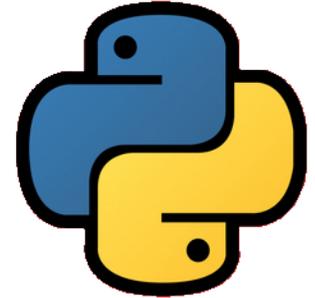


Avec quoi programmer ?

Installation des principaux logiciels et applications pour les parties « microcontrôleurs » et « programmation » des nouveaux programmes du lycée.

Python est un langage de programmation, où l'on rédige des lignes de « texte », appelé **script** ou **code**. Il faut donc installer un logiciel pour comprendre et exécuter ce code.

Ces logiciels sont appelées **I.D.E.** pour « Environnement de développement intégré ». Il en existe de nombreux exemples. Ce document présentera les principaux, utilisables au lycée.



Python étant également étudié en maths, il sera judicieux de se rapprocher de l'équipe de mathématique pour choisir et adopter une solution commune. Il faudra sans doute se rapprocher des administrateurs(trices) réseau pour une installations dans les établissements.

Ce document présentera également des logiciels pour commander les cartes microcontrôleurs **Arduino** et **Micro:bit**.

*Dans le cas d'une utilisation de la carte Arduino avec un IDE Python, il faudra installer le module **pyfirmata**.*

Edupython -----	2
Thonny-----	3
Pyzo-----	4
Spyder-----	5
PyCharm-----	6
En ligne ou pour tablette -----	7
MU editor-----	9
Arduinor-----	10
Processing-----	10
Fritzing-----	11
Tinkercad-----	11

EduPython

Projet libre développé par une équipe de professeurs de mathématiques de l'académie d'Amiens.



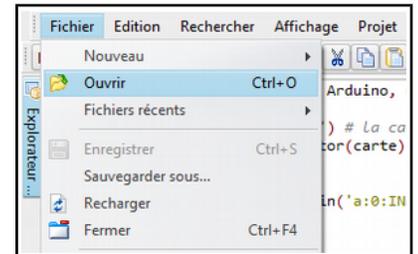
Installation d'EduPython :

Se rendre sur le site <https://edupython.tuxfamily.org/> télécharger le fichier SETUP_EP26.exe et l'exécuter.

EduPython peut s'installer à différents endroits :

- Sur une **clé USB** : dans ce cas choisissez le lecteur représentant votre clef, l'installation se fera alors dans un dossier nommé «EduPython ».
- Sur un **ordinateur personnel** : même principe que pour la clé USB, choisissez un emplacement qui vous convient.
- Installation sur **réseau** : contacter votre administrateur.

*Remarque importante: pour la version portable installée sur clé, votre ordinateur ne saura pas ouvrir de lui-même les programmes Python (.py). Pour exécuter un programme vous devrez donc ouvrir EduPython et le rechercher par **fichier > ouvrir***



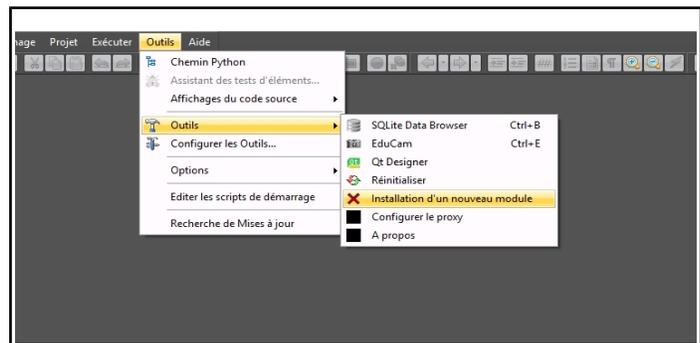
Installation d'un nouveau module :

Les principales bibliothèques sont déjà installées dans **EduPython** (numpy, math, time, matplotlib...).

Mais il est possible d'installer d'autres modules : Prenons l'exemple ici du module « pyfirmata » :

Démarrer **EduPython**.

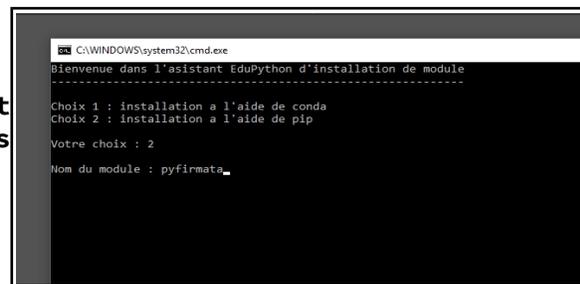
Faire **Outils > outils > installation d'un nouveau module**.



Un menu apparaît

Choisir **2** et ensuite taper **pyfirmata** (c'est le module nécessaire pour programmer les cartes arduino en python).

Suivre les instructions.



Thonny

Projet libre de l'université de Tartu en Estonie.

