Outils pour enseigner la programmation Python au lycée

Table des matières

1 Qu'est-ce que Python ? 2

2 Outils et installations 2

2.1 EduPython : Pour une installation simple et rapide . . . 2
2.2 Miniconda (ou Anaconda) : Pour une installation plus moderne . . . 3
2.3 Pyzo : Pour compléter Miniconda (ou Anaconda) . . . 4
2.4 Installation de Pyzo . . . 4

3 Librairies complémentaires 5

4 Spécificités du langage Python 5

4.1 L'indentation . . . 5
4.2 Déclaration des variables . . . 6

5 Intégrer du code Python dans vos documents 6

5.1 Avec un traitement de texte (LibreOffice, OpenOffice, Word…) et Notepad++ . . . 6
5.2 En pdf avec LaTeX et le package listings . . . 7
5.3 En html avec Jupyter . . . 7
5.4 Installation de Jupyter . . . 8

6 Exemples d'activités 9

6.1 Bouge la balle ! . . . 9
6.2 Le nombre mystère . . . 10
6.3 Calcul mental . . . 10
6.4 Ensorceler un nombre . . . 10
6.5 Un peu de probabilités et d'échantillonnage . . . 10

1 Qu'est-ce que Python ?

Python est un langage multiplateforme, c'est-à-dire disponible sur plusieurs architectures (compatible PC, tablettes, smartphones, ordinateur low cost Raspberry Pi...) et systèmes d'exploitation (Windows, Linux, Mac, Android...).

Le langage Python est gratuit, sous licence libre.

Il existe deux versions de Python : Python2 et Python3. Ces deux versions sont incompatibles. Même si la version 2 reste encore majoritaire, il faut bien avancer...

Python permet de faire :

- Du calcul scientifique
- Du traitement d'images et de sons
- Des applications et des jeux vidéo (Kivy, Pygame, TKinter)
- Des applications web et réseau
- Des communications USB, bluetooth, Wi-Fi (pour piloter des robots par exemple)

2 Outils et installations

Pour coder en Python, il faut un éditeur (pour taper les programmes) et un interpréteur (pour exécuter le code). Il y a plusieurs façons d'installer un environnement Python.

L'installation classique n'est pas recommandée sauf si vous savez ce que vous faites. En effet, il sera difficile par la suite d'ajouter des modules complémentaires.

Les distributions Miniconda3 ou Anaconda permettent de gérer plus facilement l'ensemble des modules. Accompagnées de Pyzo, vous obtiendrez un environnement Python moderne, complet et simple d'utilisation.

Miniconda3 (ou Anaconda) et Pyzo fonctionnement sous Linux, Windows et OSX.

Il existe des distributions « clé en main » et portables permettant de travailler sous un environnement Python : EduPython. L'installation et l'utilisation sont très simples. Un bémol, elles sont basées sur des versions 3.4 de Python pour lesquelles certains modules récents peuvent être incompatibles.

