

Un peu de lecture

Complément - niveau basique

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)

```
 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

1. Redémarrez l'interpréteur python du notebook (**Kernel** -> **Restart**) pour revenir à l'état de départ, et évaluez la cellule à nouveau.

Documentation

- On peut commencer par citer l'article de wikipedia sur python en français (http://fr.wikipedia.org/wiki/Python_%28langage%29)
- La page sur le langage en français (<https://wiki.python.org/moin/FrenchLanguage>)
- La documentation originale (<https://docs.python.org/2.7/>) 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.

Historique et survol

- L'article de wikipedia (en anglais) sur l'histoire du langage (http://en.wikipedia.org/wiki/History_of_Python)
- Sur wikipédia, un article (en anglais) sur la syntaxe et la sémantique de python (http://en.wikipedia.org/wiki/Python_syntax_and_semantics)

Un peu de folklore

- Le discours de Guido van Rossum à PyCon 2014 (<https://www.youtube.com/watch?v=EBRMq2loxsc>)

- Sur youtube: le sketch des monty python (<https://www.youtube.com/watch?v=anwy2MPT5RE>) 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 (http://en.wikipedia.org/wiki/Spam_%28Monty_Python%29), qui cite le langage python

Complément - niveau intermédiaire

Licence

- La licence d'utilisation est disponible ici (<https://docs.python.org/2/license.html>) . Vous remarquerez qu'elle a été modifiée récemment pour la 2.7.8
- La page de la Python Software Foundation (<https://www.python.org/psf/>), qui est une entité légale similaire à nos associations de 1901, à but non lucratif; elle possède les droits sur le langage

Le processus de développement

- Discussion sur les choix de conception du langage
 - <http://python-history.blogspot.com> (<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) ([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/> (<http://legacy.python.org/dev/peps/pep-0001/>)
- L'index de tous les PEPs
 - <http://legacy.python.org/dev/peps/> (<http://legacy.python.org/dev/peps/>)