Pour l'obtenir, se rendre sur le site :

<https://thonny.org/>,
télécharger le fichier d'installation et l'exécuter.

Pour exécuter le script ;
cliquer sur l'icône



Matplotlib et **numpy** ne sont pas installés dans **thonny**.

Installation d'un nouveau module :

Cliquer sur **Tools**, puis **Manage packages**.

Une fenêtre s'ouvre.

Taper le nom du module à installer, faire **Search**, puis **Install**.

```
Thonny - D:/Energie_cinétique.py @ 7:78
File Edit View Run Tools Help

Energie_cinétique.py x
v = 12
m = 3.5
p = m*v
print("La quantité de mouvement à une valeur de ", p, "kg.m/s")
Ec = 0.5*m*v**2
print("L'énergie cinétique vaut", Ec, "J")
# il faut remplacer Ec par round(Ec) si vous voulez afficher un nombre entier

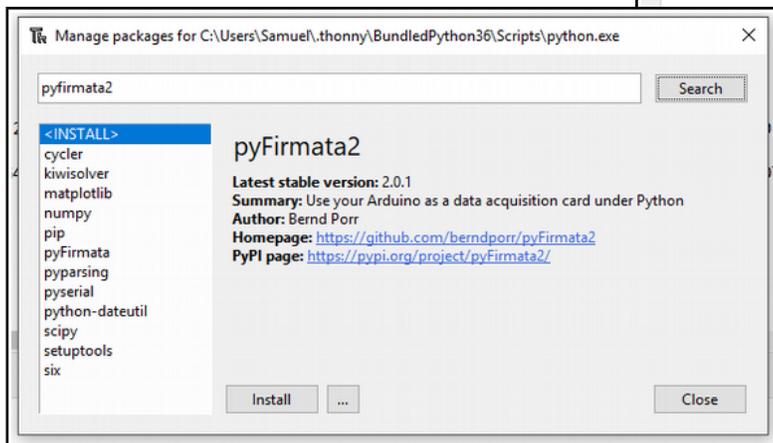
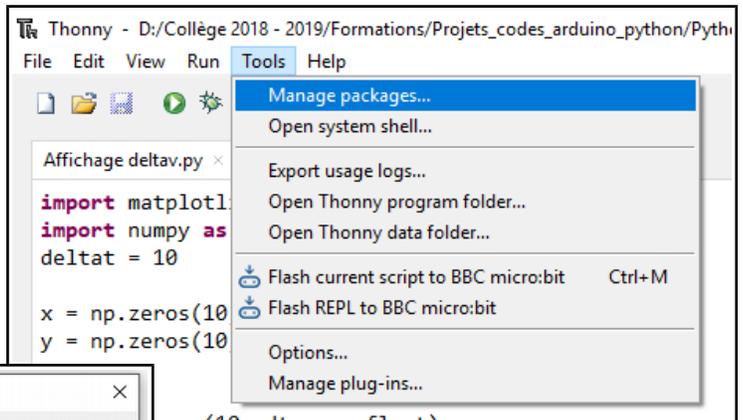
Shell
Python 3.6.6
>>> %cd 'D:\'
>>> %Run 'Energie_cinétique.py'

La quantité de mouvement à une valeur de 42.0 kg.m/s
L'énergie cinétique vaut 252.0 J

>>> %Run 'Energie_cinétique.py'

La quantité de mouvement à une valeur de 42.0 kg.m/s
L'énergie cinétique vaut 252.0 J

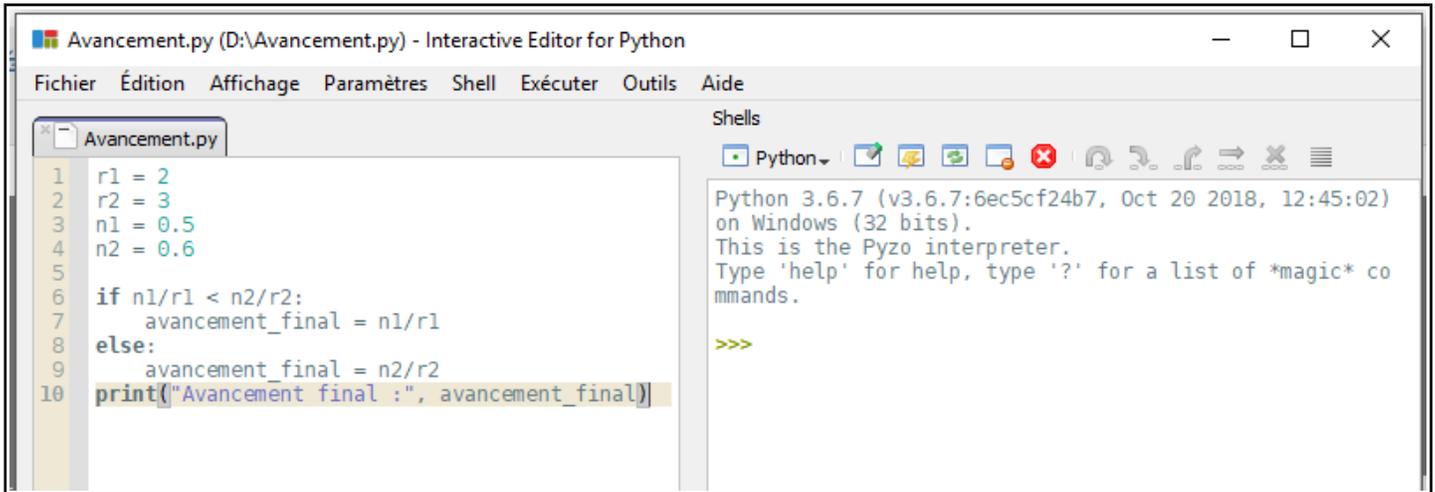
>>>
```



