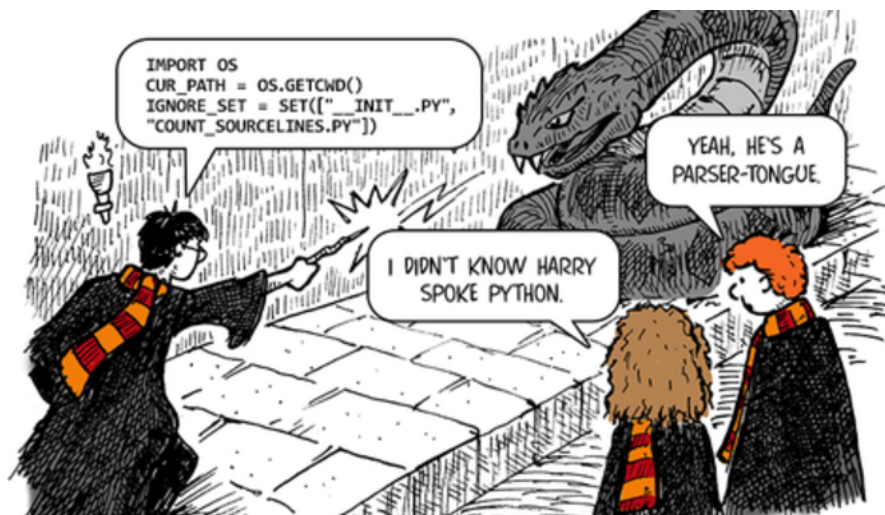


Chapitre 0

Introduction au langage Python

Simon Dauguet
simon.dauguet@gmail.com

4 septembre 2025



Nous allons présenter ici le langage informatique  `python`. On va déjà donner un aperçu de ce qu'est un langage informatique en général, et on va présenter les spécificités du langage  `python`. On va également prendre connaissance de l'environnement de travail (le logiciel que l'on va utiliser pour coder en  `python`).

Les ordinateurs sont inutiles. Ils ne peuvent
donner que des réponses.
Pablo Picasso

Table des matières

1	Structure simplifiée d'un ordinateur	3
1.1	Généralités	3
1.2	Arborescence des fichiers	5
1.3	Niveaux de langages	6
2	Environnement informatique au LPL	7

1 Structure simplifiée d'un ordinateur

1.1 Généralités

Un ordinateur est l'assemblage d'objets qui permet le traitement d'informations par l'exécution d'une séquence d'instructions. Cette séquence d'instruction se nomme un programme. De nos jours, un ordinateur est composé d'objets électroniques (et non mécanique), il est numérique (et non analogique), et est programmable.

Exemple 1.1 :

éléments possibles d'un ordinateur

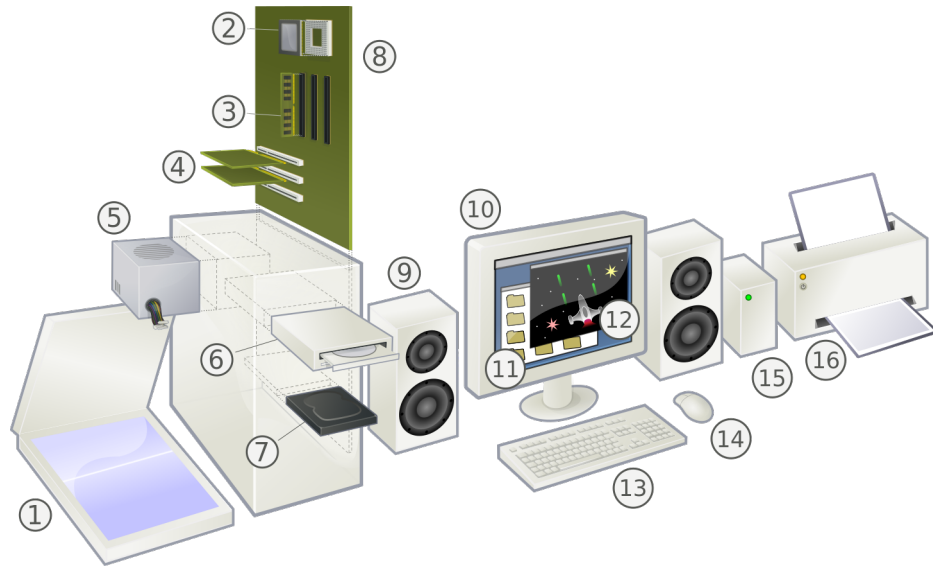


FIGURE 1 – source : wikipédia.org

1 : scanner	2 : processeur	3 : mémoire vive (RAM)
4 : carte de périphériques	5 : alimentation	6 : lecteur de disques
7 : disque dur	8 : carte mère	9 : enceintes
10 : écran	11 : logiciel système (OS)	12 : logiciel d'application
13 : clavier	14 : souris	15 : disque dur externe
16 : imprimante		

Le poste de travail se compose généralement d'une chaise, une table, un écran, un clavier, d'une souris et d'enceintes. Afin de ne pas se faire mal (dos, articulations, yeux, etc.) il est important d'adopter une bonne position à son poste de travail.

