

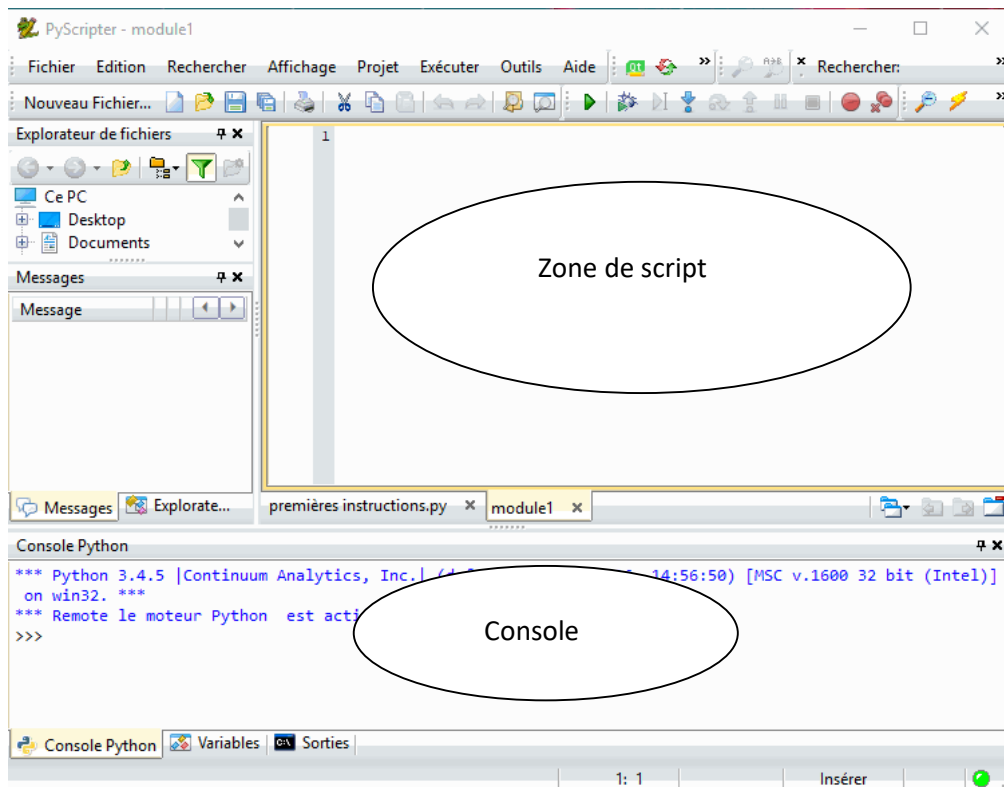
ECT1



TP 01: Prise en Main

Nous allons ensemble apprendre Python, un langage bien connu des scientifiques, des startups et des amateurs d'un certain groupe d'humoristes britanniques. Oui, vous avez bien lu ! Pour expliquer l'origine du langage, revenons un peu en arrière. En 1989, par une froide nuit néerlandaise, un développeur du plat pays nommé Guido van Rossum s'ennuie. Il cherche un moyen de s'occuper pendant la période qui précède Noël car les bureaux de son entreprise sont fermés. Quand certains auraient dévoré l'intégrale de sa série préférée ou décoré un sapin, lui se lance dans l'invention d'un langage. Étant un grand fan de la troupe de comiques " **Monty Python** " et d'humeur irrévérencieuse, il l'appelle Python. Voilà pourquoi les développeurs Python ont de l'humour et s'amuse à glisser des petites blagues dans leur code !

Exercice 1 : Découverte de Edupython



L'interface est découpée en plusieurs zones :

- ❑ La zone de script permet de saisir des programmes plus longs et de les exécuter ensuite.
- ❑ La console qui permet d'exécuter en direct des commandes Python et qui affiche les résultats des programmes.

Si vous installez un autre éditeur vous aurez un fonctionnement analogue.

Dans ce premier TP on s'intéresse à la console.