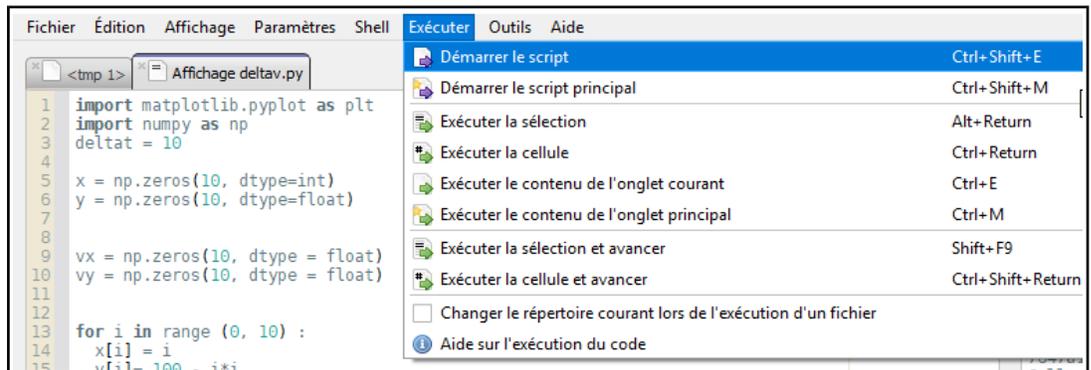
Pyzo

Pyzo est un projet libre, simple d'utilisation : <https://pyzo.org/start.html>

Le plus simple pour installer **Pyzo** et de télécharger et installer la distribution **Winpython**, qui contient tous les outils pour programmer en sciences (**matplotlib**, **numpy**) <http://winpython.github.io/>

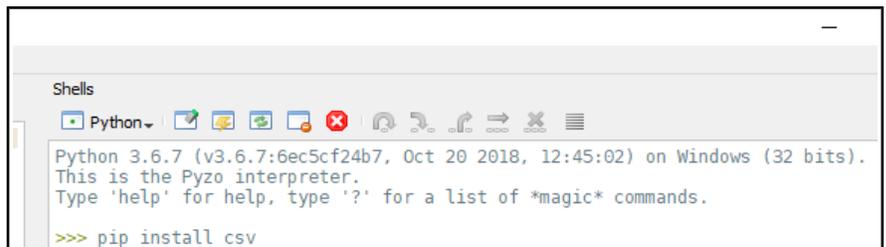


Pour lancer un script : faire **Exécuter**, puis **Démarrer le script**.



