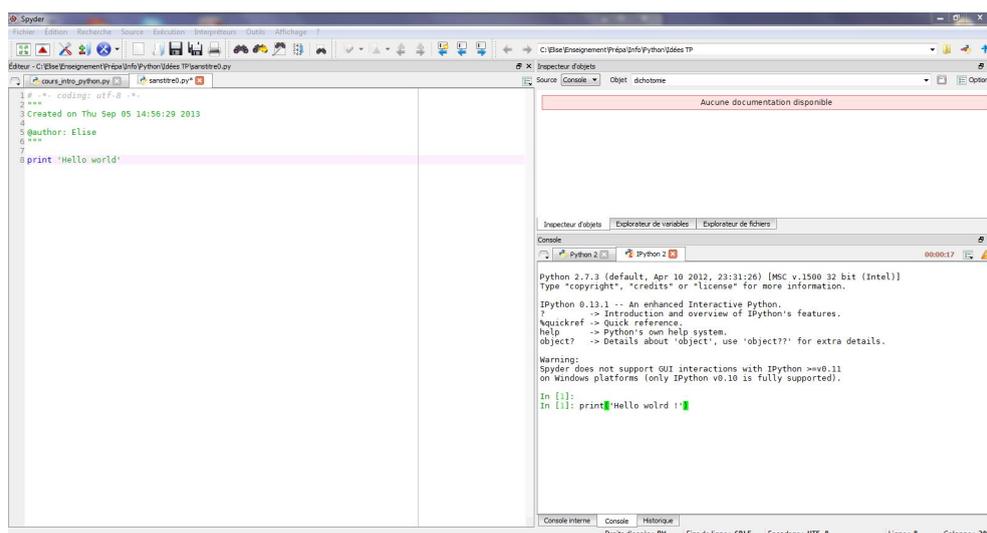


Découverte de quelques commandes utiles en python

Pyzo, sur les ordinateurs du lycée, ou Spyder chez vous si vous avez téléchargé Anaconda, est un environnement de développement intégré (IDE) : cette interface permet d'écrire des programmes avec des facilités (couleurs, indentations automatiques, auto-complétion, messages d'avertissement, etc).

Cette interface est composée de 3 fenêtres :

- Une fenêtre permettant d'ouvrir un fichier .py ou .txt et de le modifier : c'est l'**éditeur**
- Une fenêtre d'**aide** (Inspecteur d'objets, explorateur de variables, explorateur de fichiers, affichage de graphiques...)
- La fenêtre de l'**interpréteur** appelée aussi la **console** : c'est dans cette fenêtre que se font les calculs.



Le code est écrit dans l'éditeur (on n'oubliera pas de sauvegarder le fichier régulièrement).

Lorsqu'on veut le compiler, c'est-à-dire que l'on veut que l'ordinateur effectue les calculs correspond à ce code, on peut :

- Appuyer sur F5 : le code est compilé dans son intégralité, le résultat des calculs apparaît dans la console.
- Surligner une partie seulement du code et appuyer sur F9 : seules les lignes sélectionnées sont compilées.
- Appuyer sur la flèche verte dans la barre de menu, où est indiqué "Run". L'intégralité du code est compilée.

Si le code est modifié dans l'éditeur, il faut le compiler à nouveau pour que les modifications soient bien prises en compte.

Dans tout ce qui suit, votre travail consiste à taper les commandes proposées dans l'éditeur, à les compiler, à regarder le résultat qui s'affiche dans la console et à noter en commentaire à côté du code à quoi sert la commande. Si un message d'erreur apparaît, notez-le et essayez de le comprendre.

I. Bases

```
1 print(2)
```

```
1 print(a)
```

```
1 a=3
```

```
2 print(a)
```

```
1 L=[2,5,8,12]
```

```
2 print(L)
```

```
1 print(type(L))
```

```
2 print(len(L))
```

```
1 print(L[0])
```

```
1 print(L[2])
```

```
1 print(L[-1])
```

```
1 print(3**2,4**2,5**2)
```

```
1 print(abs(-3), abs(2))
```

II. Bibliothèque

```
1 print(cos(1))
```

La commande précédente ne permet pas de calculer $\cos(1)$ car la fonction \cos n'est pas encore importée. Pour l'importer, il faut importer la bibliothèque `math` :

```
1 import math
```

```
2 print(math.cos(1))
```

Pour éviter d'avoir à taper `math` devant chaque fonction de la bibliothèque `math`, on peut utiliser un "alias" :

```
1 import math as m
```

```
2 print(m.cos(1))
```

On peut aussi utiliser la commande suivante

```
1 from math import * #importer toutes les fonctions de math
```

```
2 print(cos(1))
```

Bonne pratique : La bibliothèque math sera importée sans alias (cf dernière commande). Les autres seront toujours importées avec les alias vus ci-après. Cela évitera les confusion entre 2 fonctions de même nom, ne faisant pas partie des mêmes bibliothèques et ne fonctionnant pas de la même façon.

```
1 print(pi)
2 print(exp(0))
3 print(log(1))
4 print(1e3)
5 print(2e-3)
```

III. Numpy

```
1 import numpy as np #on importe la bibliothèque numpy sous l'alias np
```

```
1 L1=[1,2,3,5]
2 L2=np.array([1,2,3,5])
3 print(L1)
4 print(L2)
5 print(type(L1))
6 print(type(L2))
```

```
1 print(2*L1)
```

```
1 print(2*L2)
```

```
1 print(L1+1)
```

```
1 print(L2+2)
```

Vous avez dû remarquer le fonctionnement différent des listes et des "tableaux numpy" (array). Les tableaux permettent des calculs sur tous les éléments de la liste, ce qui correspond à la fonctionnalité qu'on retrouve dans un tableur. Ce sont donc des tableaux numpy que nous utiliserons par défaut en physique.

```
1 print(cos(L1))
```

```
1 print(cos(L2))
```

```
1 print(np.cos(L2))
```

La fonction cos de numpy peut s'appliquer à un tableau entier, tandis que la fonction cos de la bibliothèque math ne s'applique qu'à un réel.

```
1 L3=np.ones(10)
2 print(L3)
```

```
1 L4=np.zeros(5)
2 print(L4)
```

IV. Numpy.random

```
1 import numpy.random as rd
2 #on importe le module random de la bibliothèque numpy sous l'alias rd
```

```
1 L5=rd.uniform(0,1,10)
2 print(L5)
```

```
1 L6=rd.normal(0,1,20)
2 print(L6)
```

V. Matplotlib

```
1 import matplotlib.pyplot as plt
2 #on importe le module pyplot de la bibliothèque matplotlib
3 #sous l'alias plt
```

```
1 liste_x=np.array([1,2,3,4,5])
2 liste_y=np.array([1,4,9,16,25])
3 plt.plot(liste_x,liste_y,'+b')
4 plt.show()
```

```
1 liste_x=np.array([1,2,3,4,5])
2 liste_y=np.array([1,4,9,16,25])
3 plt.plot(liste_y,liste_x,'*r')
4 plt.show()
```