

# Un nouveau type de variable en Python : les dictionnaires.

## 1. Définir un dictionnaire, ajouter, supprimer et modifier des valeurs.

Un dictionnaire en Python permet d'associer à certaines valeurs appelées clés (key) d'autres valeurs (value).

Les clés sont définies de manière unique.

On considère par exemple une épreuve sportive où chaque candidat se voit attribuer un numéro de dossard.

On peut alors considérer un dictionnaire qui va attribuer à chaque numéro de dossard ( key ) le nom du candidat ( value ).

### a. Définir un dictionnaire

```
>>>dossards=dict() # on déclare un objet dossards de type dictionnaire
>>># autre méthode de déclaration
>>>dossards={}
>>>type(dossards)
<class 'dict'>
```

### b. Affecter des valeurs à des clés

```
>>>dossards[1]="M Martin" # on affecte la valeur "M Martin" à la clé 1
>>>print(dossards)
{1: 'M Martin'}
>>>dossards[2]="M Paul" # on affecte la valeur "M Paul" à la clé 2
>>>print(dossards)
{1: 'M Martin', 2: 'M Paul'}
>>>dossards[2]="M Benoit" # on modifie la valeur affectée à la clé 2
>>>print(dossards)
{1: 'M Martin', 2: 'M Benoit'}
```

On peut modifier la valeur attribuée à une clé. Contrairement aux tuples, les dictionnaires sont donc mutables.

On peut également définir le même dictionnaire directement :

```
>>>dossards={1:"M Martin",2:"M benoit"}
>>>print(dossards)
{1: 'M Martin', 2: 'M benoit'}
```

### c. Récupérer les valeurs associées à des clés.

```
>>>dossards={1:"M Martin",2:"M benoit"}
>>> dossards.get(1) # méthode pour récupérer la valeur associée à la clé
1
'M Martin'
>>> dossards.get(0) # ne renvoie rien si la clé n'est pas dans le
dictionnaire
# on peut utiliser la méthode get pour renvoyer une valeur par défaut si
la clé
# n'est pas présente dans le dictionnaire
>>> dossards.get(0,"clé non présente dans ce dictionnaire")
'clé non présente dans ce dictionnaire'
```

### d. Ajouter un élément à un dictionnaire.

On peut ajouter un élément en précisant sa clé et la valeur associée :

```
>>>print(dossards)
{1: 'M Martin', 2: 'M benoit'}
>>> dossards[3]="M Albert"
>>> print(dossards)
{1: 'M Martin', 2: 'M benoit', 3: 'M Albert'}
```

## e. Supprimer un élément d'un dictionnaire.

On peut supprimer une clé et sa valeur :

```
>>> print(dossards)
{1: 'M Martin', 2: 'M benoit', 3: 'M Albert'}
>>> del dossards[2]
>>> print(dossards)
{1: 'M Martin', 3: 'M Albert'}
```

On peut supprimer aussi une clé et sa valeur avec la méthode pop():

```
>>> dossards.pop(1)
'M Martin'
>>> print(dossards)
{3: 'M Albert'}
>>>
```

La différence avec del et que pop() renvoie la valeur attachée à la clé que l'on supprime.

## 2. Parcourir un dictionnaire

### a) Parcourir les clés d'un dictionnaire.

```
dossards={
    1:"M Martin",
    2:"M Benoit",
    3:"M Durand",
    4:"Mme Tuple",
    5:"Mme Develle"
}

def parcour_par_cle(dictionnaire):
    ...
    parcourt le dictionnaire par clé
```

```
'''
for cle in dictionnaire:
    print(cle)
```

On a alors :

```
>>> parcours_par_cle(dossards)
1
2
3
4
5
```

On peut afficher les valeurs associées aux clés

```
def parcours_par_cle(dictionnaire):
    '''
    parcourt le dictionnaire par clés
    '''
    for cle in dictionnaire:
        print(cle,dictionnaire[cle])
```

```
>>>parcours_par_cle(dossards)
1 M Martin
2 M Benoit
3 M Durand
4 Mme Tuple
5 Mme Develle
```