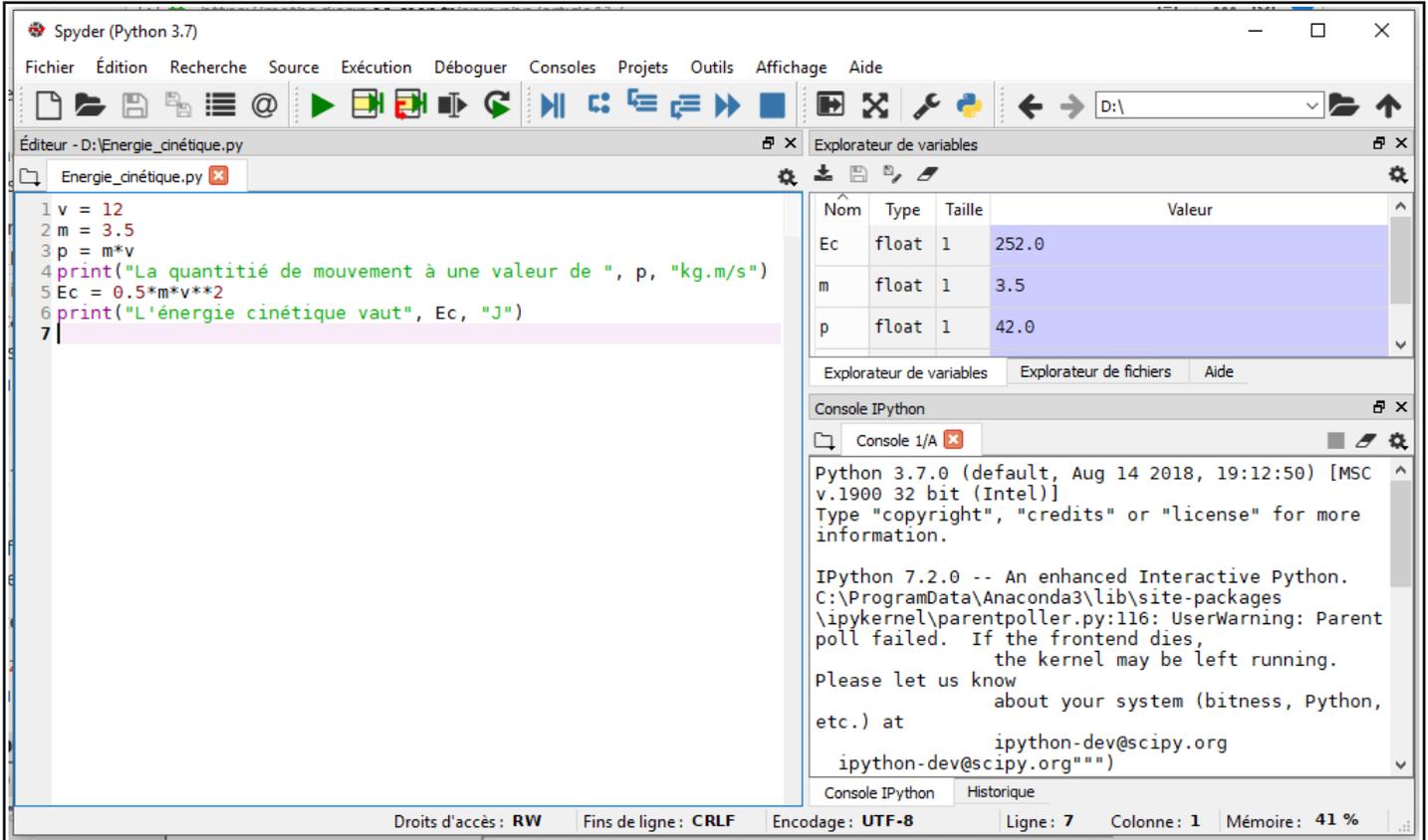
Installation d'un nouveau module :

Dans la console Python (symbole >>>), saisir **pip install nom_module** puis valider.



Spyder

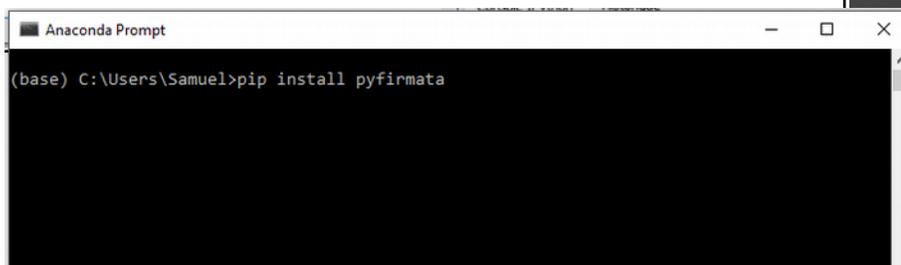
Spyder est un IDE très complet et très puissant. <https://www.spyder-ide.org/>
Pour l'installer il faut installer la distribution **Anaconda**, se rendre sur le site <https://www.anaconda.com/distribution/>, télécharger le fichier et l'exécuter.
Matplotlib et **numpy** sont présents dans la distribution **Anaconda**.



Installation d'un nouveau module :

