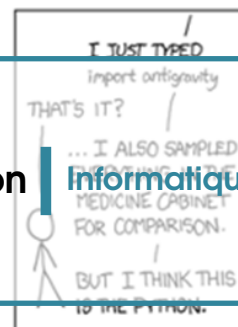
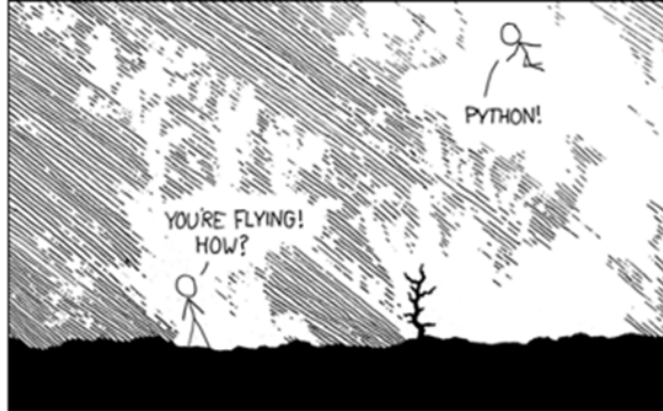
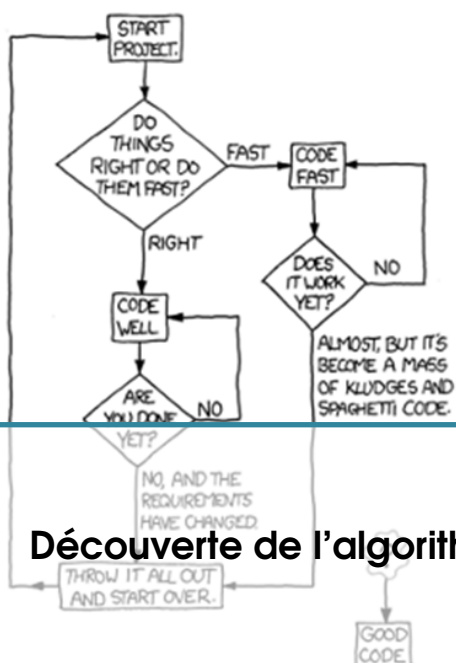


HOW TO WRITE GOOD CODE:



Découverte de l'algorithmique et de la programmation Informatique

Cours

Chapitre 0

Utilisation de Python

Savoirs et compétences :

1	Installation de PYZO sur votre ordinateur personnel	2
2	Changement de thème de la fenêtre Pyzo	2
3	IDLE python	2
4	Python via des notebook : Jupyter, Capytale, Google Colab	3
4.1	Jupyter	3
4.2	Capytale	3

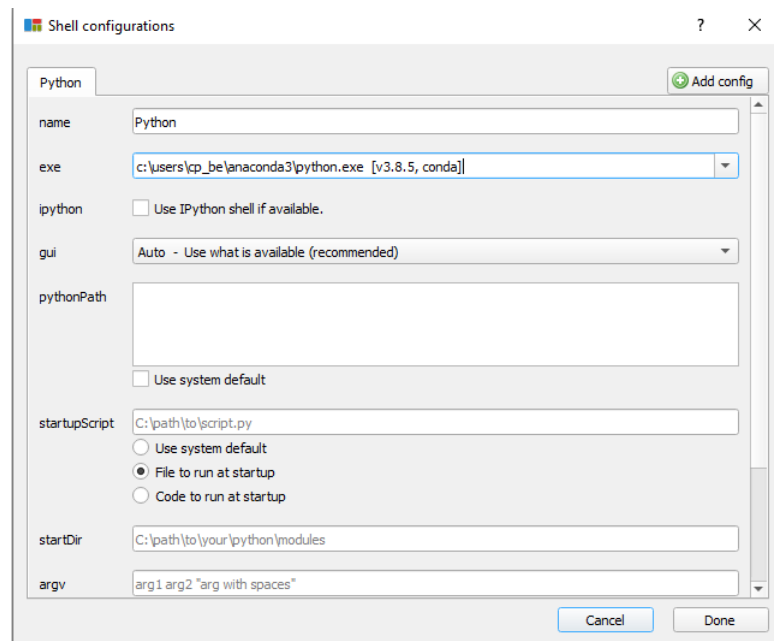
L'environnement de développement utilisé au lycée pour programmer en Python est Pyzo.

1 Installation de PYZO sur votre ordinateur personnel

Rendez-vous sur le site Pyzo.org qui vous propose une fenêtre Quickstart.

L'installation se fait en quatre temps selon les éléments choisis :

1. Installation de l'IDE, choisir la version correspondant à votre système d'exploitation (Windows, MacOS, Linux) en 32 ou 64 bits;
2. Installer la distribution Anaconda;
3. Configurer votre shell au lancement de Pyzo la première fois dans la fenêtre en haut à gauche;



4. Les bibliothèques sont installées par défaut, l'étape 4 n'est pas utile. Sauf si vous avez dû installer une distribution plus légère.

