `liste2` vaut `[0,1,3]`.
(c) La liste `liste1` vaut `[0,1,2]` et la liste `liste2` vaut `[0,1]`.
(d) La liste `liste1` vaut `[0,1]` et la liste `liste2` vaut `[0,1]`.

8. Parmi les termes suivants, lequel n'est pas une méthode d'un dictionnaire ?

- (a) `data` (b) `items` (c) `keys` (d) `values`

9. On considère le code suivant :

```
d={"if":"si","yes":"oui","no":"non"}
for c in d:
    print(c)
```

Qu'obtient-on ?

- (a) L'affichage de `if`, `yes` et `no`.
(b) L'affichage de `si`, `oui` et `non`.
(c) L'affichage des couples `('if','si')`, `('yes','oui')` et `('no','non')`.
(d) Une erreur.

Exercice 3

Écrire un fichier python qui effectue les actions suivantes (successivement) :

- Définir la liste `liste` contenant les éléments suivants dans l'ordre : 45,17,89,38,10 et 74.
- Trier `liste` puis l'afficher.
- Ajouter l'élément « 12 » puis afficher la liste.
- Renverser la liste puis l'afficher.
- Afficher l'indice de l'élément « 10 ».
- Enlever l'élément « 38 » et afficher la liste.
- Afficher la sous-liste du 2^e au 3^e élément.
- Afficher la sous-liste du début au 2^e élément.
- Afficher la sous-liste du 3^e élément au dernier.
- Afficher le deuxième élément en partant de la fin.

Exercice 4

- Définir une fonction Python prenant comme argument un entier naturel n et qui renvoie une liste de la forme : `[1,1,1,2,1,3,1,4,1,5,1,6,...,1,n]`.
- Définir une fonction Python prenant comme argument un entier naturel n et qui construit une liste de la forme : `[1,1,2,1,2,3,1,2,3,4,...,1,2,3,..., n]`.

Exercice 5

Écrire un programme Python qui :

- Définir une liste `liste` contenant tous les entiers de 1 à 10 (en utilisant la fonction `range`). On pourra chercher plusieurs manières de le faire.
- Calcule et affiche la somme des éléments de `liste` et stocke le résultat dans une variable `somme` (sans utiliser la fonction `sum` de Python).
- Calcule et affiche la moyenne de ces nombres (en utilisant la variable définie précédemment).

Exercice 6

Le Scrabble est un jeu de société où l'on doit former des mots avec un tirage aléatoire des lettres, chaque lettre valant un certain nombre de points. Le tableau suivant donne la valeur de certaines lettres :

Lettre	A	B	C	D	E	F	G	H	I	J	K
Valeur	1	3	3	2	1	4	2	4	1	8	5

- Définir un dictionnaire dont les clés sont les lettres et les valeurs sont la valeur des lettres au Scrabble indiquées par le tableau ci-dessus.
- Définir une fonction qui :
 - Demande un mot (en utilisant la fonction `input` qui retourne un élément de type `str`).
 - retourne la valeur totale du mot (on comptera 0 pour une lettre absente du dictionnaire).
Attention à transformer le mot donné en majuscules. Pour cela, rechercher dans la librairie Python les fonctions et méthodes associées aux chaînes de caractères.

Exercice 7

Le dictionnaire suivant indique les valeurs des lettres dans les règles françaises du Scrabble :

```
valeurs_Scrabble = {10: 'KWXYZ', 8: 'JQ', 4: 'FHV', 3: 'BCP', 2: 'DMG'}
```

sachant que les lettres qui manquent ont pour valeur 1.

- Définir à partir de `valeurs_Scrabble` un dictionnaire similaire à celui de l'exercice 6, c'est à dire dont les clés sont les lettres et les valeurs sont leur valeur au Scrabble, sans ajouter les lettres de valeur 1.
- Redéfinir la fonction de l'exercice 6, mais en comptant cette fois 1 pour les lettres absentes du dictionnaire.

Exercice 8

On a construit un dictionnaire ayant pour clés des couples contenant les coordonnées GPS de villes et pour valeurs les villes correspondantes. On trouve les coordonnées sur Internet par exemple. Les données sont sous forme décimale en degré. On a par exemple :

```
positions = {}
positions[(48.853585, 2.301490)] = "Paris"
positions[(11.611358, 43.147752)] = "Djibouti"
positions[(37.023113, -8.996601)] = "Fortaleza de São Vicente"
positions[(7.677989, -5.025387)] = "Bouaké"
# etc.
```

Nous supposons avoir reçu une photo prise avec un smartphone par une personne en vacances. Nous voyons dans les propriétés les coordonnées GPS au moment de la prise de vue. Écrire une fonction prenant en paramètres un couple de coordonnées GPS et le dictionnaire construit et renvoyant le nom du lieu correspondant, ou `None` si aucune ville du dictionnaire ne correspond. On tolère une précision au dix-millième de degré.

Par exemple, si les coordonnées sont (37.02311, -8.9966), la fonction doit nous renvoyer :
"Fortaleza de São Vicente".

Exercice 9 (mini-projet de groupe)

Écrire un programme Python permettant de gérer un répertoire téléphonique sous forme d'un dictionnaire. Un premier répertoire contenant des noms et numéros de téléphone sera défini au préalable dans le code Python. Le programme doit permettre, lorsque l'on exécute une fonction `main()`, d'ajouter, modifier, supprimer une entrée, mais également de faire des recherches (par nom, par numéro). Ces actions seront faites à partir d'un « menu » affiché, où le choix est demandé à l'utilisateur, y compris celui de quitter le programme, et qui revient en boucle.

Pour aller plus loin, on pourra faire en sorte de sauvegarder le répertoire dans un fichier, de le charger en mémoire.