Pièges à éviter

Quelques pièges à éviter !

Type de données

- travailler avec des nombres et ne pas mettre le point décimal s'ils ont une valeur entière les laissera dans le type 'int'.
- Ne pas confondre une liste contenant un nombre, et ce nombre.

Librairies

- Attention à ne pas avoir dans le répertoire courant où vous faite un programme un fichier dont le nom correspond à une librairie standard que vous souhaitez utiliser. C'est celui-ci que l'environnement Python tentera prioritairement d'utiliser.

Version de python

- les programmes en Python 3.x sont un peu différents de ceux en Python 2.7.y. La différence la plus fréquente est le passage de print à print(), parfois un changement de nom de librairie comme Tkinter en tkinter.

Indentations

- Ne pas mélanger des espaces et des tabulations ! Il est recommandé d'utiliser l'indentation par 4 espaces.

Symboles

- = pour l'instruction d'affectation et == pour l'opérateur de comparaison

l'affectation ne réalise pas une copie

Lorsqu'on travaille avec des objets listes (ou d'autres objets), l'instruction utilisant un signe “=” (nom=objet) ne réalise pas une copie, mais donne un second nom au même objet. Si on le manipule (le modifie) en utilisant une des appellations, on peut vérifier l'effet, en tapant les instructions suivantes dans l'interpréteur Python :

```
a=1
b=2
```

```
c=[]
c.append(a)
print(c)
d=c
print(d)
c.append(b)
print(d)
```

Sur des types numériques, l'effet est différent. Essayer ceci :

```
u=4
v=u
u=2*u
print(v)
```

Confusion dans les directives d'importation de librairies

Il y a 2 façons d'importer toutes les fonctionnalités définies dans un module “nomdemodule” :

- import nomdemodule
- from nomdemodule import *

Dans le premier cas, les fonctions seront appelables avec des noms tels que “nomdemodule.func1”, tandis que dans le deuxième mode, la même fonction sera utilisable avec le nom “func1”. La deuxième méthode permet également de n'importer qu'une seule fonction particulière au lieu de toutes celles qui sont dans le module (avec *). Exemple :

- from nomdemodule import func13

Pourquoi 2 façons de faire : l'importation via “from” est simple et permet de s'affranchir de l'écriture du namespace. Cependant, si plusieurs fonctions cohabitent avec des mêmes nom, cela va poser des problèmes car une seule, pas nécessairement “la bonne” sera effective !

référence : http://effbot.org/zone/import-confusion.htm

Suggestion : pour comprendre, essayez par exemple ces directives en mode interactif sur le module math.

Fix Me! : intégrer ces astuces