

Exercice2

March 16, 2018

1 Point fort et point faible des 4 structures intégrés

Python est un langage simple à assimiler avec la syntaxe, ce qui le rend facile à apprendre à l'inverse de certaines syntaxes dans d'autres langages. Ajoutons à cela que le langage est très tolérant puisqu'il nous considère et interprète des erreurs que d'autres langages ne considéreraient pas. Cet avantage facilite amplement l'apprentissage de la programmation, les syntaxes viennent instinctivement. Le souci du détail étant volontairement altéré, développer avec un IDE est d'une aisance rarement égalée, de plus le terminal retourne les erreurs avec des données précises.

source: <https://www.programmez.com/tutoriels/lalternative-en-python>

2 structure list

avantage : peuvent être mutable (la table, le nb d'élément et les valeurs sont changeable), ordonné, itérable et indexable (possibilité de faire `liste[0] = valeur`), et on peut mettre n'importe quoi dedans en autant de fois que voulue.

Element vide : [] source: https://www.irit.fr/~Philippe.Muller/Cours/Magistere/Python_Magistere_L3_CT

```
In [7]: # listes ... on peut mélanger les types, mais cela n'a pas trop de sens
        l=[1,2,3,6,"hello"]
        print (l[4])
```

hello

3 structure tuple

desavantage : ne peut pas être changé une fois créé (immutable) avantage : on peut mélanger les types

Element vide : ()

sources: https://www.irit.fr/~Philippe.Muller/Cours/Magistere/Python_Magistere_L3_CTD1.html

```
In [3]: x = (1, 'a', 2, 'b')
        type(x)
```

Out[3]: tuple

```
In [10]: x = (1,2,3)
```

```
x[0:2]
```

```
Out[10]: (1, 2)
```

4 structure dict

Similaire aux list et au tuple. Pour chaque valeur que l'on inclue, il faut aussi une clé pour sortir la valeur

Les dictionnaires sont des objets pouvant en contenir d'autres, à l'instar des listes. Cependant, au lieu d'héberger des informations dans un ordre précis, ils associent chaque objet contenu à une clé (la plupart du temps, une chaîne de caractères). Par exemple, un dictionnaire peut contenir un carnet d'adresses et on accède à chaque contact en précisant son nom.

Element vide : {}

source : <https://python.sdv.univ-paris-diderot.fr/#dictionnaires>

```
In [9]: ani1 = {}
        ani1['nom'] = 'girafe'
        ani1['taille'] = 5.0
        ani1['poids'] = 1100
        ani1
        {'nom': 'girafe', 'poids': 1100, 'taille': 5.0}
        ani1['taille']
```

```
Out[9]: 5.0
```

5 structure set

Un set est un ensemble, c'est-à-dire une collection non ordonnée d'éléments uniques, ce qui se révèle très pratique dans beaucoup d'usages courants. Ce nouveau type conteneur étant fourni en deux saveurs, l'une mutable (comme les listes), et l'autre immuable (comme les tuples).

sources: <https://www.programiz.com/python-programming/set>

```
In [11]: # set of integers
        my_set = {1, 2, 3}
        print(my_set)

        # set of mixed datatypes
        my_set = {1.0, "Hello", (1, 2, 3)}
        print(my_set)
```

```
{1, 2, 3}
```

```
{1.0, 'Hello', (1, 2, 3)}
```

```
In [14]: ##ipython nbconvert --to pdf Exercice2.ipynb pdf dans invite de commande
```

```
File "<ipython-input-14-dbf37aa76223>", line 2
ipython nbconvert to pdf execute Exercice2.ipynb
      ^
SyntaxError: invalid syntax
```