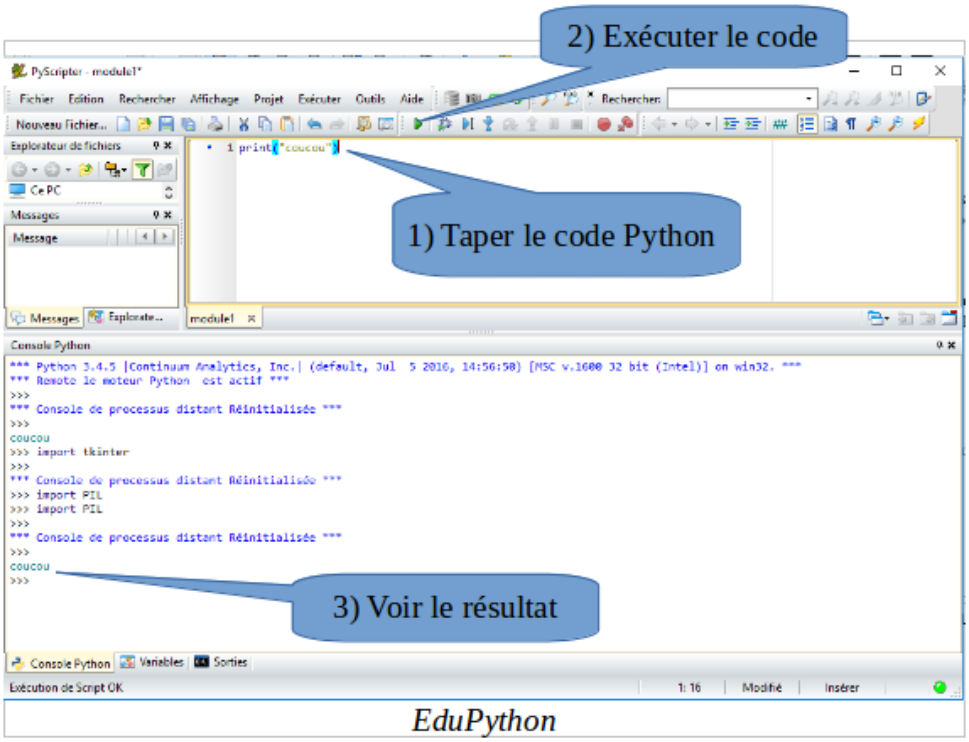
Pour résumer :

- EduPython : Pour une installation simple et rapide
- Miniconda3 (ou Anaconda) et Pyzo : Pour une installation moderne

2.1 EduPython : Pour une installation simple et rapide

EduPython est une distribution clé en main et portable pour programmer avec vos élèves sous un environnement Python3. De nombreux modules sont déjà présents notamment un module pour le lycée accompagnés d'une documentation riche.

Installation : Télécharger puis installer EduPython.



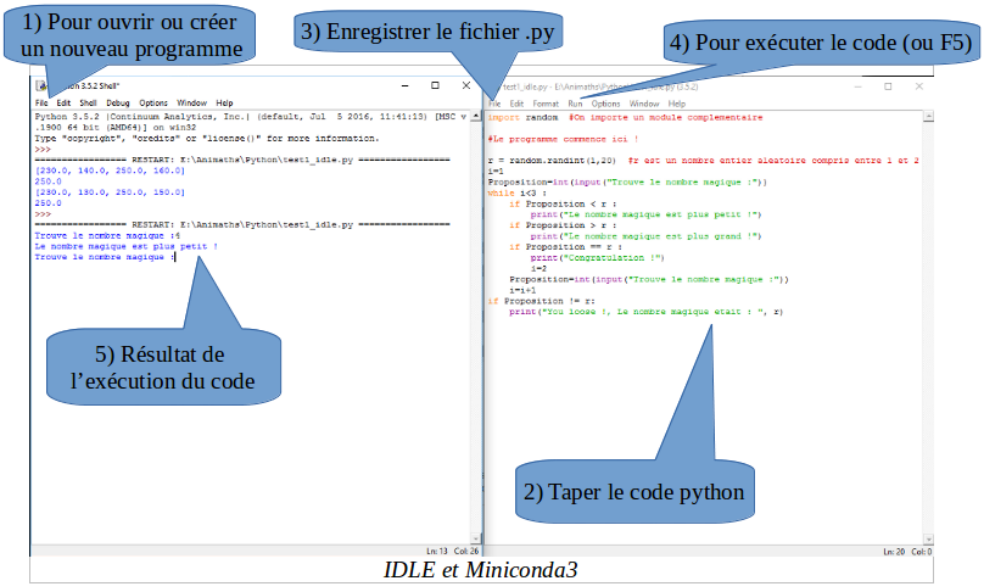
EduPython

2.2 Miniconda (ou Anaconda) : Pour une installation plus moderne

Simples et modernes, les distributions Miniconda3 et Anaconda permettent d'obtenir rapidement des environnements Python. Le gros avantage de ces deux distributions est de pouvoir installer facilement de nouvelles librairies (Jupyter, numpy, kivy, Image...) sans passer des heures à chercher des versions compatibles sur le web.

