→ Lancer un navigateur internet, qui sera notre premier logiciel à utiliser aujourd'hui.
→ Lancer le programme "Spyder.exe" et revenez lire la suite le temps de patienter...

I Les bases de python

1.1 Le site d'exercices

En ce début d'année, et en tant que "révisions", nous utiliserons le site codingame (`https://www.codingame.com/playgrounds/53303/apprendre-python-dans-le-secondaire/cours---variables-et-operations`)

Exercice 1
Compléter la feuille 3, les 5 QCM et les 2 exercices

Exercice 2
Dans la feuille 4, étudier la partie I et traiter les QCM. Une fois ceci fait, vous pouvez reprendre le cours de ce TD

Exercice 3
Pour votre prochain TD d'informatique :

1. traiter en entier, chez vous, la feuille n°5 : attention, dans les deux exercices, il ne faut pas utiliser `print` mais `return` à la place. Nous expliquerons pourquoi dans l'exercice 6 !
2. traiter la feuille n°6.

II Environnement de travail

2.1 Les dossiers réseau

Vous disposez d'un espace de travail sur le réseau du lycée. Il s'agit , dans l'explorateur de fichier, d'un disque nommé "Home" qui se situe sur le réseau. Vous devrez enregistrer les fichiers nécessaires au travail en TD d'informatique dans un dossier de ce lecteur, et pas sur la machine locale (ie. pas sur le bureau ou dans tout autre sous-dossier de C:).

Clé USB

Il peut être utile d'apporter une clé USB en TD d'informatique, pour stocker vos documents et pouvoir les consulter chez vous par la suite.
Vous pouvez même y installer votre propre version de WinPython.

2.2 Bonnes pratiques

Dès le début d'un TD, créez un dossier dans votre répertoire personnel qui contiendra l'énoncé et éventuellement d'autres documents. "TD1" paraît être un nom convenable pour le TD de ce jour.

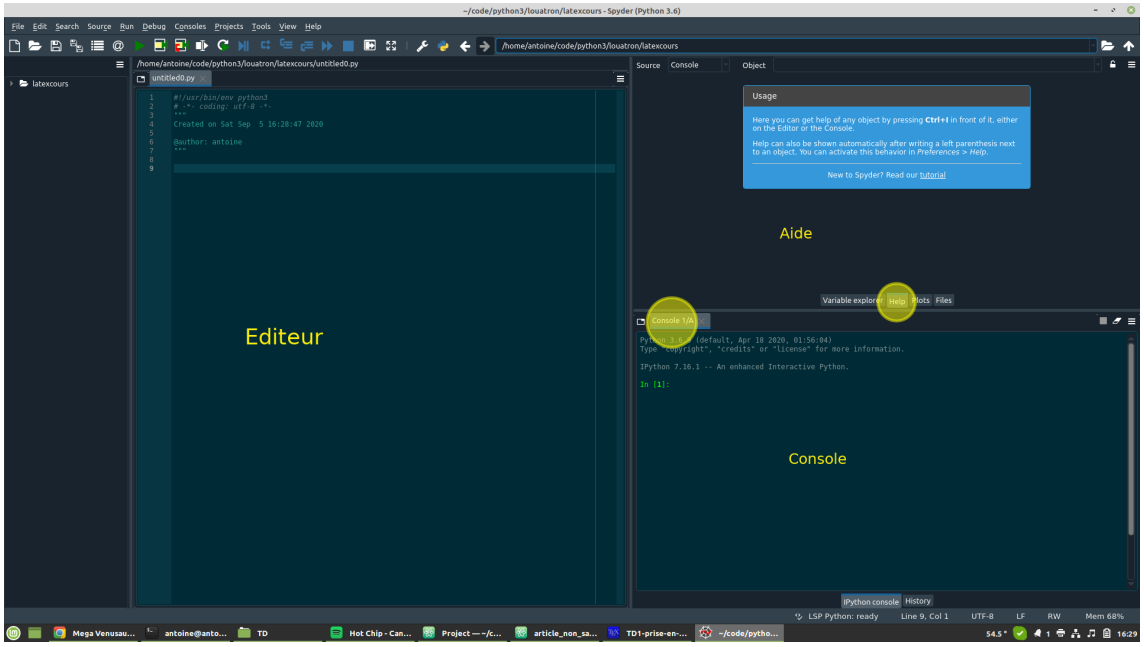
Vous aurez régulièrement des *scripts* python à créer (des fichiers texte ayant l'extension .py). Pensez à sauvegarder très régulièrement, éventuellement en utilisant le classique **CTRL-S** (ou le menu fichier).

2.3 L'IDE

Python, comme tout langage informatique, s'écrit dans un fichier texte. Il faut donc utiliser un éditeur de texte pour créer ou modifier les scripts.
Spyder sera notre éditeur par défaut. Mais ce programme est bien plus qu'un simple éditeur de texte. Vous pouvez voir au moins trois cadres

1. l'éditeur en lui-même, qui nous permettra d'écrire du texte dans un fichier
2. la console qui est le programme python.exe en lui même. Il s'agit d'un "interpréteur" (ou console) : vous tapez une ou plusieurs commandes (en langage Python évidemment) et l'interpréteur calcule le résultat de ces commandes.
3. l'inspecteur d'objet qui nous permettra souvent d'avoir accès à l'aide.

