

INFO-H-100 - Informatique

Séance d'exercices 10
Introduction à Python
Dictionnaires et fichiers

Université libre de Bruxelles
École polytechnique de Bruxelles

2013-2014

Dictionnaire

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

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

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

```
>>> phone = {}  
>>> phone["Gary"] = 3766  
>>> phone["Thierry"] = 5603  
>>> phone["Alain"] = 1234  
>>> print(phone)  
{'Gary': 3766, 'Alain': 1234, 'Thierry': 5603}
```

Les clefs doivent être immutables (nombres, chaînes, tuples).

Les valeurs peuvent être de n'importe quel type.

Opérations sur les dictionnaires

```
>>> phone.items()                                # Liste des éléments (paires)
dict_items([('Gary', 3766), ('Alain', 1234), ('Thierry', 5603)])
>>> phone.keys()                                # Itérateur contenant les clefs
dict_keys(['Alain', 'Thierry', 'Gary'])
>>> list(phone.keys())                          # Liste des clefs
['Alain', 'Thierry', 'Gary']
>>> phone.values()                              # Itérateur des valeurs
dict_values([1234, 5603, 3766])
>>> list(phone.values())                        # Liste des valeurs
[1234, 5603, 3766]
>>> for key in phone:                          # Itération
...     print(key + ": " + str(phone[key]))
...
Alain: 1234
Thierry: 5603
Gary: 3766
```

Opérations sur les dictionnaires (suite)

```
>>> "Gary" in phone           # Test d'existence de clef
True
>>> del phone["Thierry"]      # Suppression
>>> print(phone)
{'Alain': 1234, 'Gary': 3766}
>>> print(phone["Thierry"])   # Clef inexistante -> Erreur

...KeyError: 'Thierry'
>>> print(phone.get("Thierry")) # Clef inexistante -> None
None
>>> print(phone.get("Thierry", 1307)) # Clef inexistante -> ...
1307                                # Valeur choisie
                                    # (renseignements)
```

Exemple d'utilisation

```
def occurrences(word):  
    dic = {}  
    for char in word:  
        dic[char] = dic.get(char, 0) + 1  
    return dic  
  
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.readlines()  
['Anh Vu\n', 'Gary']
```

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

```
>>> f = open("info.txt")  
>>> for line in f.readlines():  
...     print(line)  
...  
'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("\nGary")  
>>> f.close() # Finalise l'écriture
```

Contenu du fichier :

```
Anh Vu  
Gary  
Gary
```

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 début et fin
'Foo\nBar spam'
>>> s.replace("\n", "-") # Remplacement
' - Foo-Bar spam-'
>>> s.replace("\n", "") # Suppression
' FooBar spam'
>>> s.split() # Découpe 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