Ci-dessous, un schéma explicatif d'une position convenable.

Enfin, afin de limiter la fatigue des membres supérieurs, il est conseillé d'utiliser une souris verticale et savoir taper au clavier.

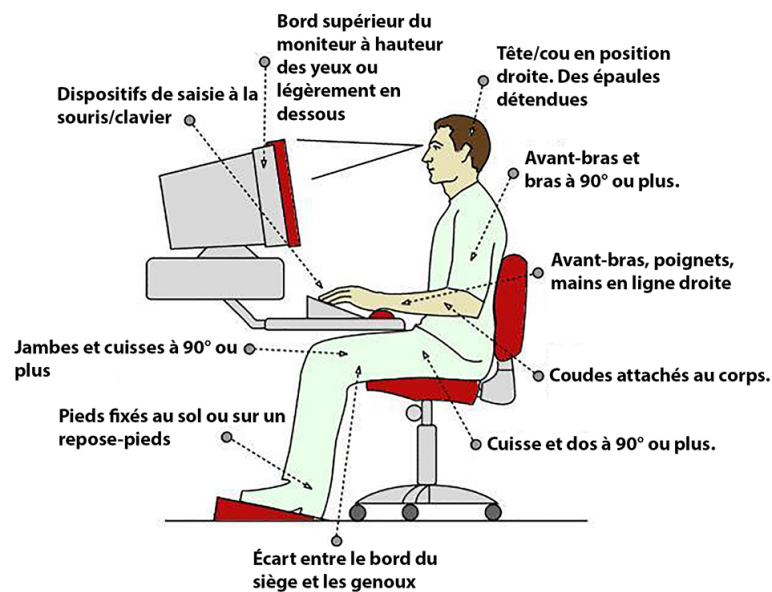


FIGURE 2 – source : linformatique.org

Pour le clavier, il faut essayer de limiter les mouvements des mains sur le clavier et utiliser ses dix doigts comme le suggère l'image suivante pour un clavier **azerty**. Selon votre langue, différentes dispositions de lettres existent (qwerty pour l'anglais, azerty pour le français, etc.), il est également possible de choisir une disposition optimisée telle que le clavier **bépo** pour le français.

Vous pouvez vous entraîner chez vous à une tape efficace sur le site [<https://agilefingers.com/fr/>].

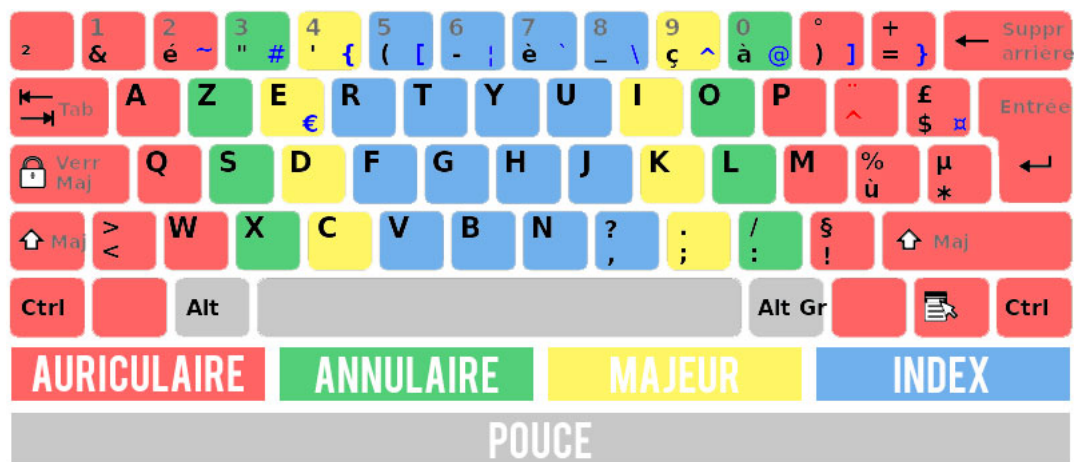


FIGURE 3 – source : latutotheque.fr

On peut faire des combinaisons de touches afin d'obtenir d'autres caractères. Par exemple, Shift

+ a donne la majuscule A ; Shift + & donne 1 ; Shift + , donne ? ; Alt Gr + " donne le dièse # ; Alt Gr + à donne l'arobase @ ;

Exercice 1 :

Préciser la (ou les) touche pour obtenir € ; è ; T ; > ; / ; { ; } ; 7 ; 0 puis +.

