

W1-S4-C1-suggestions-de-lecture

December 15, 2014

1 Un peu de lecture

1.1 Complément - niveau basique

1.1.1 Le zen de python

Vous pouvez lire le zen de python à partir de python en important le module `this` comme ceci (n'oubliez pas d'évaluer cette cellule dans votre notebook)

```
In []: import this
```

Vous remarquerez que si vous évaluez cette cellule une deuxième fois, le texte disparaît; c'est le comportement normal de `import` comme on le verra en semaine 3. Redémarrez l'interpréteur python du notebook (*Kernel -> Restart*) pour revenir à l'état de départ, et évaluez la cellule à nouveau.

1.1.2 Documentation

- On peut commencer par citer l'[article de wikipedia sur python en français](#)
- La [page sur le langage en français](#)
- La [documentation originale](#) de python-2.7 - donc, en anglais - est un très bon point d'entrée lorsqu'on cherche un sujet particulier, mais (beaucoup) trop abondante pour être lue d'un seul trait. Pour chercher de la documentation sur un module particulier: le plus simple est encore d'utiliser google - ou votre moteur de recherche favori - qui vous redirigera dans la grande majorité des cas vers la page qui va bien dans, précisément, la documentation python.

1.1.3 Historique et survol

- L'article de wikipedia (en anglais) sur l'[historique du langage](#)
- Sur wikipedia, un article (en anglais) sur [la syntaxe et la sémantique de python](#)

1.1.4 Un peu de folklore

- Le [discours de Guido van Rossum à PyCon 2014](#)
- Sur youtube: le [sketch des monty python](#) d'où provient les termes `spam`, `eggs` et autres `beans` qu'on utilise traditionnellement dans les exemples en python plutôt que `foo` et `bar`
- et l'[article wikipedia correspondant](#), qui cite le langage python

1.2 Complément - niveau intermédiaire

1.2.1 Licence

- La [licence d'utilisation est disponible ici](#). Vous remarquerez qu'elle a été modifiée récemment pour la 2.7.8
- La page de la [Python Software Foundation](#), qui est une entité légale similaire à nos associations de 1901, à but non lucratif; elle possède les droits sur le langage

1.2.2 Le processus de développement

- Discussion sur les choix de conception du langage
- <http://python-history.blogspot.com>
- Comment les choix d'évolution sont proposés et discutés, au travers des PEP (Python Enhancement Proposal)
- [http://en.wikipedia.org/wiki/Python_\(programming_language\)#Development](http://en.wikipedia.org/wiki/Python_(programming_language)#Development)
- Le premier PEP décrit en détail .. le cycle de vie des PEPs
- <http://legacy.python.org/dev/peps/pep-0001/>
- L'index de tous les PEPs
- <http://legacy.python.org/dev/peps/>