## Méthodes propres aux dictionnaires:

```
>>>dossards.keys() # retourne les clés sous la forme d'une liste
dict_keys([1, 2, 3, 4, 5])
>>> 5 in dossards.keys() # vérifie si une valeur est une clé
True
>>> 8 in dossards.keys()
False

>>> len(dossards) # renvoie la longueur d'un dictionnaire
5
```

On peut aussi parcourir les clés du dictionnaire en utilisant la méthode keys() :

```
def parcours_par_cle(dictionnaire):
    """
    parcourt le dictionnaire par clés
    """
    for cle in dictionnaire.keys():
        print(cle,dictionnaire[cle])
```

## b) Parcourir les valeurs d'un dictionnaire.

Méthode propre aux dictionnaires :

```
>>>dossards.values() # retourne les valeurs sous la forme d'une liste
dict_values(['M Martin', 'M Benoit', 'M Durand', 'Mme Tuple', 'Mme
Develle'])
>>> 'M Martin' in dossards.values() # vérifie si 'M Martin' est une
valeur du dictionnaire
True
>>> "M Trucmuche" in dossards.values()
False
```

```
def parcours_par_valeurs(dictionnaire):
    """
    parcourt le dictionnaire par valeurs
    """
    for val in dictionnaire.values():
        print(val)
```

```
>>> parcours_par_valeurs(dossards)
M Martin
M Benoit
M Durand
Mme Tuple
Mme Develle
```

## c) Parcours d'un dictionnaire par clé et valeurs

On peut utiliser la méthode items()

```
>>>dossards.items()
dict_items([(1, 'M Martin'), (2, 'M Benoit'), (3, 'M Durand'), (4, 'Mme
Tuple'), (5, 'Mme Develle')])
```

on peut alors utiliser la fonction suivante :

```
def parcours_par_cles_et_valeurs(dictionnaire):
    """
    parcourt le dictionnaire par clés et valeurs
    """
    for clé,valeurs in dictionnaire.items():
        print(clé,valeurs)
```

On a alors :

```
>>>parcours_par_cles_et_valeurs(dossards)
1 M Martin
2 M Benoit
3 M Durand
4 Mme Tuple
5 Mme Develle
```

### 3. Exercices

#### Exercice 1

On considère le dictionnaire suivant :

```
dossards={
    1:"M Martin",
    2:"M Benoit",
    3:"M Durand",
    4:"Mme Tuple",
    5:"Mme Develle"
}
```

En utilisant la console et les méthodes associées aux dictionnaires :

a) Ajouter le dossard 6 au nom de “Mme Martin”,

```
>>>
```

b) Ajouter "M Albert" avec le dossard 9,

c) supprimer le dossard 3,

d) Modifier la valeur de la clé 2 en "Mme Benoit"

## Exercice 2

On considère les 2 dictionnaires suivants donnant pour chaque numéro de dossard respectivement le nom du candidat et le temps mis lors de la course.

```
dossards={
    1:"M Martin",
    2:"M Benoit",
    3:"M Durand",
    4:"Mme Tuple",
    5:"Mme Develle"
}
temps={
    1: 45,
    2: 72,
    3: 64,
    4: 42,
    5: 66
}
>>> print(temps)
{1: 45, 2: 72, 3: 64, 4: 42, 5: 66}
>>> print(dossards)
{1: 'M Martin', 2: 'M Benoit', 3: 'M Durand', 4: 'Mme Tuple', 5: 'Mme
Develle'}
```

Créer une fonction qui permet, à partir des dictionnaires `dossards` et `temps`, d'élaborer un nouveau dictionnaire dont les clés seront les noms des candidats et les valeurs le temps des candidats.

```
def resultats_candidats(dict1,dict2):  
    '''  
    dict1 et dict2 sont 2 dictionnaires de même longueur et avec les  
    mêmes clés  
    renvoie un dictionnaire ayant pour clés les valeurs de dict1 et pour  
    valeurs le temps  
    des candidats  
    dict1 : dict  
    dict2 : dict  
    return : dict  
    >>>resultats=resultats_candidats(dossards,temps)  
    >>>resultats["Mme Tuple"]  
    42  
    '''
```