1.2 Arborescence des fichiers

Les fichiers contenus dans un ordinateur sont ordonnés dans une arborescence formée par les répertoires du système de fichiers.

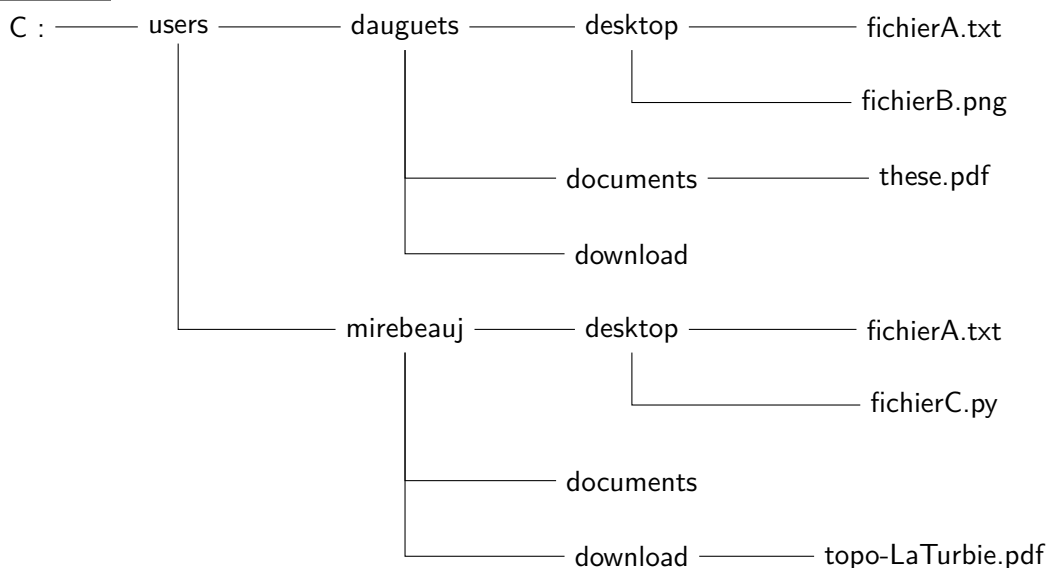
Cette arborescence débute à la racine qui est usuellement notée sous linux /, sous windows C:.

On remarque également que le nom des fichiers se décompose en deux : le nom à proprement parlé suivi d'une extension introduite par un point (.txt, .py, .jpg, .pdf, .odt, .docx, ...). Ces extensions permettent à l'OS d'appeler le programme adéquat à l'ouverture de chaque fichier.

Exercice 2 :

Proposer plusieurs extensions possibles pour chacun des types de fichiers.

texte	son	image	vidéo	langage

Exemple 1.2 :

Définition 1.1 (Adresse relative, Adresse relative d'un fichier) :

Il y a deux manières de définir l'adresse d'un fichier :

- *L'adresse absolue* d'un fichier est la suite des répertoires menant de la racine au nom du fichier (compris).
- *L'adresse relative* d'un fichier A relativement à un autre dossier B, est la suite des répertoires menant du dossier B au nom du fichier A.

L'adresse absolue d'un fichier n'est que l'adresse du fichier relativement à la racine.

Exemple 1.3 :

Le fichier `fichierB.png` est une photo (extension `.png`) dont l'adresse absolue est :

`C:\users\dauguets\desktop\fichierB.png`.

Le fichier `these.pdf` est un pdf (extension `.pdf`) dont l'adresse absolue est :

`C:\users\dauguets\documents\these.pdf`.

On peut remarquer qu'il existe sur le disque `C:` deux fichiers avec le même nom `fichierA.txt`. Leur adresse absolue étant distincte, cela ne pose aucun problème.

