

Installer un environnement wxPython sur son PC.

Parce que Python devient de plus en plus incontournable, il ne faut pas le laisser de côté. Alors, voici un petit tutoriel pour installer ce merveilleux langage de programmation et tout ce qu'il faut pour utiliser wxPython dans les meilleures conditions.

La communauté Python est large et productive, il convient donc de faire attention à la compatibilité des versions des modules, bibliothèques et autres que vous installez. A l'heure où j'écris ces lignes, Python 3 est en bêta-test, wxPython vient de sortir pour Python 2.6 pour Windows et Mac, mais pas pour Linux et Boa est en 0.6.1. Nous sommes en novembre 2008.

Pour avoir un environnement de développement Python / wxPython, nous aurons besoin :

- du langage de développement Python
- de la bibliothèque wxPython
- de l'aide wxWidget C++
- de l'IDE/RAD Boa Constructor

Nous verrons ici que l'installation pour Windows, car je ne connais pas Mac et que sous Linux certaines distributions (Debian par exemple) ont Python, wxPython et Boa en standard. Reste peut-être une mise à jour à faire.

Les téléchargements :

Créer un répertoire « Téléchargement » pour recevoir les fichiers d'installations. Ce répertoire pourra être effacé à la fin.

1 – Téléchargez Python :

Python existe en 3 implémentations. Une convient à tout (CPython communément appelée Python), une pour les plateformes .net (IronPython) et une pour les machines virtuelles Java (JPython). Nous allons installer l'implémentation CPython. Nous trouverons sur le site plusieurs versions de Python, nous allons télécharger la version 2.5.2, car Boa Constructor n'accepte pas les versions supérieures pour le moment et que je ne l'ai pas testé avec Python 2.6 (affaire à suivre).

- Se connecter au site de Python [Le site](#)
- ou aller directement sur la page de téléchargement [La page](#)

Sélectionnez le téléchargement qui correspond à votre machine.

2 – Téléchargez wxPython :

wxPython est une interface de la bibliothèque graphique wxWidget de C++ pour Python. Elle en garde donc toute la puissance, la beauté, la logique, enfin tout ou presque. Comme Python et C++ sont très proches pour ce qui est de la logique de développement, wxPython a gardé l'aide de wxWidget du C++ qui convient très bien si l'on fait un petit effort de traduction.

- Se connecter au site de wxPython [Le site](#)
- ou aller directement sur la page de téléchargement [La page](#)

Sélectionnez le téléchargement qui correspond à votre machine et ***aussi à Python 2.5 et en version***

Unicode. Pour Windows XP c'est : **win32-unicode**.

Bien sûr, vous pouvez prendre la version win32-ansi, mais elle est moins performante. Il faut savoir que Python s'oriente vers l'Unicode et que Python 3 est fait pour l'Unicode, alors que Python 2.6 devrait permettre un passage de Python 2.x.x à Python 3 avec la meilleure facilité possible. Ensuite, pour Windows seulement, téléchargez : win32-docs-demos. Pour les autres OS il y a d'autres solutions sur le site.

3 – Télécharger Boa Constructor :

Boa est un EDI pour Python. Il permet la conception d'application complète et dans presque tous les domaines de développement. C'est aussi un RAD pour wxPython à la manière de C++ Builder de Borland. Bien que le développement de Boa est actif, le site ne l'est pas. Il fait référence à une ancienne version. Dommage !

- Se connecter au site de Boa Constructor [Le site](#)
- ou aller directement sur la page de téléchargement [La page](#)

Les installations :

Les installations sont très simples. Il faut juste respecter l'ordre qu'il convient, car tous ces logiciels seront liés entre eux.

1 – Installation de Python. Double cliquez sur le fichier : « python-2.5.2.msi », que vous avez dans votre répertoire « Telechargement ». Il va créer un répertoire : « [C:\Python25](#) » pour y installer tous ses fichiers.

2 – Installation de wxPython. Double cliquez sur le fichier : « wxPython2.8-win32-unicode-2.8.9.1-py25.exe » que vous avez dans votre répertoire « Telechargement ». Acceptez la licence et donnez le chemin d'installation suivant : [C:\Python25\Lib\site-packages](#). Dans Python, les bibliothèques internes sont installées dans ce répertoire. Choisissez « Full installation » et cliquez sur « Next ». Lorsque l'installation est fini une fenêtre s'ouvre avec 3 cases à cocher. Vous pouvez décocher « View README.win32.txt » **mais les autres doivent être cochés**. Cliquez sur « Finish ». Une console Dos s'ouvre pour la compilation des bibliothèques. Normalement, wxPython est bien installé.

3 – Installation de Boa Constructor. Double cliquez sur le fichier : « boa-constructor-0.6.1.bin.setup.exe », que vous avez dans votre répertoire « Telechargement ». Il va créer un répertoire : « [C:\Program Files\Boa Constructor](#) » par défaut pour y installer tous ses fichiers, mais vous pouvez lui donner le chemin que vous voulez. Il va aussi créer un répertoire caché : « [.boa-constructor-exe](#) », là, pas de choix pour vous.

