

## I Mise en place

### 1.1 Winpython

Si ce n'est déjà fait, téléchargez **Winpython**. Pour l'installation, laissez-vous guider, en notant tout de même l'endroit où vous pourrez trouver **Spyder.exe**. Si vous utilisez un Macintosh, ou un linux, il est possible d'installer Spyder tout de même. Consulter un moteur de recherche.

### 1.2 Déroulé de devoir, rendu

Vous rendrez pour ce devoir 2 documents : une copie papier et une feuille de script python. Pour **chaque** question qui demande d'écrire du code, vous décrirez par quelques phrases votre approche (algorithme). sur la copie. La procédure de rendu des feuilles de script a été expliqué en cours.

### 1.3 Tester le code produit

Il est très important de toujours tester les fonctions que l'on vient de créer. Pour cela, une fois un premier essai de code effectué, vous exécuterez votre feuille via F5, puis si votre fonction se nomme `truc` et attend une chaîne de caractère comme argument, on pourra taper

```
truc('une chaine de test')
```

et changer la chaîne de test pour essayer le plus de cas possibles (les cas "pénibles" pour votre algorithme : un seul mot, chaîne vide...)

#### 1.3.1 Docstrings

Dans le fichier que vous allez devoir compléter (*dm1.py*), les fonctions à créer ont déjà une docstring, c'est à dire une description du comportement attendu. Ces descriptions font partie intégrante de l'énoncé.

```
def ma_super_fonction(x):  
    """  
    Je suis une docstring. Il est très malpoli de m'effacer  
    """  
    # ici on complète la fonction.
```

Il s'agit de la partie entre les triples guillemets. Vous complétez les fonctions en **dessous** de la docstring.

## II En route !

Le but de ce devoir est d'analyser une version simplifiée de la page wikipedia portant sur python.

### Exercice 1

Tester la commande

```
s.split()
```

dans la console, où `s` est une chaîne de caractère. Il faudra faire plusieurs tests pour comprendre l'effet de cette commande.

Compléter ensuite la fonction `nombre_de_mots`

### Exercice 2

Vous disposez d'un fichier `wiki-py.txt` qui contient notre page à analyser. Pour lire son contenu on utilise

```
f = open(nom_fichier)  
L = f.readlines()  
f.close()
```

où `nom_fichier` est une chaîne contenant le nom du fichier à ouvrir. Dans la variable `L` on a maintenant la liste de toutes les lignes du fichier.

Compléter la fonction `compte_mots`. On la testera via

```
compte_mots('wiki-py.txt')
```

qui ne fonctionnera que si vous avez exécuté la feuille de script via F5 (ou la flèche verte). De plus, le fichier `wiki-py.txt` doit être dans le même dossier que votre feuille de script `dm1.py`.

### Exercice 3

Trouver le nombre de fois où le mot "python" apparaît dans le fichier. Vous indiquerez sur votre copie la méthode ainsi que le résultat. Aucun code n'est attendu dans le fichier `dm1.py`

### Exercice 4

Compléter la fonction `longueur_moyenne`.

### Exercice 5 (Bonus)

Compléter la fonction `frequences`