

Dictionnaires en Python

Nicolas Markey

PCSI3 - Lycée Chateaubriand, Rennes

Les dictionnaires

Liste vs. dictionnaire

- **liste** : tableau ordonné, accès par un indice entre 0 et $n - 1$

```
>>> friends = ['Marc', 'Sophie', 'Peter']
```

```
>>> friends.pop(0)
```

```
'Marc'
```

```
>>> friends[0]
```

```
'Sophie'
```

- **dictionnaire** : tableau associatif, non-ordonné ; accès par une clé (immutable)

```
>>> phone = {'Marc': '0634050143', 'Sophie': '+33623860192',  
            'Peter': '+443535692101'}
```

```
>>> phone['Marc']
```

```
'0634050143'
```

Les dictionnaires

Liste vs. dictionnaire

- **liste** : tableau ordonné, accès par un indice entre 0 et $n - 1$

```
>>> friends = ['Marc', 'Sophie', 'Peter']
```

```
>>> friends.pop(0)
```

```
'Marc'
```

```
>>> friends[0]
```

```
'Sophie'
```

- **dictionnaire** : tableau associatif, non-ordonné ; accès par une clé (immutable)

```
>>> phone = {'Marc': '0634050143', 'Sophie': '+33623860192',  
            'Peter': '+443535692101'}
```

```
>>> phone['Marc']
```

```
'0634050143'
```

Utilisation d'un dictionnaire

Principales commandes pour manipuler des dictionnaires

- **création, ajout d'éléments :**

```
>>> marks={'Jean': [12,16,13], 'Marco': [12,9,13]}
```

```
>>> marks['Raissa']=[19,16,14]
```

```
>>> marks
```

```
{'Jean': [12,16,13], 'Marco': [12,9,13], 'Raissa': [19,16,14]}
```

```
>>> repertoire=[('Marc', '0634050143'), ('Sophie', '+33623860192')]
```

```
>>> phone=dict(repertoire)
```

```
>>> phone
```

```
{'Marc': '0634050143', 'Sophie': '+33623860192'}
```

- **suppression d'éléments**

```
>>> marks.pop('Marco')
```

```
[12,9,13]
```

```
>>> marks
```

```
{'Jean': [12,16,13], 'Raissa': [19,16,14]}
```

Utilisation d'un dictionnaire

Principales commandes pour manipuler des dictionnaires

- **création, ajout d'éléments :**

```
>>> marks={'Jean': [12,16,13], 'Marco': [12,9,13]}
```

```
>>> marks['Raissa']=[19,16,14]
```

```
>>> marks
```

```
{'Jean': [12,16,13], 'Marco': [12,9,13], 'Raissa': [19,16,14]}
```

```
>>> repertoire=[('Marc', '0634050143'), ('Sophie', '+33623860192')]
```

```
>>> phone=dict(repertoire)
```

```
>>> phone
```

```
{'Marc': '0634050143', 'Sophie': '+33623860192'}
```

- **suppression d'éléments**

```
>>> marks.pop('Marco')
```

```
[12,9,13]
```

```
>>> marks
```

```
{'Jean': [12,16,13], 'Raissa': [19,16,14]}
```

Utilisation d'un dictionnaire

Principales commandes pour manipuler des dictionnaires

- **création, ajout d'éléments :**

```
>>> marks={'Jean': [12,16,13], 'Marco': [12,9,13]}
```

```
>>> marks['Raissa']=[19,16,14]
```

```
>>> marks
```

```
{'Jean': [12,16,13], 'Marco': [12,9,13], 'Raissa': [19,16,14]}
```

```
>>> repertoire=[('Marc', '0634050143'), ('Sophie', '+33623860192')]
```

```
>>> phone=dict(repertoire)
```

```
>>> phone
```

```
{'Marc': '0634050143', 'Sophie': '+33623860192'}
```

- **suppression d'éléments**

```
>>> marks.pop('Marco')
```

```
[12,9,13]
```

```
>>> marks
```

```
{'Jean': [12,16,13], 'Raissa': [19,16,14]}
```

Utilisation d'un dictionnaire

Principales commandes pour manipuler des dictionnaires

- **accès aux clés et aux valeurs :**

```
>>> keyvals=marks.items()
>>> keyvals
dict_items([('Jean', [12,16,13]), ('Raissa', [19,16,14])])
>>> marks['Ben']=[12,5]
>>> keyvals
dict_items([('Jean', [12,16,13]), ('Raissa', [19,16,14]), ('Ben': [12,5])])

>>> for key,val in keyvals:
    print('La meilleure note de',key,'est',max(val))
La meilleure note de Jean est 16
La meilleure note de Raissa est 19
La meilleure note de Ben est 12

>>> values=marks.values()
>>> keys=marks.keys()
```

Utilisation d'un dictionnaire

Principales commandes pour manipuler des dictionnaires

- **accès aux clés et aux valeurs :**

```
>>> keyvals=marks.items()
>>> keyvals
dict_items([('Jean', [12,16,13]), ('Raissa', [19,16,14])])
>>> marks['Ben']=[12,5]
>>> keyvals
dict_items([('Jean', [12,16,13]), ('Raissa', [19,16,14]), ('Ben': [12,5])])

>>> for key,val in keyvals:
    print('La meilleure note de',key', 'est',max(val))
La meilleure note de Jean est 16
La meilleure note de Raissa est 19
La meilleure note de Ben est 12

>>> values=marks.values()
>>> keys=marks.keys()
```