Pour remonter au répertoire parent dans l'arborescence, on utilise `...`

Exemple 1.4 :

Ainsi l'adresse du fichier `these.pdf` relativement au dossier `mirebeauj\desktop` est `..\..\dauguets\documents\these.pdf`

Exercice 3 :

1. Préciser le type de fichier, l'adresse absolue de `fichierB.png` ainsi que son adresse relativement à `dauguets\download`
2. idem pour `fichierC.py`
3. idem pour `topo-LaTurbie.pdf`

1.3 Instructions et niveaux de langages informatiques

L'ordinateur est composé de milliard de transistors (sorte d'interrupteurs électroniques), qui bien agencés forment des portes logiques, qui elles-mêmes agencées forment les briques de calculs (*i.e.* additionneur), etc. puis l'ordinateur est lui même composé de différents éléments qui ont chacun une fonction (*i.e.* UAL, mémoire, carte graphique, ...).

La suite d'assemblage de transistors en modules de plus en plus complexe permet donc de former un ordinateur.

Cet assemblage de briques élémentaires utilise différents langages de communication.

	niveau du langage	exemple de langage	exemple de code
<i>programmation</i> ↓ <i>compilation</i> ↓ <i>assemblage</i> ↓	très haut niveau	langage humain	additionne 2 à 5, puis multiplie par 3
	haut niveau	 python	<code>a = 5</code> <code>a = a + 2</code> <code>a = a * 3</code>
	bas niveau	langage assembleur	<code>mov eax 5</code> <code>add eax 2</code> <code>mul eax 3</code>
	très bas niveau	binaire	<code>10001110 01101000</code> <code>10101011 11001001</code> ...

La programmation vue en CPGE consistera donc à passer du langage humain à un langage de haut niveau ( python).

Remarque :

De nos jours, les logiciels sont souvent écrits en langage haut niveau. Une hiérarchie existe également entre logiciels : l'OS (*Operating System*, par exemple linux, windows) sert à coordonner tous les logiciels communs (firefox, vlc, libreoffice, chrome, etc.) et faire l'interaction avec le matériel (*i.e.* souris, clavier, écran, enceintes, etc.).

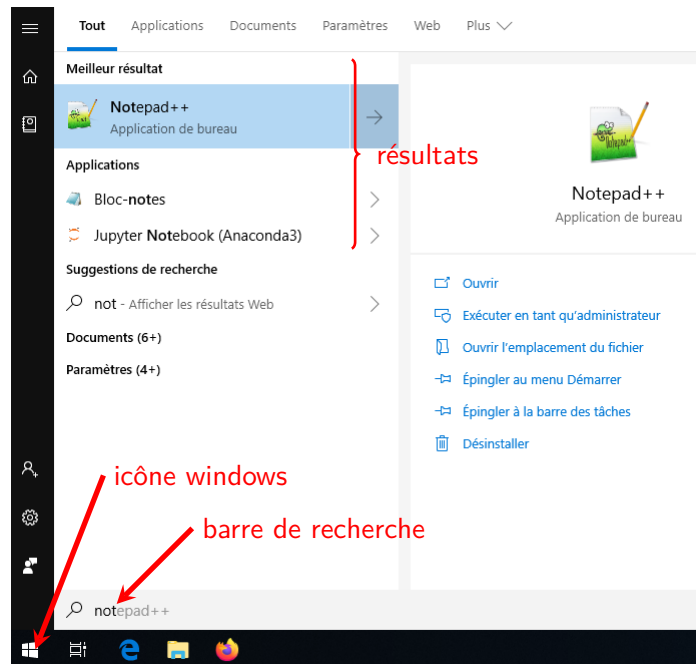
2 Environnement informatique au LPL

Au lycée, il est possible de lancer un programme python en ligne de commandes via l'invite de commandes de linux ou de windows, ou de le tester via une IDE (Integrated Development Environment) tels que **pyzo** ou **spyder**.

Remarque : sous l'OS windows, vous pouvez lancer un programme reconnu par votre OS en cliquant sur l'icône en bas à gauche (ou la touche windows de votre clavier ) puis en tapant son nom au clavier (une barre de recherche apparaît). Valider le résultat voulu.

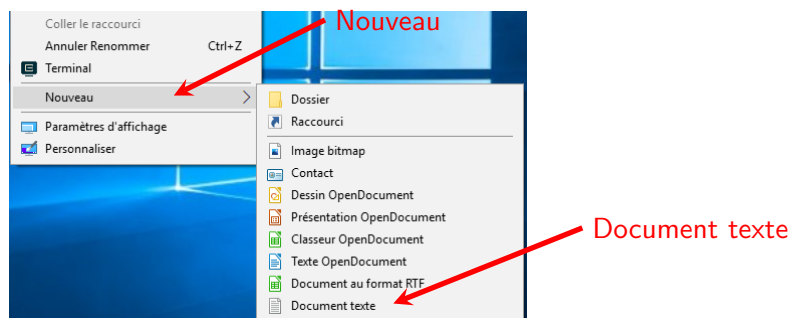
Exemple 2.1 :