4 – Installation du paquet démo et exemples de wxPython. Double cliquez sur le fichier : « wxPython2.8-win32-docs-demos-2.8.8.0.exe » que vous avez dans votre répertoire « Telechargement ». Acceptez la licence et donnez le chemin d'installation que vous voulez. Il créera un sous-répertoire : « [wxPython2.8 Docs and Demos\](#) ». Je l'ai mis dans : « [C:\Python25\Doc](#) », car c'est une doc Python.

5 — Ensuite, aller dans le répertoire de Boa : « [C:\Program Files\Boa Constructor](#) » et faite un raccourci de : « **Boa.exe** » et placez-le sur votre bureau. Faites de même avec le fichier « **demo.pyw** » que vous trouvez par exemple ici : « [C:\Python25\Doc\wxPython2.8 Docs and Demos\demo](#) », si vous l'avez installé dans le même répertoire que moi.

Le paramétrage :

Allez sur votre bureau et lancez Boa par le raccourci. Lorsqu'il est ouvert, vous avez 3 fenêtres devant vous. Celle du haut est la fenêtre principale. Vous y trouvez tous ce qui est nécessaire à la fabrication de vos applications. La fenêtre de gauche est l'inspecteur de propriété ou le débogueur, et la fenêtre de droite et l'éditeur et les commandes. Sur cette dernière fenêtre, l'Editeur, ouvrez le menu : **Option/Sélecteur d'interpréteur Python**. Il s'ouvre une fenêtre avec une sélection de chemin à droite et une sélection de versions de Python à gauche. Double cliquez sur la sélection de Python dans la fenêtre de gauche, Boa à détecter la version que nous avons installée. Dans le contrôle en dessous, le chemin de Python **doit** apparaître. Ensuite, cliquez sur « Ok ».

Pour Windows seulement :

Boa est une application encore en construction. Il y a de légers bugs. En voici un qui peut facilement être contourné.

Lorsque l'on navigue dans l'arborescence ou que l'on ouvre ou enregistre un fichier, Boa ouvre une boîte de dialogue pour nous dire qu'il ne comprend pas le codage de notre PC. Il bloque ainsi son utilisation, car il n'y a pas de solution trouvée dans Boa lui-même. Nous allons donc construire un fichier pour lui dire ce qu'il veut savoir.

Allez dans le menu démarrer : « Démarrer/Tous les programmes/Python 2.5/IDLE(Python GUI) ». Là s'ouvre la console Python. Vous pouvez, si vous le voulez, tester vos fonctions ici. Allez dans : « File/New Window » pour ouvrir un fichier vide. Copier le code suivant (attention, Python est sensible à la casse) :

- Soit pour coder en ASCII

```
import sys
sys.setdefaultencoding('iso-8859-1')
```

- Ou pour coder en Unicode

```
import sys
sys.setdefaultencoding('Latin-1')
```

« **import** » pour importer un fichier, c'est l'équivalent de « **include** » en C++. Ici on importe le fichier « **sys** » de la bibliothèque standard de Python.

A la ligne 2, nous passons un type d'encodage à notre session Boa, ici le coding iso-8859-1 pour la France ou Latin-1 pour Unicode

Enregistrer ce fichier dans un répertoire à votre convenance (je l'ai mis dans « C:\Python25 ») sous le nom de : « **BugCoging.py** » par exemple. Les fichiers de code Python ont l'extension « **.py** » comme le C++ à « **.cpp** ». Vous pouvez fermer les deux fenêtres de la console Python.

Ensuite, aller dans Boa. Dans la fenêtre de l'éditeur, il y a un onglet « **Explorateur** ». Comme son nom l'indique, vous y trouvez un explorateur de fichier. Déployez les Favoris et cherchez dans « Root » votre fichier « **BugCoging.py** ». Sélectionnez-le, et cliquez droit pour sélectionner « **Ajouter aux favoris** ». Cliquez sur « **Favoris** » et double-cliquez sur votre fichier. Votre fichier s'ouvre dans une fenêtre de l'éditeur. Allez dans le menu « **Fichier** » et cliquez sur « **Importez le module dans le shell** », puis sur « **Fichier/fermer** ».

Maintenant, à chaque fois que vous ouvrez Boa, suivez cette manipulation :

- Cliquez sur « **Favoris** » et double-cliquez sur votre fichier « **BugCoging.py** ». Votre

fichier s'ouvre dans une fenêtre de l'éditeur. Allez dans le menu « [Fichier](#) » et cliquez sur « [Importez le module dans le shell](#) », puis sur « [Fichier/fermer](#) ».

Voilà, vous pouvez développer des applications en Python avec wxPython.