Miniconda3 est une distribution plus légère. Cela permet de prendre beaucoup moins de place sur le disque. Vous pourrez alors installer des librairies complémentaires. La distribution Anaconda, plus lourde, contient déjà de nombreuses librairies.

Il sera nécessaire d'utiliser un éditeur pour taper vos programmes. Un simple bloc-notes peut suffire mais ce n'est pas recommandé. Miniconda3 et Anaconda contiennent un éditeur : IDLE *(Dans Miniconda3/Scripts)*.



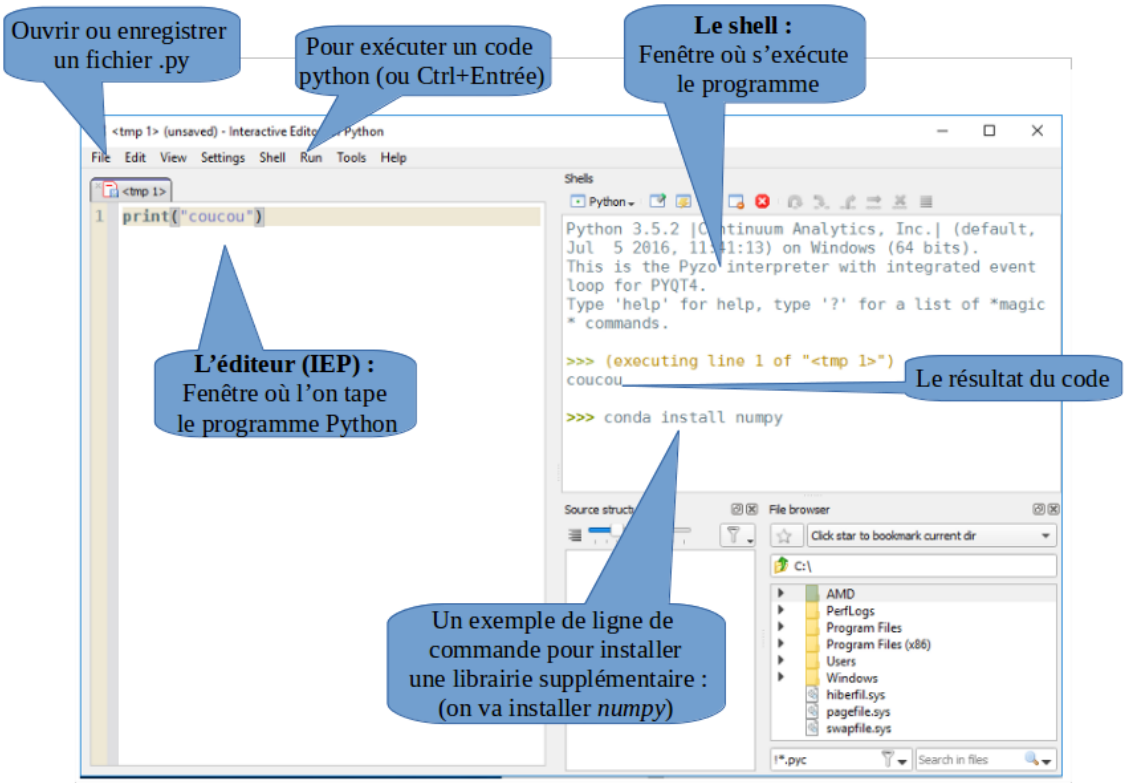
IDLE et Miniconda3

Pour installer des librairies complémentaires, il faudra passer par un terminal. Cependant, il existe un éditeur plus performant qui permet de gérer de multiples choses (éditer le code, l'exécuter, installer des librairies...) : Pyzo.

2.3 Pyzo : Pour compléter Miniconda (ou Anaconda)

Pyzo est basé sur un environnement Python. Vous pourrez éditer vos codes Python, les exécuter, installer facilement des librairies complémentaires et tout cela sur la même fenêtre.

Pyzo fonctionne avec Miniconda3 (ou Anaconda).