Pour lancer l'éditeur de texte **notepad++** (utile pour lire des fichiers **.py** sans les exécuter)



Une fois sous windows connecté à son compte,

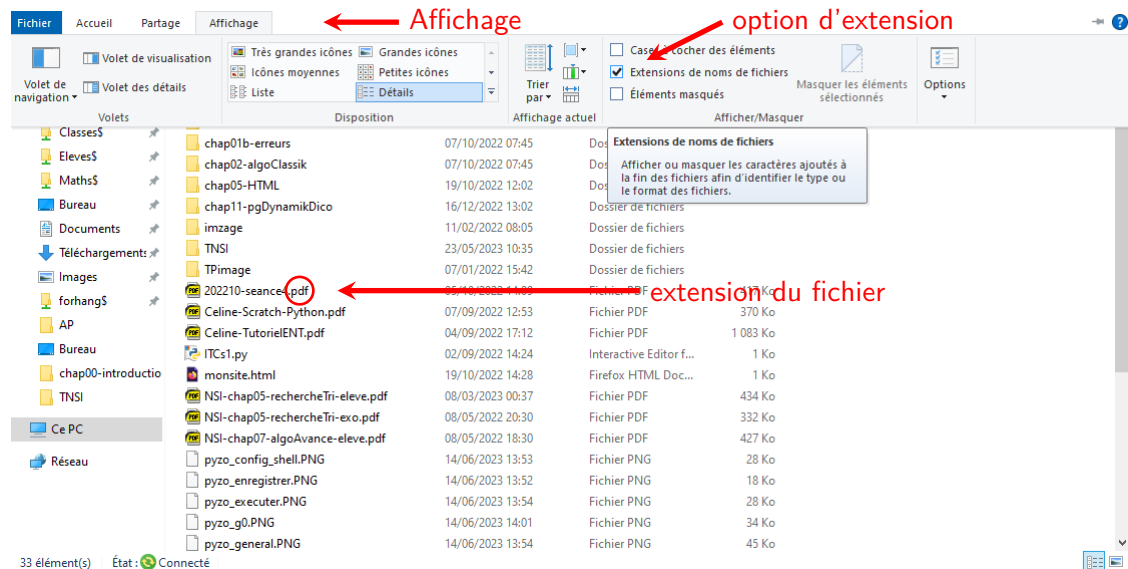
étape n° 1 : créer un fichier `introduction.py` avec un clic gauche de votre souris, **Nouveau > Document texte**



Actuellement, votre répertoire personnel est accessible de deux manières différentes,

- l'une à partir de votre **espace personnel** (*a priori* le disque avec la lettre P:),
- l'autre à partir du réseau pédagogique **SERV-2019-PEDA/nomp**.

étape n° 2 : vérifier que l'extension des fichiers soit visible (**Affichage > Extensions de noms de fichiers**).
Vérifier que l'extension soit `py` et non `txt`, et modifier-la le cas contraire.



étape n° 3 : ouvrir le fichier à l'aide de **notepad++**, **pyzo** ou **spyder**. Écrire dans ce fichier les lignes suivantes en les adaptant, puis enregistrer.

```
1 """
2 TP n°0 : introduction à l'environnement python
3 AAAA/MM/JJ : NOM Prénom
4 """
5 ## module ##
6 import os                # import de la bibliothèque os
7 os.chdir(r"P:\Desktop\TP0") # défini ce répertoire comme répertoire de travail
8
9 ## fonctions ##
10 """ pour plus tard """
11
12 ## code réponse ##
13 mfile = open("monFichier.txt", 'w') # ouvre (ou crée) le fichier "monFichier.txt"
14 print("Bonjour toi", file = mfile)  # écrit "Bonjour toi" dans le fichier
15 mfile.close()                     # ferme le fichier "monFichier.txt"
16 print("Au revoir")                # affiche "Au revoir"
```

Afin de ne pas être gêné par les spécificités de certains IDE, on change le répertoire de travail en utilisant la fonction `chdir` (change directory) de la bibliothèque `os`.