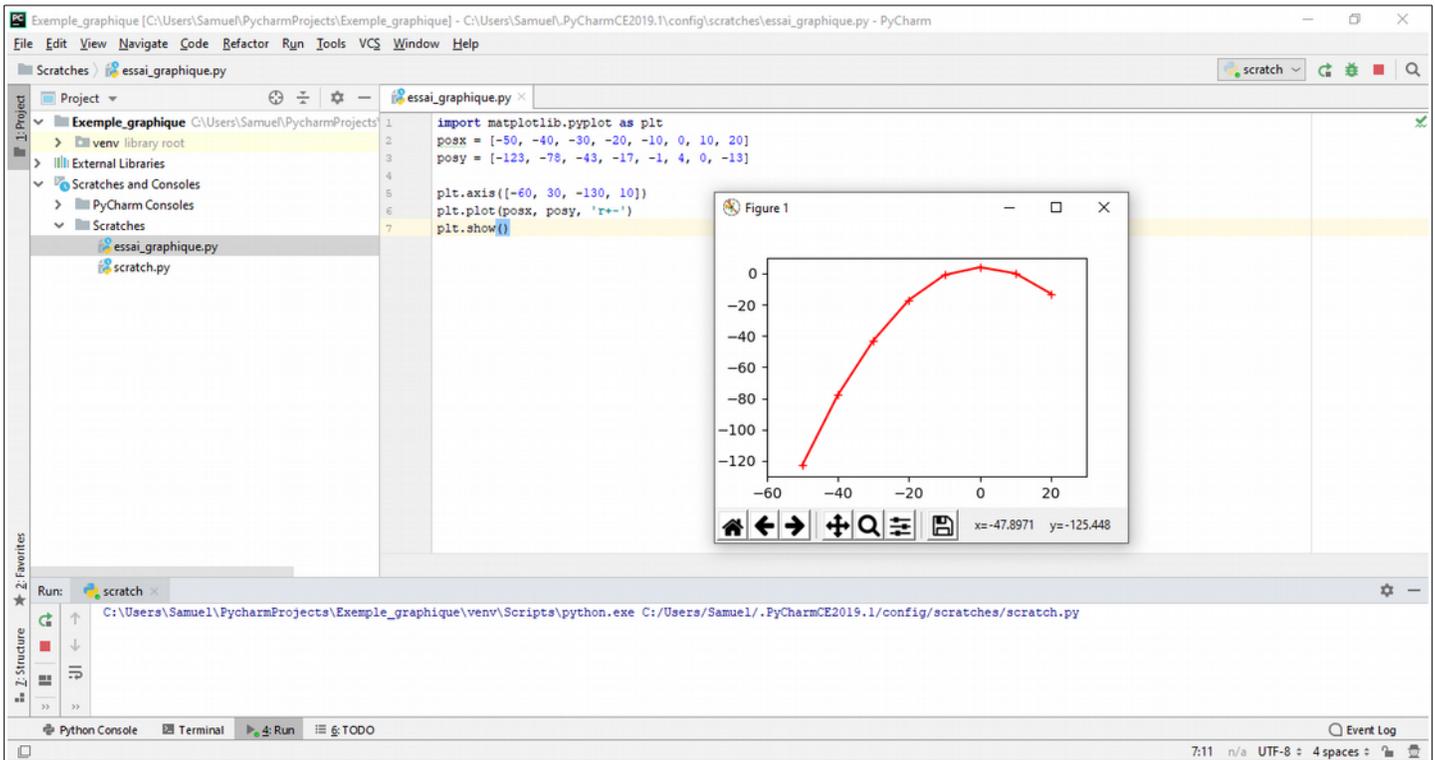
Dans le menu démarrer, lancer l'application **Anaconda Prompt**, qui se trouve dans le dossier **Anaconda**.

Une fenêtre apparaît, saisir **pip install nom_module** puis valider.



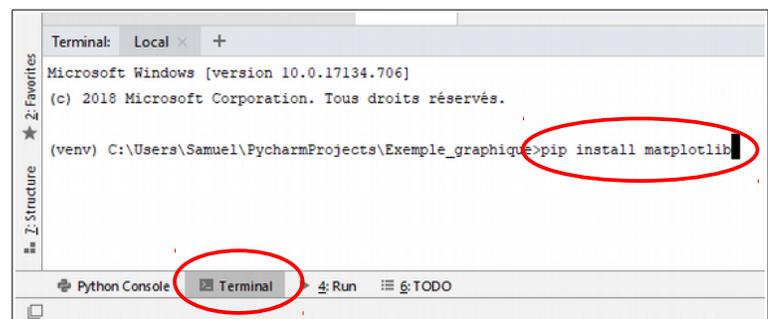
PyCharm

Un outil très complet, développé par JetBrains.
Se rendre sur le site :
<https://www.jetbrains.com/pycharm/> et télécharger
la version « community », qui est gratuite.



Installation d'un nouveau module :

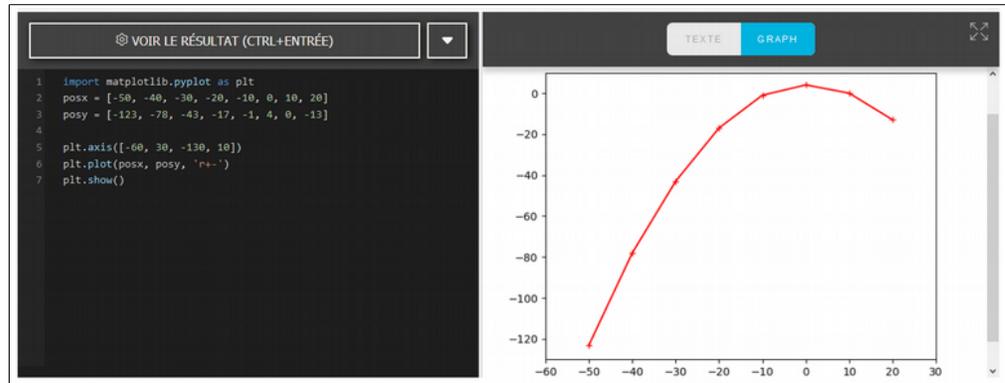
Matplotlib et **numpy** ne sont pas installés dans **Pycharm**, il faut suivre cette procédure :
Cliquer sur **Terminal** en bas à gauche de l'écran.
Taper la commande **pip install matplotlib** et valider.



