

INFO-H-100 - Informatique

Séance d'exercices 13
Introduction à Python
Dictionnaires et fichiers

Université Libre de Bruxelles
Faculté des Sciences Appliquées

2011-2012

Dictionnaire

Un **dictionnaire** est un un type de données **mutable**.

A la différence des listes et des strings qui sont ordonnées et indexées par intervalle numérique, les dictionnaires sont **non ordonnés et indexés par des clés**.

Un dictionnaire est un **ensemble non ordonné de couples clé-valeur**.

```
>>> tel = {}
>>> tel["Boris"] = 3766
>>> tel["Thierry"] = 5603
>>> tel["Alain"] = 1234
>>> print tel
{'Boris': 3766, 'Alain': 1234, 'Thierry': 5603}
```

Les clés doivent être **immuables (nombres, chaînes, tuples)**. Les valeurs peuvent être de n'importe quel type.

Opérations sur les dictionnaires

```
>>> tel.items()                                #liste des elements (paires)
[('Boris', 3766), ('Alain', 1234), ('Thierry', 5603)]
>>> tel.keys()                                 #liste des cle
['Boris', 'Alain', 'Thierry']
>>> tel.values()                               #liste des valeurs
[3766, 1234, 5603]
>>> for cle in tel:                            #iteration
...     print cle + ": " + str(tel[cle])
...
Boris: 3766
Alain: 1234
Thierry: 5603
```


Exemple d'utilisation

```
def occurrences(mot):  
    d = {}  
    for c in mot:  
        d[c] = d.get(c, 0) + 1  
    return d  
  
print occurrences("banane")
```

Affiche

```
{'a': 2, 'b': 1, 'e': 1, 'n': 2}
```

Questions :

- Que faut-il changer au niveau de la fonction pour qu'elle s'applique à n'importe quelle séquence ?
- Cette fonction s'applique-t-elle à une liste de listes ?

Lecture de fichiers

Soit le fichier `info.txt` situé dans le répertoire courant et contenant les lignes suivantes :

```
| Anh Vu  
| Gary
```

```
| >>> f = open("info.txt")  
| >>> f.read()  
| 'Anh Vu\nGary'
```

```
| >>> f = open("info.txt")  
| >>> f.readline()  
| 'Anh Vu\n'  
| >>> f.readline()  
| 'Gary'  
| >>> f.readline()  
| ''
```

```
| >>> f = open("info.txt")  
| >>> f.readlines()  
| ['Anh Vu\n', 'Gary']
```

```
| >>> f = open("info.txt")  
| >>> for l in f.readlines():  
| ...     l  
| ...  
| 'Anh Vu\n'  
| 'Gary'
```

Remarquez la présence du séparateur de fin de ligne dans certains résultats. `strip()` permet de s'en débarrasser.

```
| >>> f = open("info.txt")  
| >>> f.readline().strip()  
| 'Anh Vu'
```

Écriture de fichiers

Mode append (ajout en fin) :

```
>>> f = open("info.txt", "a")
>>> f.write("\nBoris")
>>> f.close() #finalise l'écriture
```

Contenu du fichier :

```
| Anh Vu
| Gary
| Boris
```

Mode write (remplace l'existant) :

```
>>> f = open("info.txt", "w")
>>> f.write("Alain")
>>> f.close()
```

Contenu du fichier :

```
| Alain
```

Quelques outils utiles sur les *strings*

```
>>> s = " \n Foo\nBar spam\n"
>>> s
' \n Foo\nBar spam\n'
>>> s.upper()           # majuscules
' \n FOO\nBAR SPAM\n'
>>> s.lower()          # minuscules
' \n foo\nbar spam\n'
>>> s.strip()          # nettoyage de debut et fin
'Foo\nBar spam'
>>> s.replace("\n", "-") # remplacement
' - Foo-Bar spam-'
>>> s.replace("\n", "") # suppression
'  FooBar spam'
>>> s.split()           # decoupe aux "blancs" (whitespaces)
['Foo', 'Bar', 'spam']
>>> s.split("\n")
[' ', ' Foo', 'Bar spam', '']
```

Attention, les *strings* sont **immuables**. Ces méthodes renvoient des copies.

Vous trouverez d'autres outils utiles dans la documentation.

Exercices