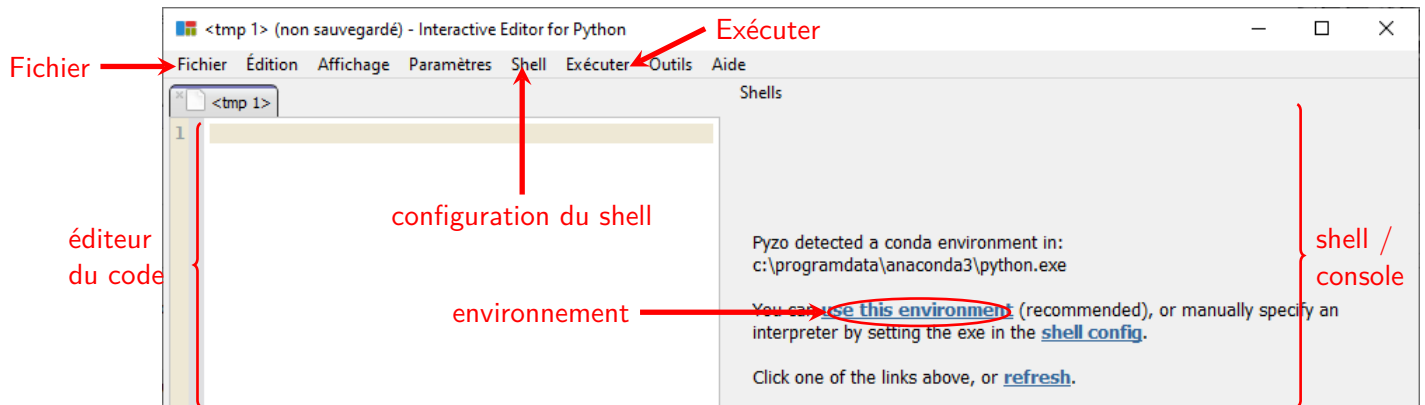
Remarque : si vous programmez sous linux, vous pouvez définir le **shebang** avec la première ligne `#!/usr/bin/env python` dans votre fichier. Le **shebang** précise à votre OS linux le logiciel à appeler lorsque vous exécutez votre fichier.

Exécuter votre programme avec pyzo

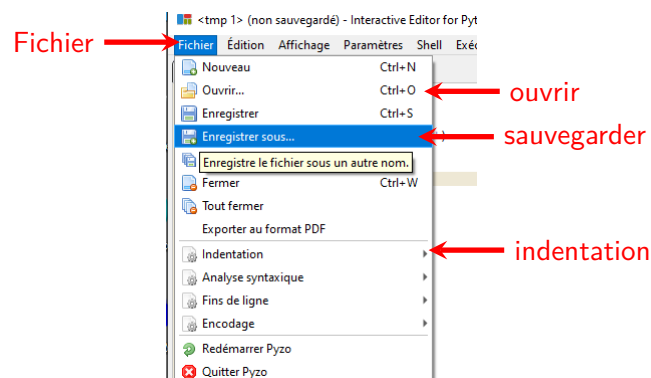
étape n° 4 : lancer pyzo et sélectionner l'environnement **python** qui sera utilisé pour interpréter votre code

Pour se faire, deux choix s'offrent à vous,

- sélectionner l'option par défaut
- aller dans **Shell > Configuration des shells**, dans le champ **exe** choisir **python** et valider



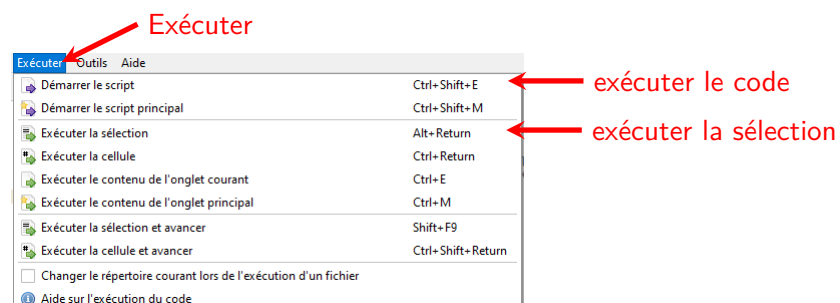
étape n° 5 : ouvrir le fichier `introduction.py` via le menu **Fichier > Ouvrir**



Remarque : parfois vous pouvez vouloir modifier les règles d'indentation. Il vous est conseillé de choisir 3 espaces pour une indentation.

