Chapitre 3
Présentation de l'environnement Python

1. Python, troisième du nom

Si vous découvrez maintenant l'environnement Python ou que vous l'avez fait après le 1er janvier 2020, alors il y a de grandes chances pour que vous ne vous posiez même pas la question du choix entre Python 2 et Python 3 : vous allez naturellement utiliser la version la plus récente. Mais ceux dont la connaissance du langage précède cette date ont peut-être remarqué cette dualité, cette coexistence qui a pu être source de confusion.

Python, au fil du temps, a accumulé quelques tares qu'il a été nécessaire de corriger. Parmi ces défauts, citons la redondance d'informations (plusieurs façons d'obtenir un certain résultat), la gestion des chaînes de caractères peu optimale, une manière désuète de déclarer des classes, etc. Ainsi, en décembre 2008, une version 3.0.0 de Python a vu le jour, en même temps qu'une version 2.6.

Cette nouvelle version majeure de Python n'est pas compatible avec Python 2. C'est-à-dire qu'un programme écrit avec une version 3.0.0 ou supérieure de Python ne peut pas fonctionner avec un interpréteur Python version 2 (et vice versa). Cela peut être problématique si jamais un projet Python 3 doit utiliser une bibliothèque externe écrite en Python 2. À cause de l'incompatibilité de version, cette dépendance ne pourra pas être utilisée, mettant un possible frein à l'avancée du projet. Là où jadis le choix de version majeure de Python était une question épineuse, depuis 2020 elle ne se pose plus : Python 3 est désormais la seule version maintenue de Python.

Lorsque Python 3 est sorti, la communauté Python annonça la mort programmée de Python 2 pour 2015 et communiqua grandement sur l'importance de migrer tous les projets Python vers cette nouvelle version majeure. Le réveil ne s'est pas vraiment effectué et, en 2014, devant l'étendue des importants et populaires projets Python restés sur la version 2, un sursis a été établi jusqu'en 2020. Aujourd'hui, les systèmes d'opération Linux et macOS ne proposent plus Python 2 dans leur installation de base, et Windows ne l'a jamais fait.

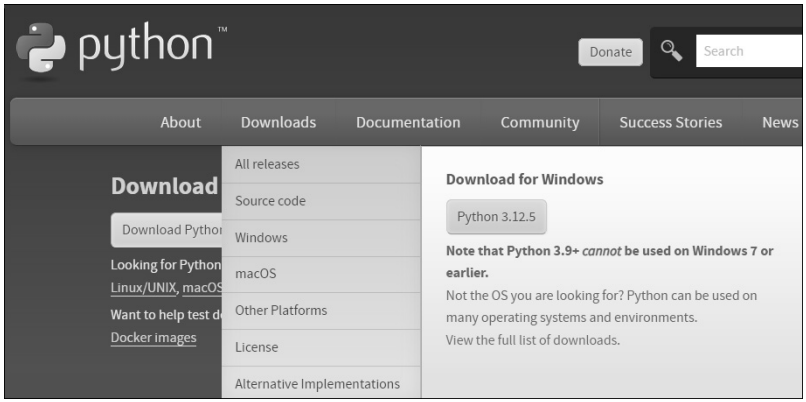
Remarque

Ces faits peuvent paraître relever de l'anecdote historique, mais étant donné l'énorme popularité de Python avant sa version 3, une grande quantité de bibliothèques ont été écrites dans cette version qui n'est pas compatible avec la version 3, actuelle et maintenue. Connaître ce pan d'histoire de Python, c'est rester sur ses gardes lorsqu'on veut inclure une bibliothèque dans son projet, et bien vérifier qu'elle est compatible avec Python 3.

2. Installation

2.1 python.org

Rendez-vous sur https://www.python.org/downloads/. Le site de Python devrait vous proposer automatiquement un fichier correspondant à votre système d'exploitation.



2.2 Windows

Par défaut, le site vous propose de télécharger un fichier d'« installation par le Web », qui requiert une connexion internet. En naviguant vers la page de téléchargement d'une version précise de Python (dont l'adresse est généralement de la forme https://www.python.org/downloads/release/python-3125/ pour la version 3.12.5), vous aurez le choix entre :

– un fichier compressé contenant une version de Python utilisable sans avoir à l'installer (cela est pratique si vous n'avez pas les droits suffisants pour effectuer une installation standard) ;
– un installateur « classique » ;
– l'installateur par le Web.

