Création et utilisation d'une version portable de Python pour Windows

Par Charles-Élie Gentil

Date de publication : 8 juillet 2013

Cet article a pour but de répondre à deux questions simples : pourquoi et comment créer une version portable de Python et comment l'utiliser. **Commentez**

I - Création de Python en version portable

I-A - Pourquoi créer une version portable de Python

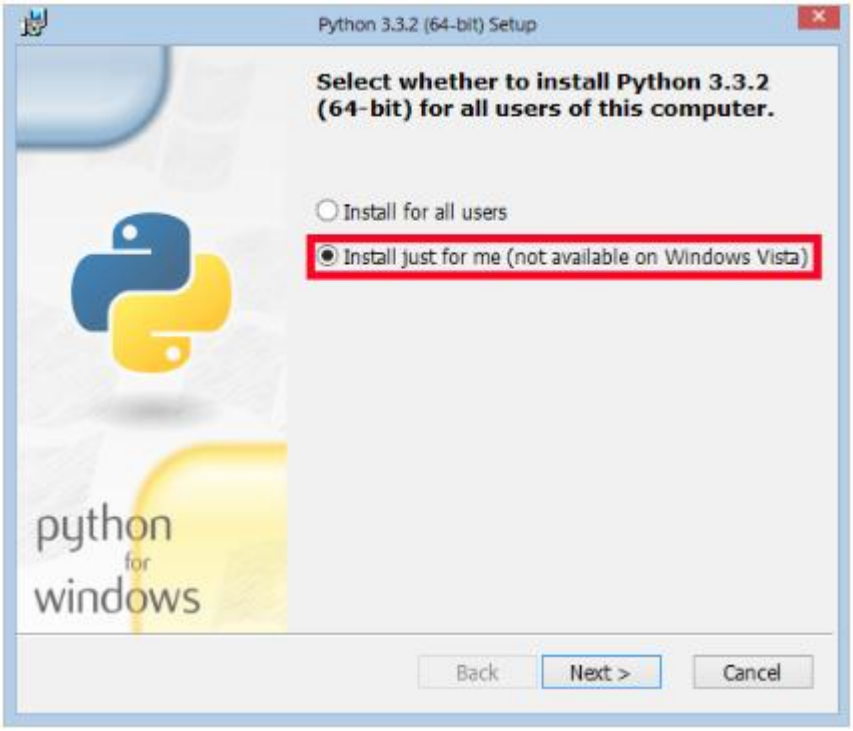
Il y a deux principales raisons au fait de vouloir avoir sa version portable de Python.
La première est de permettre au développeur de s'amuser et travailler de n'importe quel ordinateur. Qui ne rêve pas de pouvoir coder pendant que belle-maman et chérie font la vaisselle ?
Gros avantages :

- pas besoin d'installer quoi que ce soit sur le PC hôte ;
- si le PC hôte a déjà Python d'installé, garantie d'avoir les bibliothèques nécessaires.

La deuxième raison peut être liée au déploiement de votre application. En effet et en théorie, toute application a pour but d'être déployée sur diverses machines. Une solution « simple » est de créer un exécutable.
Pour cela, de nombreux outils sont à votre disposition dont les plus connus sont cx_freeze, PyInstaller, Py2exe. Vous trouverez toute la documentation sur ceux-ci sur le site developpez.com.
Mais vous pourriez aussi souhaiter distribuer votre programme en version source avec tout ce que cela entraîne (réfléchissez bien à ce point le moment venu).
Dans ce cas et puisque rien ne peut vous garantir que vos utilisateurs cibles ont une installation Python nécessaire à votre projet, il vous faudra a minima fournir les bibliothèques nécessaires voire plus comme l'interpréteur complet et tous les fichiers nécessaires à son bon fonctionnement.

I-B - Comment créer une version portable de Python

Bon, là c'est la partie la plus dure. Non, je rigole. En fait, c'est même trop simple pour être vrai. Souvenez-vous lorsque vous avez installé Python sur votre PC pour la première fois, vous avez eu cette fenêtre :



Eh bien sachez que :

- le premier vous installera une version de Python qui sera commune à tous les utilisateurs de votre ordinateur ;

- le deuxième installera une version de Python qui sera propre à chaque utilisateur.

Vous allez me dire, « là on enfonce des portes ouvertes ». Presque…
Installer une version de Python qui sera propre à chaque utilisateur revient en d'autres termes à installer une version autonome. Eh oui aussi simple que ça ! Au moment d'installer Python, faites-le sur une clé USB et vous aurez une version portable (un copier-coller de cette version peut aussi faire l'affaire).
Malgré tout pour optimiser ceci tout n'est pas réellement fini.
Dans le cas d'une version « Production », pensez à rajouter les bibliothèques dont vous avez besoin. La manière la plus simple étant de les installer de la manière la plus traditionnelle en pensant de bien renseigner le Python cible.
Dans une version « Distribution », vous pouvez aussi envisager de supprimer des bibliothèques non utilisées.

II - Utilisation de votre version portable

Considérant que toutes les bibliothèques nécessaires sont installées, il vous suffit d'indiquer à votre script où se situe l'interpréteur ou de lancer votre programme via une commande externe en indiquant les arguments nécessaires. Personnellement, je préfère utiliser un .bat. Par exemple en imaginant que :

- votre version portable de Python soit dans le dossier F:\MyProject\MyAppPy ;
- votre interpréteur se trouve à l'adresse F:\MyProject\MyAppPy\python.exe ;
- votre script principal à l'adresse F:\MyProject\MyApp\Main.py,

votre .bat qui sera dans F:\MyProject ressemblerait à ça :

```
MyAppPy\python.exe MyApp\Main.py
```

III - Conclusion

Cet article doit vous permettre d'envisager une autre manière de programmer et de déployer vos codes Python avec peut-être de nouvelles perspectives et idées de développement.

IV - Remerciements

Merci à **Thibault Cuvelier** pour son inconditionnelle patience, motivation et soutien et à **Claude Leloup** pour sa relecture orthographique.