étape n° 6 : exécuter votre code

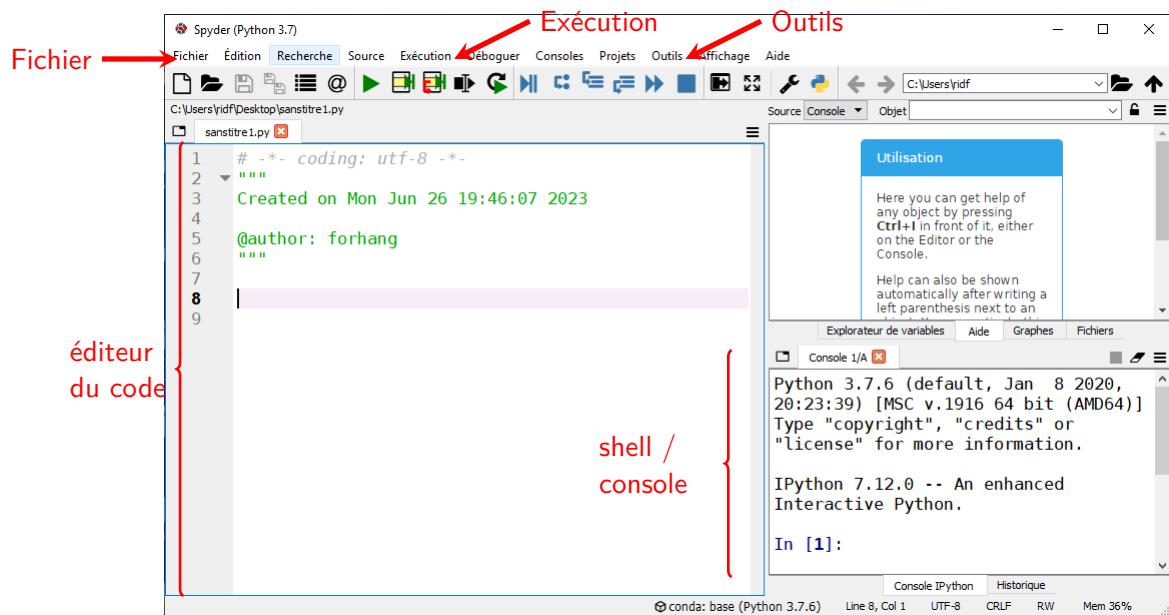
- soit entièrement avec `F5` ou en combinant les touches `Ctrl` + `Shift` + `E`
- soit en sélectionnant la partie de votre code en combinant les touches `Alt` + `Return`
- soit en passant par le menu **Exécuter**



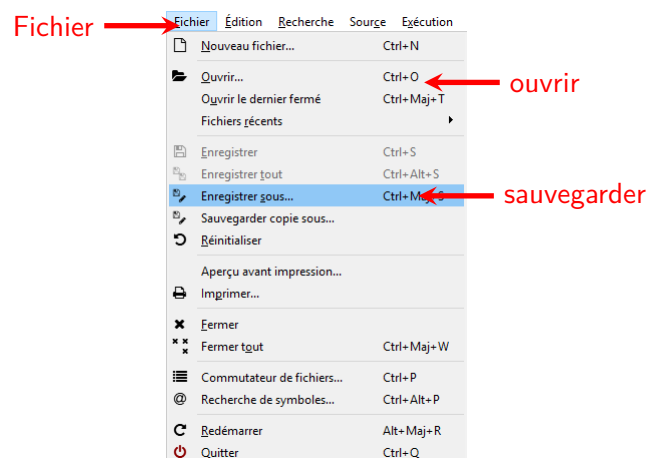
Un fichier nommé `monFichier.txt` vient d'être créé dans le répertoire `"P:\Desktop\TP0"` et contient la ligne `"Bonjour toi"`

Exécuter votre programme avec spyder

étape n° 4 : lancer spyder



étape n° 5 : ouvrir le fichier `introduction.py` via le menu **Fichier > Ouvrir**

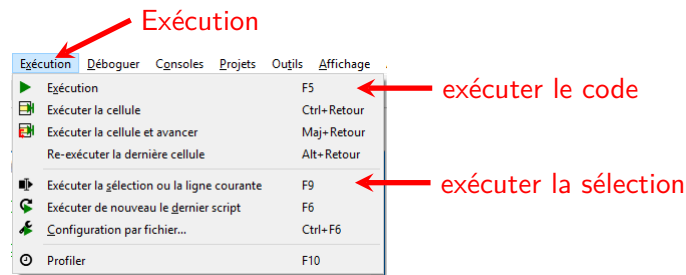


Remarque : parfois vous pouvez vouloir modifier les règles d'indentation. Pour se faire, aller dans **Tools > Preferences > Editor > Source code > Indentation characters**. Il vous est conseillé de choisir 3 espaces pour une indentation.