Utilisation d'un dictionnaire

Principales commandes pour manipuler des dictionnaires

- **accès aux clés et aux valeurs :**

```
>>> keyvals=marks.items()
>>> keyvals
dict_items([('Jean', [12,16,13]), ('Raissa', [19,16,14])])
>>> marks['Ben']=[12,5]
>>> keyvals
dict_items([('Jean', [12,16,13]), ('Raissa', [19,16,14]), ('Ben': [12,5])])

>>> for key,val in keyvals:
    print('La meilleure note de',key', 'est',max(val))
La meilleure note de Jean est 16
La meilleure note de Raissa est 19
La meilleure note de Ben est 12

>>> values=marks.values()
>>> keys=marks.keys()
```

Utilisation d'un dictionnaire

Principales commandes pour manipuler des dictionnaires

- **copie :**

```
>>> notes=marks                ## pointent vers le même objet
>>> notes['Jean'][2]=12
>>> marks
{'Jean': [12,17,12], 'Marco': [12,9,13], 'Raissa': [19,16,14]}

>>> from copy import deepcopy
>>> notes=deepcopy(marks)      ## copie profonde : 2 dicts distincts
>>> notes['Jean'][2]=13
>>> marks
{'Jean': [12,17,12], 'Marco': [12,9,13], 'Raissa': [19,16,14]}
>>> notes
{'Jean': [12,17,13], 'Marco': [12,9,13], 'Raissa': [19,16,14]}
```

Utilisation d'un dictionnaire

Principales commandes pour manipuler des dictionnaires

- **copie :**

```
>>> notes=marks                ## pointent vers le même objet
>>> notes['Jean'][2]=12
>>> marks
{'Jean': [12,17,12], 'Marco': [12,9,13], 'Raissa': [19,16,14]}

>>> from copy import deepcopy
>>> notes=deepcopy(marks)      ## copie profonde : 2 dicts distincts
>>> notes['Jean'][2]=13
>>> marks
{'Jean': [12,17,12], 'Marco': [12,9,13], 'Raissa': [19,16,14]}
>>> notes
{'Jean': [12,17,13], 'Marco': [12,9,13], 'Raissa': [19,16,14]}
```

Utilisation d'un dictionnaire

Principales commandes pour manipuler des dictionnaires

- **test d'égalité, appartenance, taille :**

```
>>> marks=dict()
```

```
>>> marks=={}
```

```
True
```

```
>>> marks['Raissa']=[19,16,14]
```

```
>>> marks=={}
```

```
False
```

```
>>> 'Raissa' in marks
```

```
True
```

```
>>> len(marks)
```

```
1
```

Utilisation d'un dictionnaire

Principales commandes pour manipuler des dictionnaires

- **test d'égalité, appartenance, taille :**

```
>>> marks=dict()
```

```
>>> marks=={}
```

```
True
```

```
>>> marks['Raissa']=[19,16,14]
```

```
>>> marks=={}
```

```
False
```

```
>>> 'Raissa' in marks
```

```
True
```

```
>>> len(marks)
```

```
1
```

Utilisation d'un dictionnaire

Principales commandes pour manipuler des dictionnaires

- **test d'égalité, appartenance, taille :**

```
>>> marks=dict()
```

```
>>> marks=={}
```

```
True
```

```
>>> marks['Raissa']=[19,16,14]
```

```
>>> marks=={}
```

```
False
```

```
>>> 'Raissa' in marks
```

```
True
```

```
>>> len(marks)
```

```
1
```

Effacité des dictionnaires par rapport aux listes

Temps de calcul d'un test d'appartenance

```
>>> from time import time
>>> liste = [i for i in range(1_000_000)]
>>> dico = {i:i for i in range(1_000_000)}
>>> def time_in_liste():
    start=time()
    'a' in liste
    return time()-start
>>> def time_in_dico():
    start=time()
    'a' in dico
    return time()-start

>>> time_in_liste()
0.019706010818481445
>>> time_in_dico()
4.0531158447265625e-06
```

Utilisation d'un dictionnaire

Exercice

Écrire une fonction qui prend en entrée une liste itérable d'éléments immutables (par exemple, une chaîne de caractères) et retourne un dictionnaire contenant le nombre d'occurrences de chacun de ces éléments (par exemple, le nombre d'occurrences de chaque caractère)

```
>>> def occurrences(liste):
```

```
>>> occurrences("abracadabra")  
{ 'a': 5, 'b': 2, 'r': 2, 'c': 1, 'd': 1 }
```

Utilisation d'un dictionnaire

Exercice

Écrire une fonction qui prend en entrée une liste itérable d'éléments immutables (par exemple, une chaîne de caractères) et retourne un dictionnaire contenant le nombre d'occurrences de chacun de ces éléments (par exemple, le nombre d'occurrences de chaque caractère)

```
>>> def occurrences(liste):
    dico={}
    for elt in liste:
        if elt in dico:
            dico[elt]=dico[elt]+1
        else:
            dico[elt]=1
    return dico
>>> occurrences("abracadabra")
{'a': 5, 'b': 2, 'r': 2, 'c': 1, 'd': 1}
```