Pyzo et Miniconda3

2.4 Installation de Pyzo

- Télécharger puis installer Miniconda3
- Télécharger puis installer Pyzo
- Lancer Pyzo (*pyzo.exe* sous Windows)
- Dans la fenêtre des Shells, Pyzo vous demande d'indiquer un environnement Python présent sur votre ordinateur. Indiquer le fichier *python.exe* dans le répertoire Miniconda3 précédemment installé.

Fonctionnement :

- Taper un premier code dans la fenêtre de l'éditeur : `print("coucou")`
- Appuyer sur les touches Ctrl + Entrée pour exécuter le code.

3 Librairies complémentaires

Afin de compléter votre installation Miniconda3 et Pyzo, vous pouvez ajouter des modules supplémentaires.

Pour installer `numpy` par exemple :

- Ouvrez Pyzo.
- Dans le shell, tapez : `conda install numpy` 

.

Après avoir validé votre choix, le module s'installera automatiquement.

Par exemple, pour le calcul scientifique, vous pouvez installer `scipy, pyqt, matplotlib, pandas, sympy` :

```
conda install scipy pyqt matplotlib pandas sympy
```


4 Spécificités du langage Python

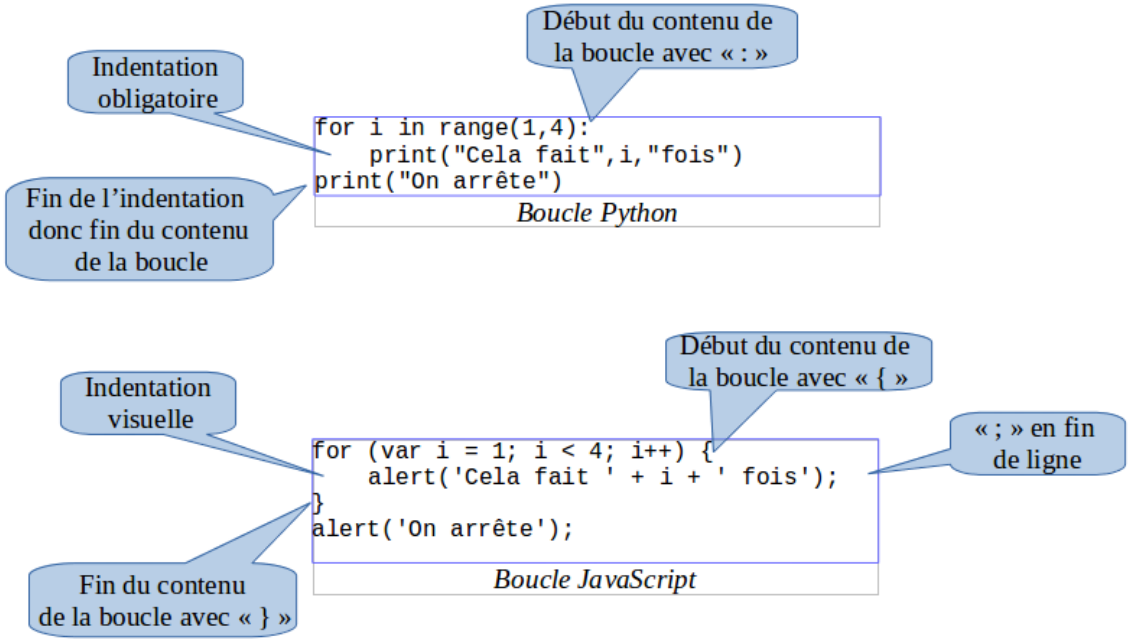
4.1 L'indentation

En informatique, l'**indentation** consiste en l'ajout de tabulations ou d'espaces dans un fichier, pour une meilleure lecture et compréhension du code. Cela permet de délimiter visuellement les blocs de codes comme des boucles, des fonctions, des instructions conditionnelles...

Dans la plupart des langages, cette indentation est simplement recommandée.

En Python, l'indentation est obligatoire.