étape n° 6 : exécuter votre code

- soit entièrement avec **F5**
- soit en sélectionnant la partie de votre code et appuyer sur la touche **F9**

— soit en passant par le menu **Exécution**



Un fichier nommé `monFichier.txt` vient d'être créé dans le répertoire `P:\Desktop\TP0` et contient la ligne "Bonjour toi"

Conseils à suivre :

1. ne faites qu'un seul fichier python par TP, ainsi vous garderez visibles les algorithmes les uns à la suite des autres. Vous pourrez ainsi réutiliser la structure des algorithmes qui fonctionnent.
2. **tester** votre code **au fur et à mesure**
3. utilisez des noms de variables explicites et sauf cas exceptionnel, utiliser `i`, `j` ou `k` comme indices de boucles
4. utilisez les commentaires simple (`#`) et multilignes (`"""`) pour vous expliquer votre code (14 jours plus tard vous ne saurez plus pourquoi vous avez écrit ce code)

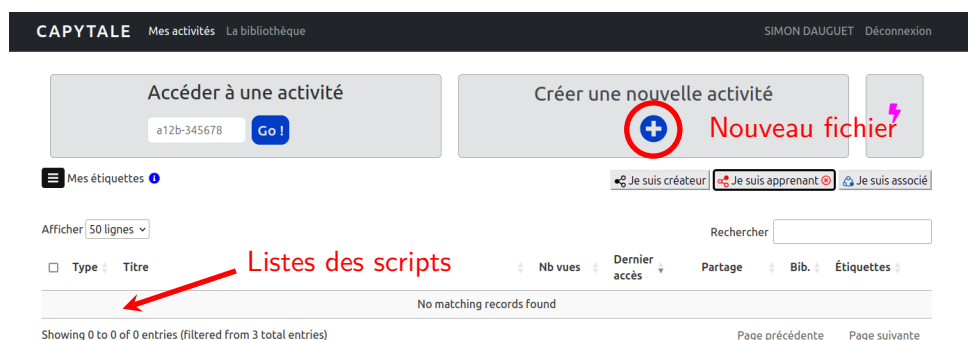
Avec des codes ENT

À partir du moment des codes ENT, une application sur l'ENT permet de faire des scripts python sur une interface simplifiée. Vous pourrez alors continuer à travailler vos scripts chez vous sans avoir besoin de vous envoyer ce qui aura été fait au lycée.

L'application en question est l'application CAPYTALE.

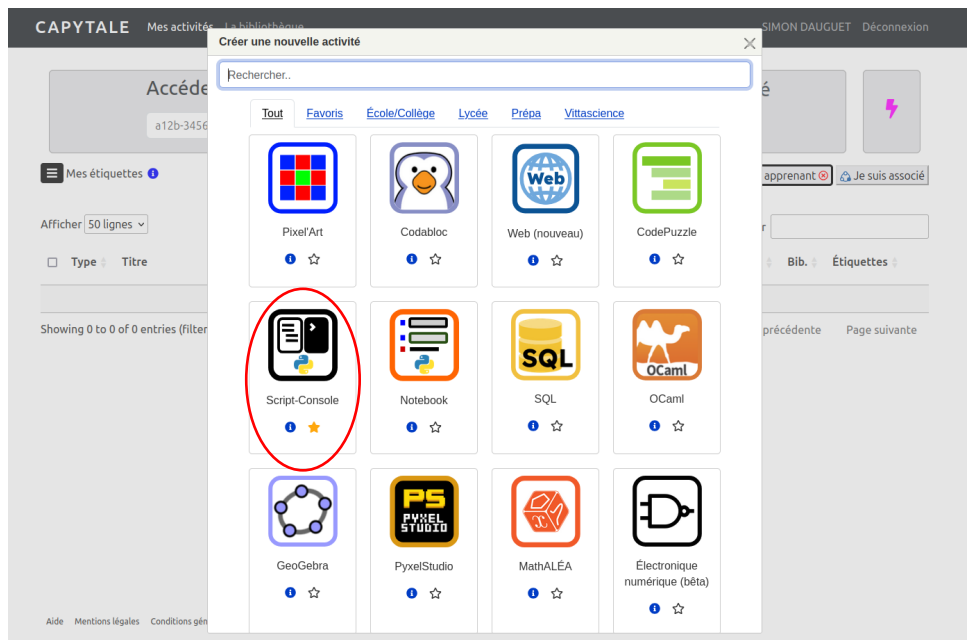
Pour accéder à CAPYTALE : taper **CAPYTALE** dans google. Le premier lien (<https://capytale2.ac-paris.fr/web/>) est le bon. Vous serez redirigé vers une page vous demandant de choisir votre ENT (en l'occurrence, Île-de-France), puis de vous connecter avec vos codes ENT.

Vous serez alors sur la page d'accueil de CAPYTALE :



2 ENVIRONNEMENT INFORMATIQUE AU LPL

Vous pouvez alors choisir l'activité **Script-Console**.



Et vous aurez accès à votre script :

