

# Dictionnaires en Python

**Zhentao Li**

École Normale Supérieure

5 mars 2014

# Dictionnaires

Les listes nous permettent d'accéder aux éléments de la liste par indices.

```
>>> animaux = ['girafe', 'hippopotame', 'singé', 'dahu']  
>>> animaux[2]  
'singé'
```

# Dictionnaires

Les listes nous permettent d'accéder aux éléments de la liste par indices.

```
>>> animaux = ['girafe', 'hippopotame', 'singe', 'dahu']  
>>> animaux[2]  
'singe'
```

Mais parfois, on aimerait que les indices soient autre chose que des numéros. Des chaînes de caractères, par exemple.

```
>>> nombre["giraffe"]  
3  
>>> nombre["singe"]  
5  
>>> emplacement["giraffe"]  
"Cage nord-ouest"
```

# Dictionnaires

Les listes nous permettent d'accéder aux éléments de la liste par indices.

```
>>> animaux = ['girafe', 'hippopotame', 'singe', 'dahu']
>>> animaux[2]
'singe'
```

Mais parfois, on aimerait que les indices soient autre chose que des numéros. Des chaînes de caractères, par exemple.

```
>>> nombre["giraffe"]
3
>>> nombre["singe"]
5
>>> emplacement["giraffe"]
"Cage nord-ouest"
```

Une telle representation de l'information est possible avec le dictionnaire.

```
>>> nombre = {"giraffe":3, "singe":5}
>>> emplacement = {"giraffe":"Cage nord-ouest"}
```

# Dictionnaire

- La **liste** associe à chaque **indice** une valeur.  
Le **dictionnaire** associe à chaque **clé** une valeur.
- Une clé de dictionnaire peut être un **chiffre**, une **chaîne de caractères**, une **paire**, un ***n*-tuple**, etc. Mais pas une **liste** ou un autre **dictionnaire**.
- Un dictionnaire est créé en utilisant les accolades `{}` avec la syntaxe

```
{cle1: valeur1, cle2: valeur2, ...}
```

# Dictionnaire

- La **liste** associe à chaque **indice** une valeur.  
Le **dictionnaire** associe à chaque **clé** une valeur.
- Une clé de dictionnaire peut être un **chiffre**, une **chaîne de caractères**, une **paire**, un ***n*-tuple**, etc. Mais pas une **liste** ou un autre **dictionnaire**.
- Un dictionnaire est créé en utilisant les accolades `{}` avec la syntaxe

```
{cle1: valeur1, cle2: valeur2, ...}
```

## Remarque

Les dictionnaires sont une structure de donnée de base en Python mais est inhabituelle pour d'autre langages de programmation.

De plus il est possible de simuler un dictionnaire avec des listes (c'est-à-dire écrire un ensemble de fonctions qui remplissent les mêmes fonctions que celle d'un dictionnaire).

# Opération sur les dictionnaires

- Ajout ou mise à jour de valeur.

```
>>> nombre['dahu'] = 1
>>> nombre["singe"] = 6
>>> nombre
{'giraffe': 3, 'dahu': 1, 'singe': 6}
```

- Déterminer si une clé existe dans un dictionnaire.

```
>>> 'singe' in nombre
True
>>> 'ornithorynque' in nombre
False
```

Obtenir les clés, valeurs ou paires clé valeur comme liste.

```
>>> nombre.keys()
['giraffe', 'dahu', 'singe']
>>> nombre.values()
[3, 1, 6]
>>> nombre.items()
[('giraffe', 3), ('dahu', 1), ('singe', 6)]
```

# Opération sur les dictionnaires

- Conversion de liste de paires en dictionnaire.

```
>>> liste_de_paires = nombre.items()
>>> dict(liste_de_paires)
{'giraffe': 3, 'dahu': 1, 'singe': 6}
```

- Itération sur les clés

```
>>> for cle in nombre:
...     print cle
...
giraffe
dahu
singe
```

- Enlever une paire clé-valeur.

```
>>> del nombre['giraffe']
>>> nombre
{'dahu': 1, 'singe': 6}
```