En ligne ou sur tablette

La console Python du livre scolaire :

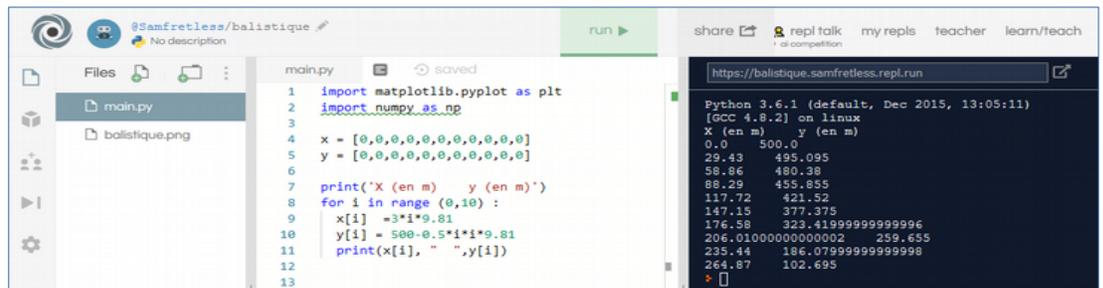
<https://www.livrescolaire.fr/console-python>



Le site Repl.it

<https://repl.it/>

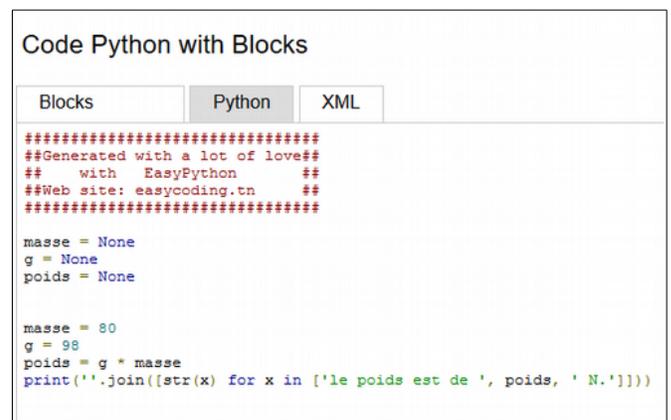
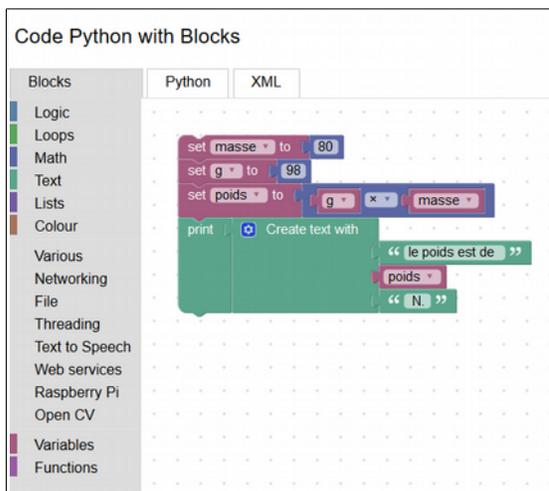
(nécessite une inscription, gratuite pour les enseignants)



EasyCoding

<http://easycoding.tn/ep/demos/code/>

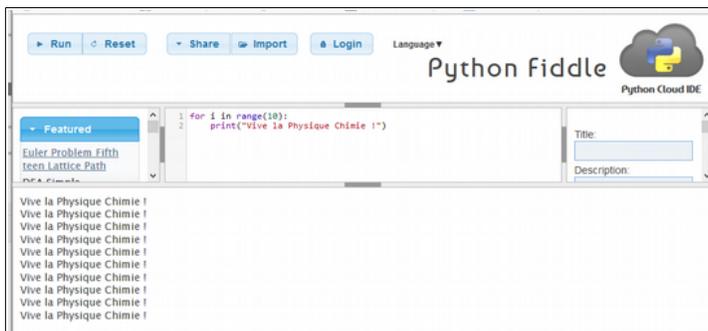
Permet de programmer en **Blocs** et l'application convertit en **Python**.



Python Fiddle :

<http://pythonfiddle.com/>

Site pour programmer directement en ligne (sans inscription).



