

R1.13 Introduction à *Python*

Getting started

Th. Godin, T. Ferragut, M. Le Lain

Ce guide vous aidera à installer Python et à commencer à l'utiliser. Nous allons couvrir l'installation de Python, la configuration de l'environnement de développement, et quelques concepts de base.

On va utiliser plusieurs méthodes qu'on distinguera selon la couleur :

bash

editeur de texte

interpreteur console

Ce tutoriel est conçu principalement pour Linux. Des informations sur les environnements sous Windows ou Mac seront donnés mais plus succinctement.

Il existe deux sous-langages de Python : Python 2 et Python 3. Python 3 aurait du remplacer complètement la version 2 mais elle continue cependant d'être utilisée. Nous n'utiliserons que la version 3 mais vous pourriez rencontrer du Python 2 en entreprise ou sur internet. La compatibilité n'est pas totale, mais on traduit généralement facilement du 2 vers le 3.

En plus du cours, un coup d'œil à la page wikipedia [https://fr.wikipedia.org/wiki/Python_\(langage\)](https://fr.wikipedia.org/wiki/Python_(langage)) est conseillé.

Pourquoi Python en plus de Java ?

Théoriquement parlant, Python et Java ont la même puissance (on dit qu'ils sont Turing complets) et on tout ce que l'on réussit à programmer en Java pourrait l'être en Python et vice-versa ¹. En pratique Python est un langage très populaire pour les sciences des données, l'IA et le prototypage. Ses performances en temps sont généralement moins bonnes que Java ou C. L'utilisation de bibliothèques spécialisées (Cython) permet de pallier largement à ce problème, mais la simplicité et la présence de nombreuses bibliothèques spécialisées en font un des langages de programmation les plus populaires.

Pour l'instant vous avez principalement utilisé Java pour de l'algorithmique, la traduction vers le Python sera donc assez immédiate. Plus tard l'emploi de techniques plus avancées rendra le passage d'un langage à l'autre plus délicat.

Mise en place

Créez une arborescence R1.13/TP1 et placez-vous dedans. On lancera les terminaux depuis ce point.

Dans le répertoire TP1 Créez un fichier Python (`hello.py`) avec votre éditeur de texte ou IDE. Écrivez un programme simple, par exemple :

1. Attention tout de même, ici "*pourrait*" ne signifie pas "*pourrait tout aussi facilement*". Parmi les "langages" Turing-complets et donc théoriquement aussi expressifs que Java et Python, on trouve Excel, Minecraft, Démineur ou encore Magic the Gathering. L'écriture des programmes à venir dans ces langages est laissée en exercices aux étudiant-es les plus curieux-es et motivé-es ; <https://fr.wikipedia.org/wiki/Turing-complet>

```
print("Hello ,_World!\_n\_t\t\_—send\_from\_file")
```

Enregistrez le fichier.

1 Installation de Python

1.1 Téléchargement

A priori votre distribution Linux inclut Python

```
python3 --version
```

Rendez-vous sur le site officiel de Python à l'adresse <https://www.python.org/downloads/> et téléchargez la dernière version de Python pour votre système d'exploitation.

Une alternative possible (sous Windows et Mac) est d'installer **Anaconda** (se reporter au site officiel pour les détails).

1.2 Installation

Suivez les instructions d'installation fournies sur le site Web. Assurez-vous d'inclure Python dans le chemin d'accès (option souvent disponible lors de l'installation).

1.3 Installation de Packages

Python permet l'utilisation de nombreuses bibliothèques. Utilisez **pip**, le gestionnaire de packages de Python, pour installer des packages. Par exemple, pour installer **numpy**, exécutez la commande suivante dans le terminal ou l'invite de commande :

```
pip install numpy
```

Si vous utilisez Anaconda, il faudra entrer dans le prompt Anaconda

```
conda install numpy
```

2 Premier Programme Python

2.1 Interpréteur-Console

Python est un langage interprété. Pour nous cela signifie qu'il est possible de le lancer depuis un terminal et l'utiliser directement, comme une calculatrice (en plus puissant)

Pour lancer l'interpreteur, on utilise la commande **python3** (ou juste **python**, selon votre installation)

```
python3
```

Le terminal lance alors python en mode interprété. On peut directement coder et executer dans la console

```
2 + 2
```

```
print("Hello_(from_the_interpreter),_Word!")
```

```
a = 1+1  
print("un_calcul_complexe:_", a*5 - 3 )
```

Remarquez que la fonction d’affichage `print` a un comportement et une syntaxe légèrement différentes de Java.

On peut utiliser des programme écrits ailleurs :

```
exec(open("hello.py").read())
```

Et on quitte (proprement) l’interpréteur par la fonction `exit` :

```
exit()
```

2.2 Exécution au terminal

En R1.01 vous avez appris à compiler et exécuter un fichier Java en utilisant `javac` puis `java`. En Python l’équivalent se fait en une commande, en tapant :

```
python nom_du_fichier.py
```

Vous devriez voir la sortie terminal `Hello, World!`.

C’est, avec les IDE, la méthode la plus commune quand on gère un projet un peu conséquent.

2.3 Jupyter-notebook

La troisième alternative est une méthode à la popularité croissante qui permet de créer des documents interactifs mêlant texte (basé sur markdown et html) et Python. C’est un très bon outil pour les projets de taille modeste et pour les présentations, nous l’utiliserons beaucoup à l’IUT.

L’installation sous Linux se fait part :

```
sudo apt-get install jupyter-notebook
```

ou bien via pip :

```
pip install jupyter
```

Sous Mac et Windows, on passera par Anaconda.

On lance jupyter depuis un terminal² (dans le dossier TP1) :

```
jupyter-notebook
```

Cela ouvre un onglet dans votre navigateur web. Comme vous n’avez pas de fichier `.ipynb` (l’extension Interactive PYthon NoteBook), vous ne devriez voir que `hello.py`. Jupyter permet de modifier ce fichier comme un éditeur de texte mais ce n’est pas son intérêt.

Cliquez sur "Nouveau" ("New") en haut à droite et sélectionnez "Python3". Cela crée et ouvre un nouveau fichier `Untitled.ipynb`, modifiez son nom (dans le navigateur).

2. Là encore, sous Mac et Windows, on passera par Anaconda.

Télécharger maintenant le fichier `R1-13_Intro_Jupyter.ipynb` disponible sur Moodle, placez le dans le répertoire TP1 et lancer le (depuis le navigateur Jupyter).

Normalement vous êtes prêt-e pour le TP1 !