

## QUELQUES RAPPELS EN PYTHON

Le but de ce document est de (re)voir quelques éléments de base sur les fonctions et les listes.

### 1 Prise en main et première fonction

Pour créer et utiliser un programme en Python, il a deux phases :

- une phase d’écriture que l’on fait généralement dans un éditeur de code.
- une phase d’utilisation que l’on effectue dans une console. Dans la suite toutes les commandes à écrire dans la console seront précédées de `>>>`

1. Cliquer sur le lien : Exemple n°1

La page est divisée en 2, à gauche l’éditeur avec une fonction déjà écrite et à droite la console.

2. Cliquer sur le bouton **Exécuter** (en bas à gauche).

Si tout se passe bien le texte `>>> # script executed` s’affiche dans la console.

3. Taper dans la console : `>>> valeur_absolue(-5)` (ne pas réécrire les trois `>`).

4. Taper ensuite `>>> valeur_absolue(12)`

Explication sur la fonction :

```
1 def valeur_absolue(x) :
2     """calculé la valeur absolue du nombre réel x"""
3     if x >= 0 :
4         reponse = x
5     else :
6         reponse = -x
7     return reponse
```

— ligne 1 : `def` permet de définir une fonction qui s’appelle ici `valeur_absolue` et qui a pour paramètre d’entrée le nombre `x`. Notez-bien les « deux points » en fin de ligne.

— ligne 2 : phrase d’explication sur la fonction appelée DocString.

— ligne 3 à 6 : Test conditionnel et affectation d’une valeur dans la variable `reponse`.

— ligne 7 : la commande `return` permet à la fonction de renvoyer la réponse. (Nous verrons à la rentrée la différence avec la commande `print` que vous connaissez peut être.)

**Remarque :** l’indentation, c’est à dire le décalage des lignes (ou tabulation) est très important en python, il est impératif de la respecter.

À cette étape, vous pouvez faire l’exercice n° 1 du devoir à la maison.

### 2 Utilisation des éléments d’une liste en Python

#### 2.1 Quelques points les listes en Python

Une liste en Python (souvent appelée tableau dans d’autres langages informatiques) est un ensemble ordonné d’objets. Pour l’instant nous ne considérerons que des listes de nombres.

Taper les lignes suivantes dans la console de Basthon :

1. Une liste est notée entre crochets avec les éléments séparés par des virgules :

```
>>> L = [5, 17, 8, 9, 1, 25]
```

2. Le nombre d’éléments d’une liste est obtenue avec la commande `len`

```
>>> len(L)
```

3. Les éléments d’une liste sont numérotés à partir de 0 et on les obtient de la façon suivante :

```
>>> L[0]
```

```
>>> L[1]
```

```
>>> L[5]
```

4. La commande suivante provoquera un message d’erreur :

```
>>> L[6]
```

## 2.2 Somme des éléments d'une liste

1. Cliquer sur le lien suivant : Exemple n°2  
Exécuter et utiliser la fonction sur quelques listes de votre choix.

2. Voici la fonction :

```
1 def somme(L):
2     """calculé la somme des éléments d'une liste"""
3     total = 0
4     for k in range(len(L)):
5         total = total + L[k]
6     return total
```

3. Quelques explications

- A la ligne 3, on crée une variable à laquelle on affecte la valeur 0
- Dans les lignes 4 et 5, on effectue la boucle `for`, en ajoutant à la variable `total` les éléments de la liste l'un après l'autre.  
Si la liste possède  $n$  éléments, alors `len(L)` est égal à  $n$  et la commande `range` permet de faire varier  $k$  de 0 à  $n-1$ .

4. Voici le tableau des valeurs prises par les différentes variables lors de l'exécution de la commande `somme([4, 9, 2, 11])` :

	k	L[k]	total
Initialisation :			0
Boucle for, étape n° 1 :	0	4	4
Boucle for, étape n° 2 :	1	9	13
Boucle for, étape n° 3 :	2	2	15
Boucle for, étape n° 4 :	3	11	26

Valeur renvoyée par la fonction : 26