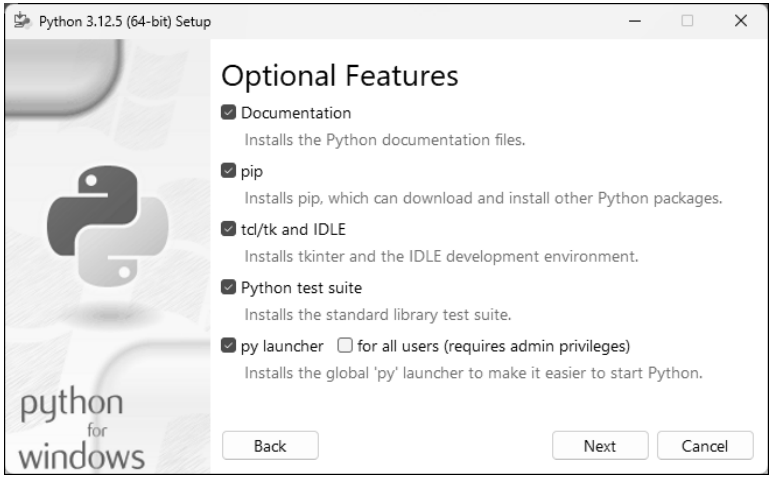
Lors de l'installation, vous aurez à décider de quelques options dont voici le détail :

En bas de cet écran, deux options s'offrent à vous :

– **Use admin privileges when installing py.exe** (utiliser les privilèges administratifs lors de l'installation de py.exe) détermine si le programme **Python (py.exe)** est accessible pour tous les utilisateurs du système ou uniquement pour l'utilisateur actuel.

– **Add python.exe to PATH** (ajouter ou non l'exécutable dans le PATH). Le PATH est une variable d'environnement contenant généralement plusieurs répertoires. Lorsqu'un utilisateur veut exécuter une commande (typiquement `python` ou même `dir`), le système d'exploitation va chercher dans les répertoires du PATH si l'exécutable désiré est présent. Si c'est le cas, alors la commande fonctionnera. Sinon, un message d'erreur apparaîtra indiquant que l'exécutable n'a pas été trouvé. Si vous comptez utiliser Python en ligne de commande, il vaut mieux cocher cette option. Si vous pensez n'utiliser que des IDE pour travailler, vous pouvez la laisser décochée.

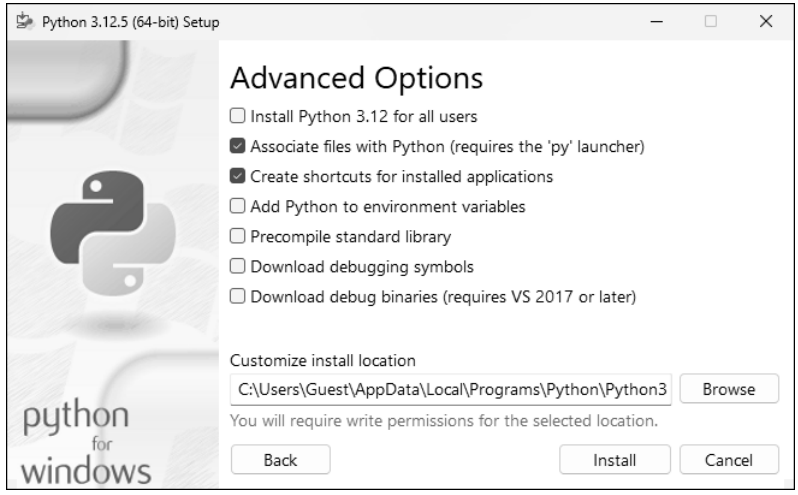
Si les choix de destination et de contenu à installer vous conviennent, vous pouvez lancer l'installation. Sinon, une personnalisation des options d'installation est nécessaire :



– **Documentation** : installer la documentation peut être intéressant si vous êtes souvent sans connexion Internet (en déplacement par exemple).

– **pip** : pip est un outil simple d'utilisation pour installer des bibliothèques plutôt que de les récupérer et de les installer manuellement.
– **tcl/tk and IDLE** : tcl/tk est un module pour générer des interfaces graphiques ; IDLE est un environnement graphique de développement. Cochez cette case selon les besoins de votre projet et votre méthode de travail.
– **Python test suite** : la suite de test Python est un ensemble de modules pour tester efficacement votre application, et devrait être installée sans hésiter.
– **py launcher** : le launcher Python a été explicité précédemment.
– **for all users (requires admin privileges)** permet d'installer Python pour tous les utilisateurs de la machine, mais les droits administrateurs sont requis.

Après le contenu de l'installation, voici les paramètres :

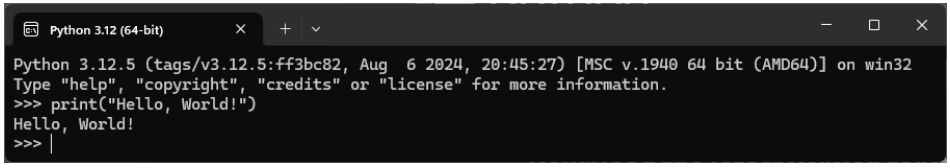


Pour les quelques options qui ne sont pas explicites :

– **Add Python to environment variables** (ajouter Python aux variables d'environnement) permet au système de savoir où se trouvent les exécutables Python, les bibliothèques, etc.