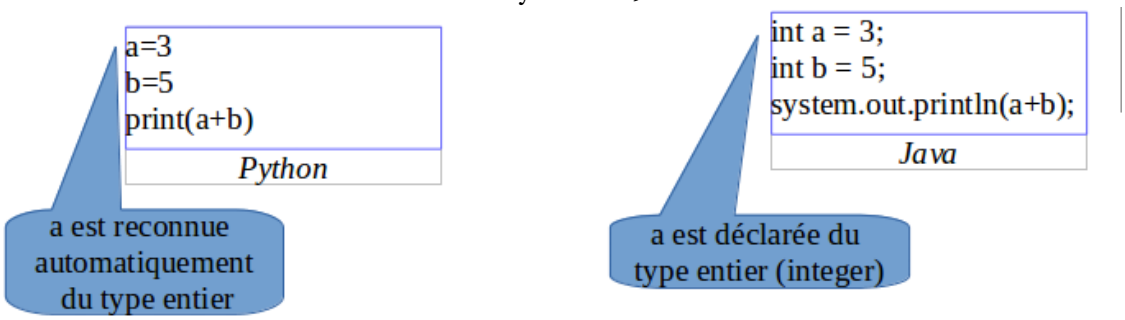
Exemple pour une boucle `for` en Python et JavaScript :



4.2 Déclaration des variables

En python, une variable a un type dynamique, c'est-à-dire qu'elle prend le type de l'objet qu'on lui affecte. Autrement-dit, on ne déclare pas explicitement une variable, il suffit de l'initialiser pour qu'elle prenne le bon type (entier, liste, chaîne de caractères...). Ce n'est pas le cas de tous les langages. Par exemple, en java, il est nécessaire de spécifier le type de variable en la déclarant.

Exemple pour une addition de deux variables en Python et Java :



5 Intégrer du code Python dans vos documents

Toute la difficulté se situe dans le « copier/coller ». En effet, les élèves peuvent avoir besoin de copier un programme dans un document (pour le corriger ensuite par exemple). L'**indentation** en Python étant obligatoire, il faut pouvoir la conserver lors d'un « copier/coller ».

Voici donc trois méthodes :

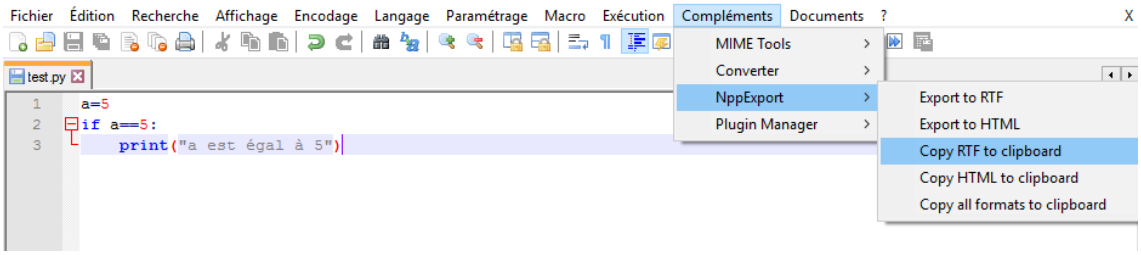
- Avec un traitement de texte (LibreOffice, OpenOffice, Word...) et Notepad++
- En **pdf** avec LaTeX et le package **listings**
- En **html** avec **Jupyter**

5.1 Avec un traitement de texte (LibreOffice, OpenOffice, Word...) et Notepad++

Avec des documents du type odt, doc..., le copier/coller ne pose pas de problème. Hélas, il va peut-être falloir verrouiller vos documents et surtout, comment avoir de jolies couleurs ?

Télécharger et installer Notepad++

Ouvrir votre fichier *truc.py* avec Notepad++ puis dans l'onglet « Compléments », « NppExport », sélectionner « copy RTF to clipboard ». Il ne reste qu'à coller le contenu de votre presse-papier dans votre document LibreOffice (ou autre...).



Attention, si vous exportez votre document en pdf, vous conserverez les couleurs mais vous perdrez l'indentation lors d'un copier/coller.