Pyodide :

Le projet en développement de **Mozilla** pour faire du **Python** avec un navigateur. Pour plus d'infos : [Iodide: an experimental tool for scientific communication and exploration on the web.](#)

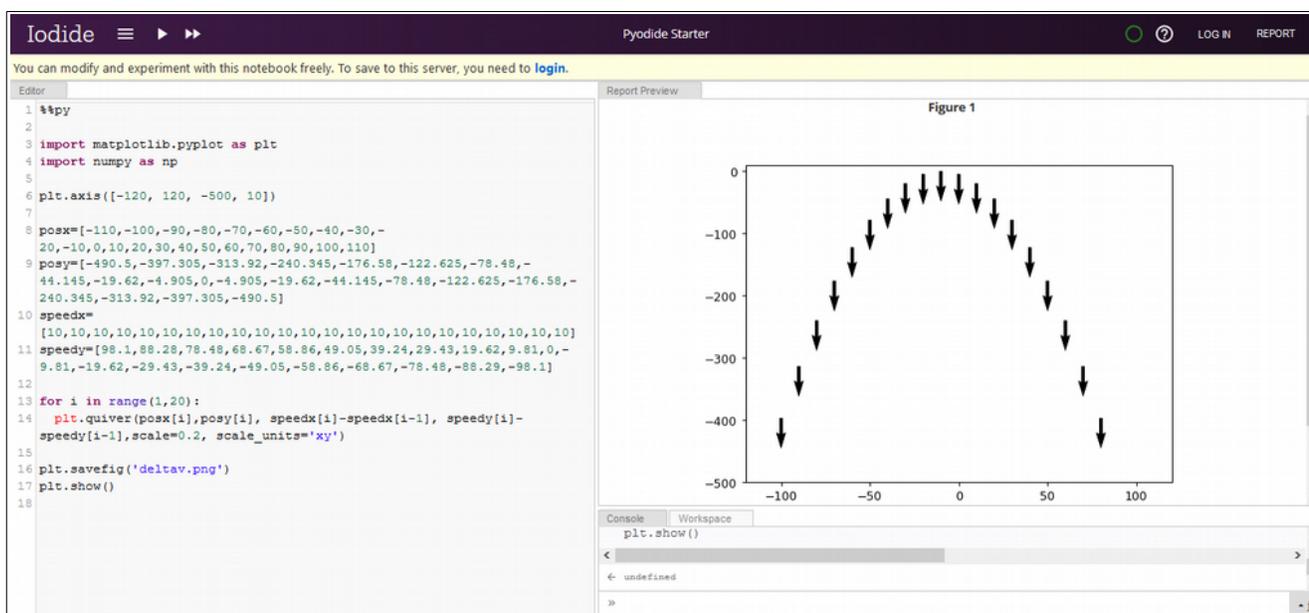
Pour tester : <https://alpha.iodide.io/notebooks/222/>



Supprimer tout ce qui est inscrit dans la colonne de gauche (Editor).

Inscrire `%%py` en première ligne et ensuite il est possible de taper un code python.

Matplotlib et **numpy** sont accessibles sans action préalable.



Pydroïd

(sur tablette et téléphone) :



Avec quoi programmer ?

Pour la carte micro:bit

MU editor est un logiciel permettant de programmer la carte microcontrôleur **Micro:bit** en langage **Python**.



MU editor permet également de déposer directement le microprogramme sur la carte, sans avoir à passer par l'étape manuelle de dépôt du fichier .HEX et il permet également de recevoir et d'envoyer des données en temps réel à la carte (on appelle cela la **console REPL**).

```
1 from microbit import *
2 import utime
3 import math
4
5 pin2.write_digital(False)
6 utime.sleep_ms(1)
7 pin2.write_digital(True)
8 utime.sleep_us(10)
9 pin2.write_digital(False)
10 depart = utime.ticks_us()
11 while not pin0.read_digital():
12     depart = utime.ticks_us()
13
14 while pin0.read_digital():
15     retour = utime.ticks_us()
16
17
18 temps = (retour-depart)/2
```

Pour l'installer, aller sur le site <https://codewith.mu/> et suivre les instructions.

(Pour les versions inférieure à windows 10, il faut également installer le Windows Serial driver :

<https://os.mbed.com/docs/mbed-os/v5.7/tutorials/windows-serial-driver.html>)

Les modules numpy et matplotlib ne sont pas accessibles avec MU Editor. En revanche, il est possible de tracer des graphiques de mesures en temps réel et générer des fichier .csv.

A noter : il existe une application en ligne permettant de commander la carte micro:bit en python.

<https://python.microbit.org/v/1.1>

On télécharge un fichier .hex généré et on le dépose sur la carte avec l'explorateur de Windows.

```
1 # Add your Python code here. E.g.
2 from microbit import *
3
4
5 while True:
6     display.scroll('Hello, World!')
7     display.show(Image.HEART)
8     sleep(2000)
9
10
11
12
13
14
15
16
17
18
19
20
```

Pour la carte Arduino

IDE Arduino

Arduino est le Logiciel permettant d'installer les drivers des cartes Arduino, de les programmer **en langage C**.