– Lorsque vous déboguez votre application, Python génère des fichiers de débogage dans lesquels se trouvent votre code ainsi que des informations supplémentaires pour vous faciliter la démarche. Afin de pouvoir naviguer dans les modules de base de Python lors d'un débogage, il faut installer les symboles de débogage de ces modules, via l'option de téléchargement des symboles de débogage (**Download debugging symbols**) et téléchargement des binaires de débogage (**Download debug binaries (requires VS 2017 or later)**).

▶ Cliquez sur **Install** et, après quelques secondes, Python est installé. Pour tester votre installation, ouvrez une fenêtre de commande (sous Windows : touche **[Windows] + R**, puis `cmd`) et entrez la commande `python`. Vous entrez dans l'environnement de l'interpréteur Python. Vous pouvez taper tout le code Python que vous voulez, mais pour ce test d'installation, un simple texte affiché à l'écran suffira : `print('Hello, World!')`.



2.3 macOS

À partir de macOS Monterey (12.3), publié en octobre 2021, Python 3 est installé par défaut avec le programme python3. Le suffixe 3 est un reliquat de l'époque où les deux versions de Python étaient installées côte à côte.

▶Pour vérifier votre installation, ouvrez l'application « Terminal » et tapez :

```
$> python3 --version Python 3.12.5

$> python3
Python 3.12.5 (main, Aug  6 2024, 19:08:49) [Clang 15.0.0
(clang-1500.3.9.4)] on darwin
Type "help", "copyright", "credits" or "licence" for more information.
>>> print("Bonjour Python!") Bonjour Python!
```

Le fait d'avoir à suffixer la version au nom de l'exécutable peut être problématique si jamais des scripts utilisent la commande `python` (sans suffixe). Il peut donc être nécessaire d'indiquer à votre shell que la commande `python` pointe vers l'exécutable `python3`. Pour ce faire, il vous faut créer un alias dans votre fichier de configuration de shell, qui est `.bashrc` par défaut et se situe à la racine de votre répertoire utilisateur :

```
$> echo "alias python=/usr/local/bin/python3.12.5" >> ~/.bashrc
```

Après avoir exécuté la commande ci-dessus, il convient de vérifier que l'alias fonctionne :

```
$> python --version Python 3.12.5
```

2.4 Unix/Linux

La majorité des distributions populaires comportent déjà par défaut Python 3. Il n'y a donc pas d'installation requise. Si toutefois vous deviez installer Python, le site de Python n'offre pas de fichier d'installation. Selon la distribution sur laquelle vous travaillez, il existe très certainement un outil de gestion de logiciels qui vous permettra d'installer Python 3 :

```
$> apt install python3 # Sous Ubuntu
$> emerge --ask dev-lang/python # Sous Gentoo
```

3. Outillage

3.1 pip

L'exécutable pip permet d'installer des bibliothèques Python en ligne de commande. Ceci est très utile car il peut être assez fastidieux de se rendre sur le site web hébergeant la bibliothèque, de la télécharger et de la déposer dans l'endroit adéquat afin que Python puisse y accéder.

Tout cela est automatiquement effectué par pip, qui est d'ailleurs l'outil officiel du groupe de travail chargé de maintenir les principaux projets de bibliothèques Python (le PyPA : *Python Package Authority*). Ce groupe de travail alimente un index officiel des modules disponibles auquel vient se connecter pip. C'est donc le meilleur moyen de s'assurer d'avoir des dépendances parfaitement stables et à jour.

À l'instar de l'exécutable `python`, il faudra peut-être utiliser la commande `pip3` et non `pip` afin d'appeler l'installeur dédié à Python 3 (notamment sous macOS ou Ubuntu). Vérifier la version en appelant la commande `pip --version` peut préserver de quelques maux de tête.

```
$> pip install pyside6 # Installe la bibliothèque Qt pour Python
```

pip offre également d'autres commandes pratiques pour gérer les bibliothèques installées sur votre système :

– `pip uninstall` pour désinstaller des bibliothèques dont vous n'avez plus l'utilité ;

– `pip list` pour obtenir la liste des bibliothèques installées ;

– `pip help` pour afficher toutes les autres options disponibles.

3.2 IDLE

IDLE (*Integrated Development and Learning Environment* : environnement d'apprentissage et de développement intégré) est un IDE basique pour Python et disponible sur tous les systèmes d'exploitation. Il offre plusieurs fonctionnalités dont :

– la coloration syntaxique ;

– l'autocomplétion ;

– un débogueur ;

– une indentation « intelligente ».

Cependant, si cet outil est suffisant pour débuter en Python, il est clairement limité pour une utilisation plus avancée. Il manque cruellement d'ergonomie, des fonctionnalités importantes sont absentes de l'interface graphique (numérotation des lignes, navigation facilitée entre les fichiers, personnalisation des couleurs et polices, etc.).