2 Changement de thème de la fenêtre Pyzo

Si l'écran blanc de la fenêtre Pyzo fatigue vos yeux, vous pouvez le changer :

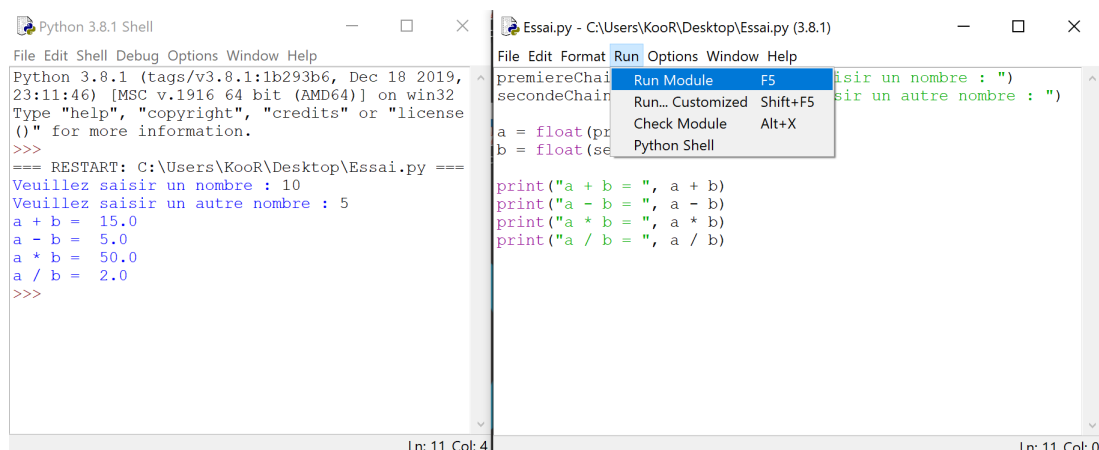
setting \ edit syntax styles... \

Vous pouvez choisir solarized_dark.

3 IDLE python

Vous serez amené en math-info à utiliser l'IDLE Python aussi installé sur les ordinateurs du lycée.

Moins convivial que Pyzo, il permet de réaliser les mêmes travaux en utilisant les deux types de fenêtres celle du shell (ou console) et celle d'édition.

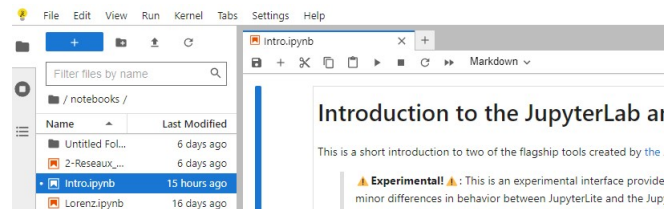


L'IDLE Python est disponible avec toute installation de Python. Avec Pyzo, aller dans dossier contenant Pyzo ou Anaconda puis dans le répertoire `Lib\idlelib` et lancer le programme `idle.bat`.
Le chemin peut être celui-ci `C:\Users\...\anaconda3\Lib\idlelib` par exemple.

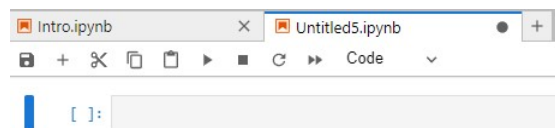
4 Python via des notebook : Jupyter, Cappytale, Google Colab

4.1 Jupyter

Jupyter.org est accessible par votre moteur de recherche préféré.
Je vous propose : Try Jupyter puis JupyterLab



Ouvrir une page vierge avec le + de l'onglet et sélectionner notebook et Python.



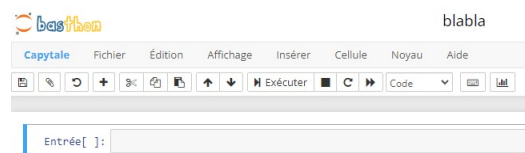
Vous pouvez écrire des lignes de codes dans la case et les exécuter avec la flèche.

4.2 Cappytale

Cappytale est accessible par l'ENT du lycée via ressources numériques. L'utilisation est la même que celle de Jupyter.
Vous y trouverez des travaux à réaliser en autonomie.



1. Un cadre institutionnel
2. Un environnement de travail standardisé conçu pour l'enseignement secondaire
3. Un service accessible sans installation avec un simple navigateur web
4. Une bibliothèque d'activités pédagogiques partagées entre enseignants



Application 01

Applications – Bases

Savoirs et compétences :



Découverte de quelques types dans l'interpréteur

Lancer PYZO.

Une fenêtre s'ouvre sur votre écran partagée en 3 zones, il est nécessaire d'ouvrir le shell en choisissant la version de python, alors quelques lignes de message introductif s'inscrivent. C'est l'**interpréteur** de Python, qu'on appelle aussi le **shell**.

Les symboles `>>>` invitent à taper quelque chose (invite de commande).

Fonctionnement : tapez une *instruction*, par exemple `1 + 1`, et tapez "Entrée" :

```
>>> 1 + 1
2
```

Différents types de nombres

Le shell de Python peut s'utiliser comme une calculatrice pour manipuler des nombres. Selon la nature (entiers, réels, complexes) des nombres que l'on souhaite manipuler, il y aura différents *types* sous python pour les représenter.