A noter : Arduino permet également de déposer sur les cartes Arduino le microprogramme « **Firmata** » permettant aux cartes de communiquer avec un logiciel IDE Python.

<https://www.arduino.cc/>

```
AnalogReadSerial | Arduino 1.8.8
Fichier Édition Croquis Outils Aide

AnalogReadSerial
*/

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

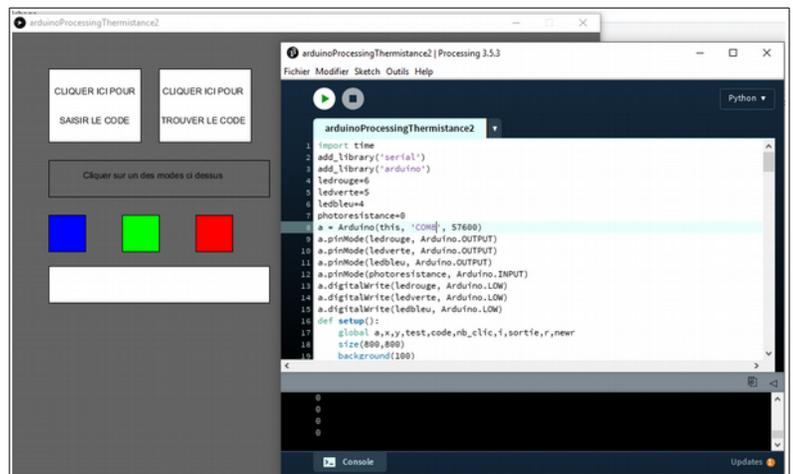
// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.println(sensorValue);
  delay(1);        // delay in between reads for stability
}
}
```

Processing :

Processing est un IDE développé par et pour des artistes, dans le but de programmer facilement, notamment des créations visuelles.

Processing permet de programmer en Python, et de créer des interfaces graphiques facilement.

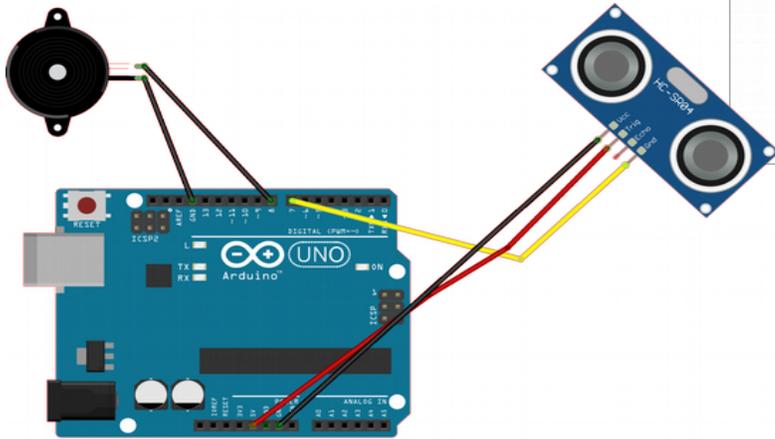
Ce logiciel permet également de communiquer avec une carte microcontrôleur Arduino en utilisant le module « **firmata** ».



Fritzing :

Logiciel de schématisation de circuits, avec une bibliothèque de composants intégrés.

<http://fritzing.org/home/>



fritzing



Tinkercad :

Application en ligne créée par Autodesk. <https://www.tinkercad.com>

Permet de réaliser des circuits virtuels avec une carte **Arduino**.

L'atout de **Tinkercad** est de permettre la simulation du fonctionnement du circuit. Nécessite la création d'un compte gratuit.



```
19 This example code is in the public domain.
20 */
21 int inches = 0;
22 int cm = 0;
23
24 long readUltrasonicDistance(int triggerPin, int echoPin)
25 {
26   pinMode(triggerPin, OUTPUT); // Clear the trigger
27   digitalWrite(triggerPin, LOW);
28   delayMicroseconds(2);
29   // Sets the trigger pin to HIGH state for 10 microseconds
30   digitalWrite(triggerPin, HIGH);
31   delayMicroseconds(10);
32   digitalWrite(triggerPin, LOW);
33   pinMode(echoPin, INPUT);
34   // Reads the echo pin, and returns the sound wave travel time in m
35   return pulseIn(echoPin, HIGH);
36 }
37
38 void setup()
39 {
40   Serial.begin(9600);
41 }
42
43
44
45
46
```

Below the code editor is a 'Moniteur série' button. The top of the interface shows the user name 'Sizzling Juttuli', the duration of the simulation 'Durée du simulateur: 00:00:01.965', and a status message 'Toutes les modifications ont été enregistrées'. There are also buttons for 'Code', 'Arrêter la simulation', 'Exporter', and 'Partager'.