

# Python Description

December 8, 2016

## 1 Avantages / Inconvénients Python

### 1.1 Avantages

- Pas d'inscruction de fin de commande
- Meilleure lisibilité du code grâce à l'indentation du obligatoire des boucles
- Beaucoup de librairies disponibles (à vérifier et approfondir)
- Pas forcément besoin d'un IDE

### 1.2 Désavantages

- Erreurs liées à l'indentation de fin de boucle
- Flexibilités au niveau des classes (créer un attribut à la volée ...)

### 1.3 Sources

<http://sametmax.com/10-raisons-pour-lesquelles-je-suis-toujours-marie-a-python/>

## 2 Cas d'utiliation pour les 4 structures de données intégrées

### 2.1 List

Les listes sont modifiables. Elles doivent donc être utilisées uniquement lorsque l'on sait que cette liste est suceptible d'évoluer.

```
In [18]: cats = ['Tom', 'Snappy', 'Kitty', 'Jessie', 'Chester']  
         cats.append('Minou')
```

```
         print(cats)
```

```
['Tom', 'Snappy', 'Kitty', 'Jessie', 'Chester', 'Minou']
```

```
In [19]: del cats[0]
```

```
         print(cats)
```

```
['Snappy', 'Kitty', 'Jessie', 'Chester', 'Minou']
```

## 2.2 Tuple

Les tuples sont comme des listes mais non modifiables. Elles ont donc l'avantage d'être plus rapide pour les manipulations (extractions etc ...) que les listes.

```
In [20]: months = ('January', 'February', 'March', 'April', 'May', 'June', \
                  'July', 'August', 'September', 'October', 'November', 'December')
```

## 2.3 Dictionnaires

Les dictionnaires fonctionnent sur un système de clé-valeurs. On a donc un index.

```
In [21]: ages = {}

        #Ajout de données
        ages['Sue'] = 23
        ages['Peter'] = 19
        ages['Andrew'] = 78
        ages['Karren'] = 45

        print(ages)

{'Andrew': 78, 'Karren': 45, 'Sue': 23, 'Peter': 19}
```

Il est très facile de retrouver une valeur associée à la clé en connaissant cette clé.

```
In [22]: if 'Sue' in ages:
        print ("Sue est dans le dictionnaire. Son age : ", ages['Sue'])

        else:
        print ("Sue n'est pas dans le dictionnaire")

Sue est dans le dictionnaire. Son age : 23
```

## 2.4 Set

Un set est une collection d'éléments uniques non ordonnée. On va pouvoir l'utiliser pour supprimer des doublons et pour des opérations mathématiques telles que union, intersection ... plus rapidement qu'avec une liste.

```
In [27]: basket = {'apple', 'orange', 'apple', 'pear', 'orange', 'banana'}
        print(basket)                                # duplicatas enlevés
        print('orange' in basket)                    # test de présence rapide
        print('crabgrass' in basket)

{'banana', 'apple', 'orange', 'pear'}
True
False
```

### **3 Conclusion**

La liste est un objet peut être trop général. Les trois autres éléments sont des spécialisations de l'objet liste. Il convient donc d'utiliser le bon objet en fonction de la situation.

### **4 Sources**

<http://sthurlow.com/python/lesson06/> <https://docs.python.org/2/library/sets.html>