Ci-dessus, on a manipulé 1 qui est de type `integer` (*entier* en anglais).

On manipule aussi les "nombres à virgule", dits nombres *flottants*, dont le type s'appelle `float` en python :

```
>>> 3.1 + 10.09
13.19
>>> type(2.7)
<type 'float'>
```



Bien remarquer que l'on utilise la convention anglosaxonne : un point et non une virgule.

Exercice 1 –

Question 1 Taper les deux instructions suivantes : `3` , `3.0` puis `'3'` puis demander le type de ces données. Commenter

Question 2 Taper les trois instructions suivantes : `3.` , `0.1` puis `.1`. Commenter.

Exercice 2 –

Question 3 Testez les opérateurs suivants avec des entiers et/ou des flottants :

- * / // % ^ **

Question 4 Que dire du résultat et du type du résultat pour les opérateurs `/` , `//` et `%`, par exemple comment relier les résultats obtenus de ces deux opérations : `9/2` et `9%2`? L'opérateur `^` renvoie-t-il le résultat attendu? Que ce passe-t-il en cas d'opération entre un entier et un flottant?

Les fonctions `float` et `int` permettent d'obtenir une donnée du type voulu. Taper les instructions suivantes, où l'on voit des premiers exemples d'*affectations*, ici des *variables* `x` et `y` :

```
>>> x = 42\\
>>> x \\
..... \textit{à compléter !}\\✓
      cache{42}\\
>>> float(x)\\
..... \cache{42.0} \\
>>> y = -3.67\\
>>> y \\
..... \cache{-3.67} \\
>>> int(y)\\
..... \cache{-3}
```



`int` appliqué à un flottant ne fait pas un arrondi, ni ne calcule la partie entière : cela donne juste l'entier qui apparaît dans l'écriture décimale du flottant.

Les booléens

Une variable de type booléen ne peut prendre que deux valeurs : `True` ou `False`. Il s'agit souvent de propositions logiques plus ou moins longues que python peut évaluer. Quelques exemples :

```
>>> 1 > 0
True
>>> a = (1 > 3)
>>> a
False
>>> type(a)
<type 'bool'>
```

```
>>> x = 3.14
>>> y = 3.1
>>> x == y
False
```

Les opérateurs de comparaison disponibles sont :

$x == y$	x est égal à y
$x != y$	x est différent de y
$x > y$	x est strictement supérieur à y
$x < y$	x est strictement inférieur à y
$x >= y$	x est supérieur ou égal à y
$x <= y$	x est inférieur ou égal à y

Bien noter la différence entre $=$ (pour affecter une variable se trouvant à gauche du signe) et $==$ (pour définir une égalité qui sera un booléen).

Pour combiner des variables booléennes, on dispose de trois opérateurs logiques : `not`, `and`, `or`.

Exercice 3 –

Question 5 Créer une variable x égale à l'entier 3, et tester les trois opérateurs logiques ci-dessus sur les expressions $x < 4.1$ et $x \% 2 == 0$.

Question 6 Mathématiquement, à quoi correspond le booléen $x \% 2 == 0$?

Les chaînes de caractères

Python permet aussi de manipuler du texte. Le type utilisé s'appelle une chaîne de caractère : *string* en anglais, abrégé en `str` en Python. Il suffit de délimiter le texte par des apostrophes (*quote*) ou par des guillemets (*double quote*) :

```
>>> c = 'ptsi'
>>> d = "bonjour à tous"
```

On peut avoir accès aux caractères qui composent une chaîne de caractères; attention, en Python, ils sont numérotés à partir de 0, c'est-à-dire que le premier caractère est d'indice 0.

Question 7 Noter les résultats et commenter.

```
>>> c[0]
>>> c[2]
>>> c[-1]
>>> c[0] = 'm'
```

Trois autres choses à retenir sur le type `string` :

- la longueur d'une chaîne de caractères, c'est-à-dire le nombre de caractères qui la composent, s'obtient avec la fonction `len`.
Tester avec `c` et `d`.
- on peut concaténer des chaînes de caractères (c'est-à-dire les "accoler") à l'aide de `+`.
- comme pour les fonctions `int` et `float`, il existe une fonction `str` qui permet d'obtenir un objet de type `string`.

Question 8 Créer une chaîne de caractère `e` obtenue en concaténant `c` et `d`.

Question 9 Tester les instructions suivantes; comment modifier *la dernière* pour qu'il n'y ait pas de message d'erreur ?

```
>>> numero = 2
>>> f = c + numero
```

Remarque : Le fait qu'un même opérateur (ici `+`) puisse fonctionner différemment en fonction du contexte illustre le mécanisme appelé *surcharge des opérateurs*.

Remarque : Pour ne pas avoir à tout recopier à chaque ligne du *shell* en cas d'erreur...

Sous PYZO (windows), Avec la flèche \uparrow (éventuellement plusieurs fois successivement), on peut remonter dans l'historique de commandes.

Sous IDLE, Se placer sur la ligne à répéter et "entrée".