

Python : les dictionnaires:

Comme les listes, les dictionnaires permettent de "stocker" des données. Chaque élément d'un dictionnaire est composé de 2 parties, on parle de pairs "clé/valeur".

Voici un exemple de dictionnaire:

```
mon_dico = {"nom": "Durand", "prenom": "Christophe", "date de naissance": "29/02/1981"}
```

Comme vous pouvez le constater, nous utilisons des accolades pour définir le début et la fin du dictionnaire (alors que nous utilisons des crochets [] pour les listes et les parenthèses pour les tuples). Dans le dictionnaire ci-dessus, "nom", "prenom" et "date de naissance" sont des clés et "Durand", "Christophe" et "29/02/1981" sont des valeurs. La clé "nom" est associée à la valeur "Durand", la clé "prenom" est associée à la valeur "Christophe" et la clé "date de naissance" est associée à la valeur "29/02/1981". Les clés sont des chaînes de caractères ou des nombres. Les valeurs peuvent être des chaînes de caractères, des nombres, des booléens...

Pour créer un dictionnaire, il est aussi possible de procéder comme suit :

```
mon_dico = {}  
mon_dico["nom"] = "Durand"  
mon_dico["prenom"] = "Christophe"  
mon_dico["date de naissance"] = "29/02/1981"
```

Exercice 1: La variable "mon_dico" référence un dictionnaire. Il est possible d'afficher le contenu du dictionnaire référencé par la variable "mon_dico" en saisissant "mon_dico" dans la console. Faites le test après avoir exécuté le programme ci-dessous.

```
mon_dico = {"nom": "Durand", "prenom": "Christophe", "date de naissance": "29/02/1981"}
```

Il est possible d'afficher la valeur associée à une clé :

Exercice 2: Soit le programme suivant :

```
mon_dico = {"nom": "Durand", "prenom": "Christophe", "date de naissance": "29/02/1981"}  
print(f'Bonjour je suis {mon_dico["prenom"]} {mon_dico["nom"]}, je suis né le {mon_dico["date de naissance"]}')'
```

Quel est le résultat attendu après l'exécution de ce programme ? Vérifiez votre réponse.

Il est facile d'ajouter un élément à un dictionnaire (les dictionnaires sont mutables)

Exercice 3: Soit le programme suivant :

```
mon_dico = {"nom": "Durand", "prenom": "Christophe", "date de naissance": "29/02/1981"}
print(f'Bonjour je suis {mon_dico["prenom"]} {mon_dico["nom"]}, je suis né le {mon_dico["date de naissance"]}')
mon_dico['lieu naissance'] = "Bonneville"
print (f'à {mon_dico["lieu naissance"]}')
```

Quel est le résultat attendu après l'exécution de ce programme ? Vérifiez votre réponse.

L'instruction "del" permet de supprimer une paire "clé/valeur"

Exercice 4: Quel est le contenu du dictionnaire référencé par la variable "mes_fruits" après l'exécution du programme ci-dessous ? Vérifiez votre réponse à l'aide de la console.

```
mes_fruits = {"poire": 3, "pomme": 4, "orange": 2}
del mes_fruits["pomme"]
```

Il est possible de modifier une valeur :

Exercice 5: Soit le programme suivant :

```
mes_fruits = {"poire": 3, "pomme": 4, "orange": 2}
mes_fruits["pomme"] = mes_fruits["pomme"] - 1
```

Quel est le contenu du dictionnaire référencé par la variable "mes_fruits" après l'exécution du programme ci-dessus ? Vérifiez votre réponse à l'aide de la console.

Il est possible de parcourir un dictionnaire à l'aide d'une boucle for. Ce parcours peut se faire selon les clés ou les valeurs. Commençons par parcourir les clés à l'aide de la méthode "keys"

Exercice 6: Tester le programme suivant :

```
mes_fruits = {"poire": 3, "pomme": 4, "orange": 2}
print("liste des fruits :")
for fruit in mes_fruits.keys():
    print(fruit)
```

La méthode values() permet de parcourir le dictionnaire selon les valeurs.

Exercice 7: Tester le programme suivant :

```
mes_fruits = {"poire": 3, "pomme": 4, "orange": 2}
for qte in mes_fruits.values():
    print(qte)
```

Enfin, il est possible de parcourir un dictionnaire à la fois sur les clés et les valeurs en utilisant la méthode `item()`.

Exercice 8: Tester le programme suivant :

```
mes_fruits = {"poire": 3, "pomme": 4, "orange": 2}
print ("Stock de fruits :")
for fruit, qte in mes_fruits.items():
    print (f"{fruit} : {qte}")
```

Vous avez sans doute remarqué l'utilisation de deux variables ("fruit" et "qte") au niveau du "for...in"