


Cours	Cours	
BTS SNIR	Python	
2 ^{ème} année	Les tuples	

Dans cette leçon, nous allons nous intéresser à un autre type de données Python composites : les dictionnaires. Nous allons découvrir l'intérêt des dictionnaires et apprendre à manipuler ce nouveau type de données.

Présentation des dictionnaires Python

Les dictionnaires sont un type natif de données Python. Ce type de données peut, de la même façon que les données séquentielles, contenir plusieurs valeurs et chaque valeur va être indexée ce qui signifie qu'un indice unique va être attribué à chaque valeur.

La grande différence entre les données séquentielles et les dictionnaires se situe dans la façon d'indexer les valeurs et dans la nature de l'index. Dans le cas des séquences, les différentes valeurs sont associées à des index numériques commençant à 0.

Les dictionnaires nous laissent une bien plus grande liberté ici puisqu'on va pouvoir choisir nous-mêmes nos clefs (ou index ou indice) et attribuer la clef de notre choix à chaque valeur à partir du moment où cette clef n'est pas déjà utilisée dans ce dictionnaire et où la clef est une valeur immuable. La conséquence de cela est que les valeurs d'un dictionnaire ne sont pas ordonnées à la différence des valeurs d'une séquence.

Création d'un dictionnaire Python

Pour créer un nouveau dictionnaire, nous allons devoir utiliser un couple d'accolades { } et définir les paires `clef : valeur` à l'intérieur des accolades comme ceci :

```
[>>> d = {"nom": "Pierre", "age": 29, "sport" : ["Trail", "Triathlon"]}
[>>> d["nom"]
'Pierre'
[>>> d["age"]
29
[>>> d["sport"]
['Trail', 'Triathlon']
```

Comme vous pouvez le voir dans l'exemple ci-dessus, on utilise une nouvelle fois le syntaxe avec des crochets pour lire les différentes valeurs de notre dictionnaire. Pour être tout à fait précis, on passe ici une clef entre crochets pour récupérer la valeur qui lui est associée.

Ajout et modification de valeurs dans un dictionnaire

Les dictionnaires sont altérables, ce qui signifie qu'on va pouvoir modifier ou ajouter des valeurs dans un dictionnaire après sa création.

Pour cela, on va tout simplement utiliser le nom de notre dictionnaire suivi d'un couple de crochets [] en passant une clef entre ces crochets puis affecter une valeur à cette clef. Dans le cas où la clef existe déjà, l'ensemble clef : valeur n'est pas ajouté mais la valeur originellement liée à la clef est plutôt remplacée par la nouvelle valeur.

Illustrons immédiatement cela avec quelques exemples de manipulation des données de dictionnaires Python :

```
>>> d = {"nom": "Pierre", "age": 29, "sport" : ["Trail", "Triathlon"]}
>>> d["nom"]
'Pierre'
>>> d["age"]
29
>>> d["sport"]
['Trail', 'Triathlon']
>>>
>>> d["mail"] = "pierre.giraud@edhec.com"
>>> d["age"] = 30
>>> d
{'nom': 'Pierre', 'age': 30, 'sport': ['Trail', 'Triathlon'], 'mail': 'pierre.giraud@edhec.com'}
```

Pour supprimer une paire clef : valeur d'un dictionnaire, nous allons utiliser l'instruction `del` (abréviation de “delete” = supprimer en anglais) suivi du nom du dictionnaire avec la clef de l'élément à supprimer entre crochets comme ceci :

```
>>> del d["mail"]
>>> d
{'nom': 'Pierre', 'age': 30, 'sport': ['Trail', 'Triathlon']}
```