

Les dictionnaires Python

1 Définition

1.1 Définition

Comme avec les tableaux dynamiques (type list en Python), les dictionnaires sont une structure linéaire qui permet de "stocker" des données.

Chaque élément d'un dictionnaire est composé de deux parties :

- La clé
- La valeur

⚠ A l'inverse des tableaux, les dictionnaires ne sont pas indexés, c'est à dire que les valeurs ne sont pas repérées par un indice.

Définition 3.1

Un dictionnaire est une structure de données où les différentes valeurs sont associées à des clés.

Le couple (clé-valeur) est appelé une entrée.

Les différentes valeurs, à la différence d'un tableau, ne sont pas indexés, mais associées à des clefs.

☞ L'accès aux valeurs se fait en utilisant des fonctions de hachage, très utilisé en informatique, mais hors programme dans le cadre de la NSI.

1.2 Mon premier dictionnaire

La structure des dictionnaires est implémenté nativement dans python. C'est un objet qui possède des méthodes.

```
mon_dico_vide = {} # ou bien mon_dico_vide= dict()  
mon_dico = {"Pierre": "0689787475", "Romane": "0656585978"}
```

Remarque

On utilise des accolades pour définir un dictionnaire

Dans l'exemple précédent, "Pierre" et "Romane" sont des clés.
Tandis que "0689787475" et "0656585978" sont des valeurs.

Les clés peuvent être des nombres, des chaînes de caractères, des tuples.
Les valeurs peuvent être des nombres, des chaînes de caractères, des tuples, des listes..etc..

1.3 Ajout d'un élément

```
mon_dico["Aline"] = "0789868583"
```

1.4 Accès à la valeur associée à une clé

```
mon_dico["Pierre"]
```

Remarque

Il est aussi possible d'accéder à une valeur du dictionnaire en utilisant la méthode `get()`, mais nous étudierons cela davantage en terminale.

1.5 Effacer une entrée

```
del mon_dico["Pierre"]
```

⚠ Cela efface la clé et la valeur associée.

● Exercice 3.1

Soit le dictionnaire :

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

1. (a) Afficher le nom 'Dupuis'
(b) puis le prénom 'Jacque',
(c) et enfin l'âge '30'
2. Corriger l'erreur dans le prénom, la bonne valeur est 'Jacques'.
3. Ecrire la phrase "Jacques Dupuis a 30 ans".

2 Les méthodes associées au dictionnaires

Comme tout objet, les dictionnaires possèdent différentes méthodes. En première, nous étudierons uniquement 3 méthodes, qui permettent de parcourir un dictionnaires (avec un `for` par exemple).

2.1 La méthode `dico.keys()`

```
>>> ani2 = {'nom': 'singe', 'poids': 70, 'taille': 1.75}  
>>> for key in ani2.keys():  
    print(key)
```

2.2 La méthode dico.values()

```
>>> ani2 = {'nom':'singe', 'poids':70, 'taille':1.75}
>>> for val in ani2.values():
    print(val)
```

2.3 La méthode dico.items()

```
>>> ani2 = {'nom':'singe', 'poids':70, 'taille':1.75}
>>> for key,val in ani2.items():
    print (key,val)
```

Exercice 3.2

Soit le dictionnaire :

```
>>> d = {'nom': 'Dupuis', 'prenom': 'Jacques', 'age': 30}
```

1. Afficher la liste des clés du dictionnaire.
2. Afficher la liste des valeurs du dictionnaire.
3. Afficher la liste des entrées clé/valeur du dictionnaire.

Exercice 3.3

Dans cet exercice, nous nous familiarisons avec les manipulations de dictionnaires sur une thématique de magasin en ligne.

«*Chez Geek and sons, tout ce qui est inutile peut s'acheter, et tout ce qui peut s'acheter est un peu trop cher.*»

La base de prix des produits de Geek and sons est représentée en Python par un dictionnaire de type dict avec : — les noms de produits, de type str, comme clés— les prix des produits, de type float, comme valeurs associées.

1. Donner une expression Python pour construire la base des prix des produits correspondant à la table suivante :

| Nom du produit | Prix TTC |
|---------------------|----------|
| Sabre laser | 229 |
| Mitendo DX127 | 30 |
| Coussin Linux | 74.5 |
| Slip Goldorak | 30 |
| Station Nextpresso1 | 84.60 |

2. Créer une fonction **prix_moyen(dico)** qui prend en argument un dictionnaire (où les clefs sont des str, et les valeurs des float) et qui retourne le prix moyen des produits disponibles.
3. Créer une fonction **fourchette_prix(dic,mini,max)** qui, étant donné un prix minimum mini, un prix maximum maxi et une base de Prix, retourne l'ensemble des noms de produits disponibles dans cette fourchette de prix.
4. La notion de panier est un concept omniprésent dans les sites marchands, Geeks and sons n'échappe pas à la règle. En Python, le panier du client sera représenté par un dictionnaire de type dict avec :
 - les noms de produits comme clés
 - une quantité d'achat comme valeurs associées

Donner une expression Python correspondant à l'achat de 3 sabres lasers, de 2 coussins Linux et de 1 slip Goldorak.

5. Créer la fonction **prix_achats(panier,base_prix)** qui, étant donné un panier d'achat **Panier** et une base de **Prix**, retourne le prix total correspondant.

Exercice 3.4

Les réponses correctes d'un QCM de NSI sont stockées dans un dictionnaire nommé **reponses_valides**.

Les clés sont des chaînes de caractères de la forme "Q1". Les valeurs possibles sont des chaînes de caractères correspondant aux quatre réponses "a","b","c","d".

Les réponses données par Alice sont stockées dans le dictionnaire **reponses_Alice**.

Lorsqu'Alice n'a pas répondu à une question, il n'y a pas de clef correspondant au nom de l'exercice.

La notation d'un QCM de NSI est la suivante : 3 points par réponse correcte, -1 point par réponse incorrecte et 0 si l'on n'a pas répondu.

```
reponses_valides = {"Q1":"c","Q2":"a","Q3":"d","Q4":"c","Q5":"b"}  
reponses_Alice = {"Q1":"b","Q2":"a","Q3":"d","Q5":"a"}
```

Compléter la fonction **correction_QCM_Alice(reponses_Alice,reponses_valides)** qui, à partir des dictionnaires **reponses_Alice** et **reponses_valides** passées en argument renvoie le nombre de points obtenus au QCM par Alice.