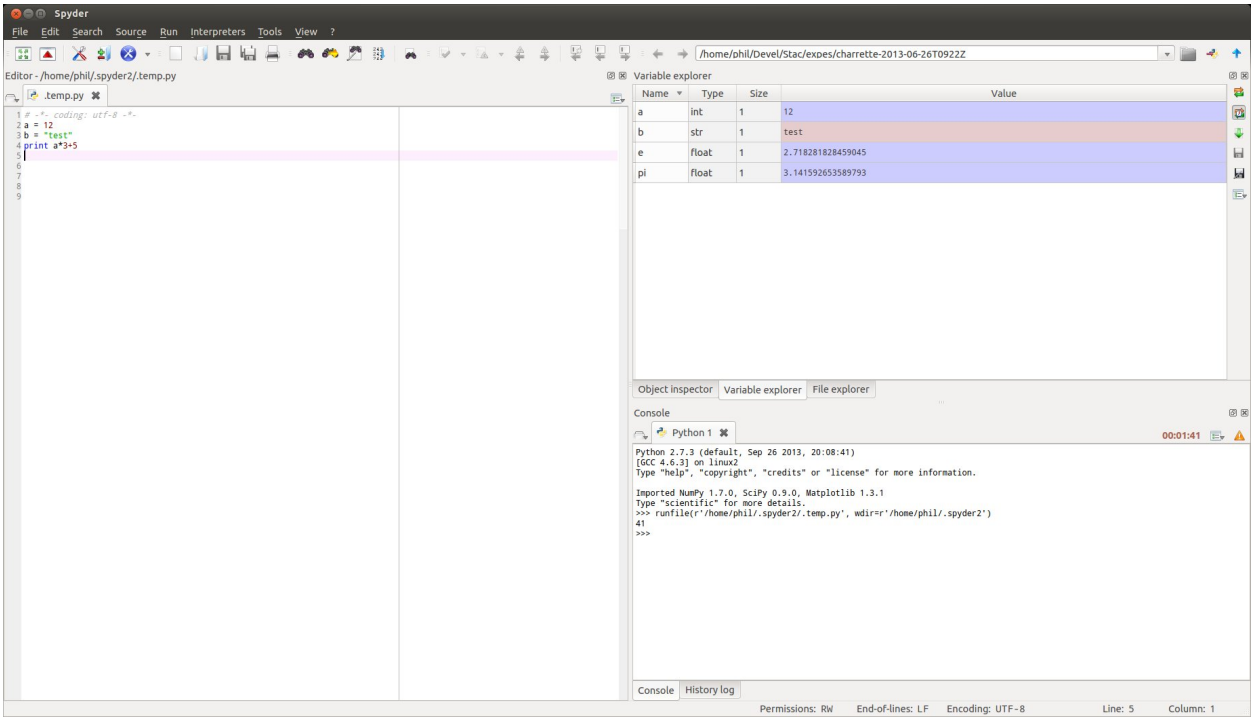
TP_Python_SRI_1

September 7, 2021

Programmation en Python --- TP1



Environnement de travail

Vous êtes libres d'utiliser l'éditeur de votre choix pour écrire vos programmes, et vous pouvez lancer votre programme avec l'interpréteur python à partir du terminal, par exemple~:

```
python -i monprog.py argument1 argument2 ...
```

Si votre programme attend des arguments (cf exemple ci-dessous), et avec l'option -i pour pouvoir rester dans l'interpréteur et faire différents tests.

Mais il vaut mieux utiliser l'environnement de développement installé sur votre machine: spyder. Cela vous fournit éditeur/environnement d'éxécution/débuggueur/inspection du programme et aide à la demande.

Vous avez l'éditeur à gauche, l'interpréteur en bas à droite, et différentes aides en haut à droite.

Vérifier dans le menu Affichage/Volets que vous avez bien activé l'inspecteur d'objets et l'inspecteur de variables.

Pour récupérer des arguments d'un programme lancé depuis le terminal, vous pouvez utiliser le module sys

sys.argv est alors une liste contenant tous les arguments de l'interpréteur (y compris le nom du programme lancé).

Vous pouvez donner des arguments au programme à l'exécution dans Spyder en allant dans le menu "Execution/Configurer"

```
In [1]: import sys
        print(sys.argv[0])
```

NB: N'hésitez pas à réutiliser des fonctions d'une partie à l'autre !

Analyse de texte : listes, dictionnaires, ensembles

Afin de gagner en familiarité avec les structures de données Python les plus utiles, vous allez développer des fonctions pour analyser le vocabulaire utilisé dans un texte. Vous trouverez sur ma page une version d'Alice aux pays des merveilles, mais vous pouvez récupérer le texte de votre choix. J'ai choisi un texte an anglais pour simplifier les problèmes d'accents.

http://www.irit.fr/P̃hilippe.Muller/alice_wonderland.utf8.txt

Première approche simple Définissez un ensemble de fonctions pour lire un texte d'un fichier, et compter le nombre d'occurrence de chaque mot. Il faudra bien sûr gérer la ponctuation, entre autres. Le but est de sortir les mots les plus intéressants utilisés dans l'oeuvre considérée.

Texte prétraité Vous avez du remarquer entre autres problèmes, que certains mots que l'on voudrait regrouper apparaissent sous des formes différentes (pluriel des noms, verbes conjugués), et que les mots fonctionnels (déterminants, prépositions par exemple) sont courants sans être très intéressants.

Vous trouverez dans le fichier http://www.irit.fr/~Philippe.Muller/alice_wonderland.utf8.conll une version du texte déjà prétraité, où chaque ligne correspond à une analyse prélable d'un mot du texte, avec sa forme telle qu'elle apparait dans le texte, son lemme (cad la forme normalisée correspondant à son entrée dans le dictionnaire), et une étiquette donnant sa catégorie: nom, verbe, déterminant, etc. Ecrivez de nouvelles fonctions pour refaire les analyses précédentes de façon plus simple avec ce fichier, en essayant de paramétrer le plus possible (faire varier les catégories à garder par exemple).

Analyse de séquences Pour avoir des informations plus intéressantes, on peut aussi regarder les séquences de 2 mots consécutifs. Ecrivez des fonctions pour compter toutes les séquences avec l'approche simple, et garder les plus intéressantes

Généraliser pour compter des séquences de longueur arbitraire (fixée à l'avance). On appelle ces séquences de n mots des n-grammes (bigrammes pour n=2, trigrammes pour n=3, etc).

Vous pouvez aller voir par curiosité l'inventaire historique fait par Google https://books.google.com/ngrams.

Bonus: tout combiner Généraliser la question précédente aux fichiers prétraités.