5.2 En pdf avec LaTeX et le package listings

La difficulté est la différence de gestion des caractères blancs (espaces) par les lecteurs pdf. En effet, une méthode va fonctionner sur un visionneur mais pas sur un autre.

Il existe aussi un problème pour copier/coller les lignes vides. Dans l'exemple suivant, une ligne vide sera remplacée par un #.

Voici un exemple :
(Visionneur : AdobeReader)

Exemple de source tex : Exemple de fichier tex

Le résultat : Le résultat pdf

```
#Fonction  ensorcelle :
def ensorcelle(x):
    resultat=-1/x+1
    return resultat

#Début  du  programme:
a=int(input("Entrez un nombre : "))
a=ensorcelle(a)
print(a)
```

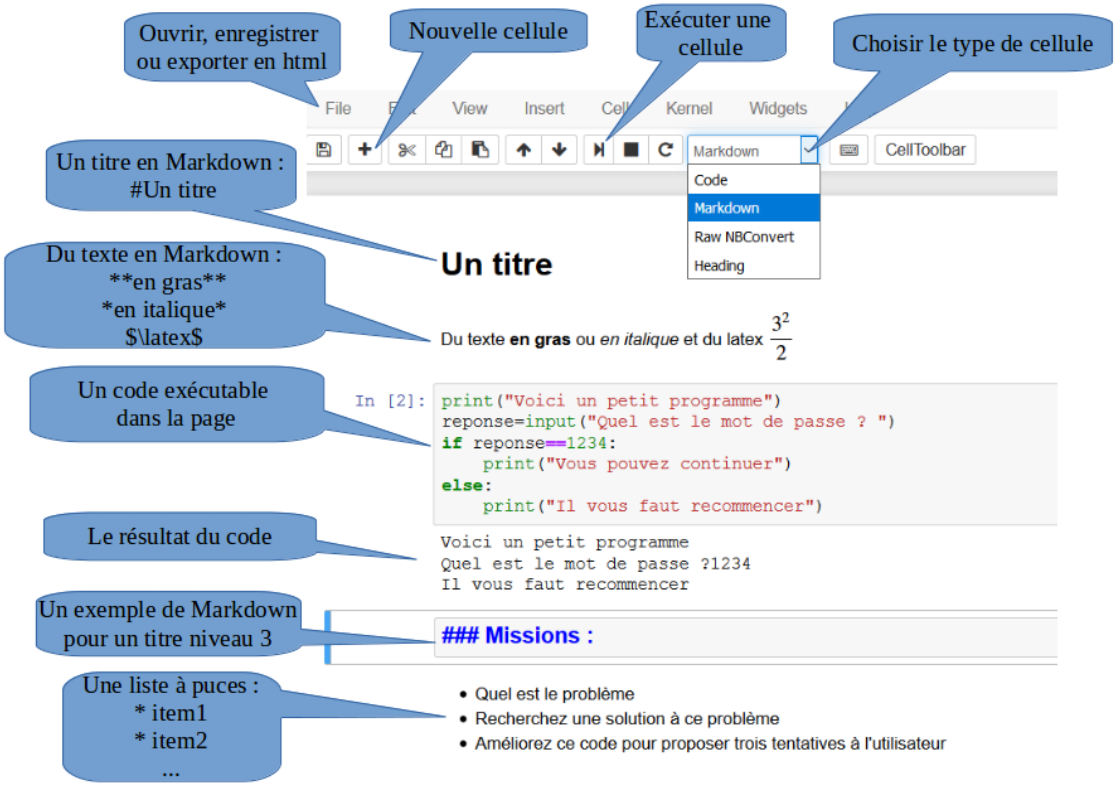
Vous l'aurez compris, si vous ouvrez ce pdf avec votre navigateur ou un autre visionneur, vous ne pourrez peut-être pas copier/coller l'indentation.

5.3 En html avec Jupyter

Jupyter Notebook permet de créer des fichiers contenant du code (Python ou autres...) avec du texte et une mise page basée sur du Markdown (Titres, texte, images, liens...). En plus, les codes insérés dans le **NoteBook** peuvent être exécutés directement dans la page (par les élèves par exemple).