La gestion de ces cadres s'effectue via le menu "Affichage".



Coin Culture Un éditeur de texte qui possède en plus des fonctionnalités pour programmer (comme une console) est appelé un IDE pour Integrated Development Environment.

2.4 Aide de python

Il y a plusieurs sources d'aide pour programmer en Python. Dans l'ordre :

1. la documentation de python (menu Aide ou https://docs.python.org/3/)
2. l'aide intégrée à la console (cette commande est à taper dans la console, pour tester son effet) :

```
1  help(abs)
```

   Durant toute l'année, le contenu des cadres pourra être exécuté directement dans la console, pour observer l'effet. Méfiez-vous des copier-coller en provenance d'un PDF, ils réservent souvent de mauvaises surprises.
3. le professeur pendant les TD
4. Google...

III Activités

3.1 Autour des variables

Exercice 4
Quelle valeur contient la variable `x` après ces deux instruction ?

```
1  x = 4
2  x = x + x/2
```

Tester, dans la console. Pour vérifier la valeur contenue dans `x` deux méthodes : taper `x` puis entrée, ou alors `print(x)`. Maintenant, appuyer sur la flèche du haut (toujours dans la console). La dernière instruction réapparait, puis celle d'encore avant... Changer une nouvelle fois la valeur de `x` par l'instruction de la ligne 2 précédente, deviner la valeur contenue dans cette variable puis vérifier.

Exercice 5
On souhaite obtenir des valeurs de la suites définies par $u_0 = 1$ et $\forall n \in \mathbb{N}\ u_{n+1} = \frac{u_n}{2} + \frac{1}{u_n}$.

1. calculer à la main u_1, u_2, u_3 sous forme de fractions.
2. vérifier les calculs précédents grâce à python.
3. calculer u_6 et u_7 avec python. Indication : une seule variable est réellement nécessaire, et le but est de faire en sorte que cette variable prenne comme valeurs successives les différents termes de la suite. On pourra utiliser la flèche du haut pour répéter les opérations.

3.2 Autour des fonctions

Exercice 6
Dans l'éditeur, cette fois, taper

```python
def f1(x):
    return x**2

def f2(x):
    print(x**2)
```

Remarquez que certaines lignes sont indentées, c'est à dire commence par des espaces blancs. Pour obtenir ces espaces on peut utiliser la touche TAB.

1. Ces lignes de code python définissent deux fonctions `f1` et `f2` python. Pour l'instant nous ne pouvons pas utiliser ces fonctions car elles ont été tapées dans l'éditeur, c'est à dire dans un fichier de texte. Sauvegarder ce fichier dans le répertoire créé en début de TD (par exemple sous le nom *td1.py*).
2. une fois le fichier sauvegardé, on peut le faire exécuter en entier dans la console : pour ce faire, appuyer sur F5 (ou cliquer sur la flèche verte intitulée "Run file"). Une ligne commençant par `runfile....` doit apparaître dans la console. Pour simplifier, ce qui vient de se produire est équivalent au fait de taper toutes les lignes précédentes dans la console avant de valider par entrée.
3. Tester successivement (un test se fait uniquement dans la console) les commandes

   ```
   f1(3)
   f2(3)
   ```

   Quelle différence remarquez-vous ?
4. Tester la commande

   ```
   a = f2(3)
   ```

   puis vérifier la valeur contenue dans la variable `a`. Rien n'apparaît car `a` contient une valeur spéciale : `None`.

   Nous touchons ici du doigt la différence fondamentale entre `return` et `print` : un `print` ne fait qu'afficher le résultat à l'écran est n'est utile que pour l'humain en face de l'écran. `return` permet à une fonction de renvoyer/retourner une valeur. Testons avec

   ```
   a = f1(3)
   ```
5. Un des intérêts d'une fonction en informatique est de pouvoir être utilisée plusieurs fois pour répéter une calcul. Par exemple, on souhaite calculer $(3^2+4)^2$. En utilisant deux fois une des deux fonctions précédente, effectuer ce calcul dans la console. On pourra éventuellement stocker un résultat intermédiaire dans une nouvelle variable.

3.3 Bonus

Si tout le reste vous a paru simple

Exercice 7
Dans une nouvelle feuille Python créez :

- une fonction qui détermine si un entier est pair ou non. On retournera un booléen : `True` ou `False`
- une fonction **maxi** qui prend deux arguments numérique et retourne le plus grand des deux.
- une fonction qui étant donnés trois nombres a, b, c retourne les solutions de l'équation $ax^2+bx+c=0$ (pour retourner plusieurs valeurs depuis une fonction, on utilise un seul `return` et on sépare les différentes valeurs à retourner par des virgules).
- Sachant que le nombre complexe $a+ib$ avec $a, b \in \mathbb{R}$ se note `complex(a, b)` en python (ou encore `a + bj`), compléter la fonction précédente pour le cas où le discriminant est négatif, et en utilisant les nombres complexes de python.
- une fonction qui étant donné un entier n retourne la plus grande puissance de 2 qui soit inférieur ou égal à n.
- une fonction qui, étant donné un entier n, affiche successivement les chiffres de l'écriture décimale de n en commençant par le chiffre des unités.