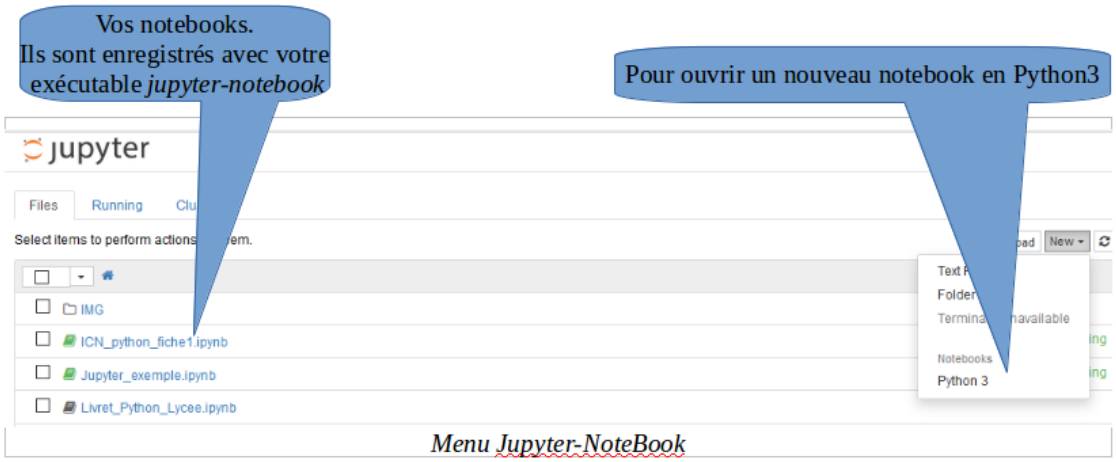
Un autre avantage est de pouvoir télécharger le document au format **html**. Les codes alors insérés seront colorés et le copier/coller ne pose pas de problème. Pour lire un fichier html, il suffit d'un navigateur internet (Mozilla, Chrome...).

Un exemple en image :



Le fichier html correspondant : jupyter-exemple.html

Le menu de Jupyter Notebook est plutôt simple :



Menu Jupyter-NoteBook

5.4 Installation de Jupyter

Vous pouvez utiliser directement Jupyter en ligne. Pour cela, il n'y a rien a installer :

- Aller sur le site : Jupyter Notebook
- Cliquer sur « Try it in your browser »
- Créer un nouveau Notebook Python3.

Remarques :

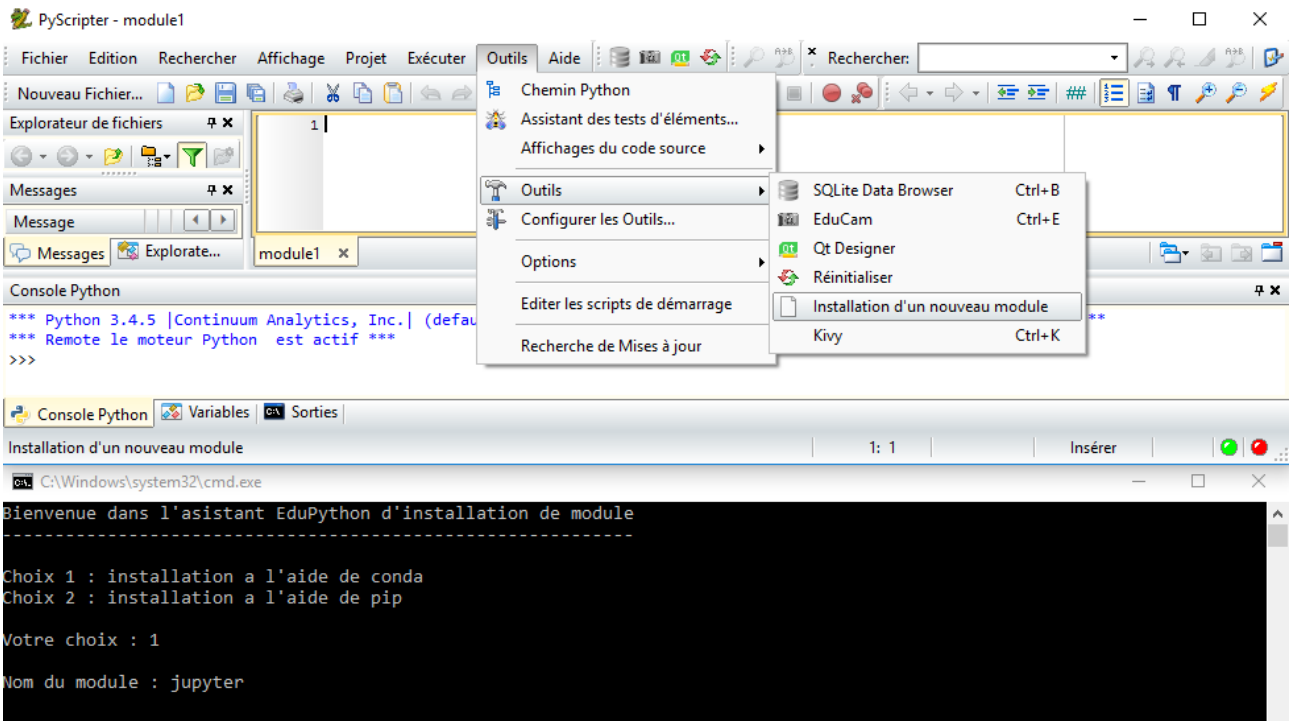
Il est possible qu'à certains moments le serveur soit saturé. Aussi, vous n'aurez pas accès aux interfaces graphiques comme `tkinter`...

C'est pourquoi il semble intéressant de pouvoir l'installer sur l'ordinateur (Pyzo ou EduPython).

Pour installer et utiliser Jupyter-Notebook avec Pyzo sur Miniconda3 :

- Ouvrir Pyzo
- Exécuter dans le shell la ligne : `conda install jupyter`
- Jupyter est installé dans Miniconda3/Scripts. Ouvrir jupyter-notebook (il faudra peut-être l'ouvrir en tant qu'administrateur : clic-droit sur le fichier puis « Exécuter en tant qu'administrateur »)
- Une console puis un onglet dans votre navigateur vont s'ouvrir.

Pour installer et utiliser Jupyter-Notebook avec EduPython :



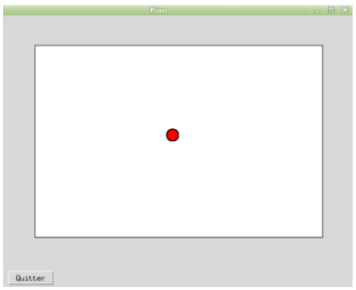
- Ouvrir EduPython
- Aller dans Outils Outils Installation d'un nouveau module
- Un console s'ouvre, taper 1 pour choisir une installation conda
- Taper jupyter pour installer ce module
- Jupyter est installé dans votre dossier *EduPython/App/Scripts*. Ouvrir *jupyter-notebook*.
- Une console puis un onglet dans votre navigateur vont s'ouvrir.

6 Exemples d'activités

6.1 Bouge la balle !

Une activité de découverte au format **html**. Les interfaces graphiques ne sont pas au programme mais cette activité permet de découvrir de façon légère la programmation Python.

En exécutant le code, les élèves découvrent une balle rouge :



Les élèves doivent corriger et améliorer comme ils le souhaitent ce programme :

Lien vers l'activité

6.2 Le nombre mystère

Une activité de découverte au format **html** :

Lien vers l'activité

6.3 Calcul mental

Lien vers l'activité

6.4 Ensorceler un nombre

Une activité connue autour des fonctions et des boucles

Lien vers l'activité

6.5 Un peu de probabilités et d'échantillonnage

Une activité connue autour des boucles et des instructions conditionnelles :

Lien vers l'activité

Une activité connue autour des boucles, des instructions conditionnelles et des fonctions :

Lien vers l'activité