La saisie des instructions dans la console se fait après les `>>>` que l'on appelle **chevrons**.

Exercice 2 : Python peut être utilisé comme une calculatrice

1- Évaluez dans la console l'instruction suivante :

`1+5` . En déduire l'interprétation de ce symbole

2- Évaluez dans la console, de manière successive, les instructions suivantes :

`6-4` `4*7` `8/2` `9/2` `5**2`

En déduire l'interprétation de ces symboles en particulier / et **

3- Dans la console, vérifiez que rien n'est inscrit devant l'invite de commande `-->`.

Appuyez alors plusieurs fois sur la touche  puis sur la touche .

À quoi servent ces touches ?

4- Évaluez dans la console, de manière successive, les instructions suivantes :

`9//2` et `9%2`

En déduire l'interprétation de ces symboles

- 5- Évaluez dans la console, l'instruction suivante
`2 - 9 #les espaces sont optionnels`
En déduire l'interprétation du symbole « # ».

- 6- Évaluez dans la console, l'instruction suivante
`7+3*4`
`(7+3)*4`
La hiérarchie des opérations mathématiques est-elle respectée ?



Bilan 1 : Priorité des opérations

PEMDAS pour le mémoriser !

1. **P** pour parenthèses. Ce sont elles qui ont la plus haute priorité. Elles vous permettent donc de « forcer » l'évaluation d'une expression dans l'ordre que vous voulez.
Ainsi $2*(3-1) = 4$, et $(1+1)**(5-2) = 8$.
2. **E** pour exposants. Les exposants sont évalués avant les autres opérations.
Ainsi $2**1+1 = 3$ (et non 4), et $3*1**10 = 3$ (et non $3^{10} = 59049$!).
3. **M** et **D** pour multiplication et division, qui ont la même priorité. Elles sont évaluées avant l'addition **A** et la soustraction **S**, lesquelles sont donc effectuées en dernier lieu.
Ainsi $2*3-1 = 5$ (plutôt que 4), et $2//3-1 = -1$
4. Si deux opérateurs ont la même priorité, l'évaluation est effectuée de gauche à droite. Ainsi dans l'expression $59*100//60$, la multiplication est effectuée en premier, et la machine doit donc ensuite effectuer $5900//60$, ce qui donne 98. Si la division euclidienne était effectuée en premier, le résultat serait 59

Remarque : le symbole # (se lit « croisillon », « hash » en anglais, symbole proche du « dièse ») permet de faire figurer dans le corps du programme un commentaire qui ne sera pas pris en compte lors de son exécution. # porte sur le reste de la ligne

- 7- Évaluez dans la console, les instructions suivantes
`1,5*2`
`1.5*2`
En déduire comment noter les nombres décimaux

Exercice 3 : Limite de la calculatrice

1- Évaluez dans la console, de manière successive, les instructions suivantes :

pi e sqrt (4) log (e)

Les fonctions intégrées au langage sont relativement peu nombreuses : ce sont seulement celles qui sont susceptibles d'être utilisées très fréquemment. Les autres sont regroupées dans des fichiers séparés que l'on appelle des **modules ou bibliothèques**. Durant les 2 années de prépas, vous verrez un certain nombre de bibliothèques.

Ainsi exponentielle, logarithme népérien, pi, racine carrée ne sont pas intégrées directement dans python. Elles sont regroupées dans la bibliothèque

Pour importer une bibliothèque, vous disposez de 2 Méthodes:

Méthode 1: `from numpy import *` qui importe la totalité de la bibliothèque. C'est rapide mais très coûteux en mémoire.

2- Taper à la console `from numpy import *` puis évaluez dans la console, de manière successive, les instructions suivantes :

pi e sqrt (4) log (e)

3- Fermer edupython puis l'ouvrir à nouveau puis évaluez dans la console, de manière successive, les instructions suivantes sans taper autre chose

pi e sqrt (4) log (e)

Que se passe-t-il?

Méthode 2: `import numpy as np` qui importe la bibliothèque sous l'acronyme np. C'est rapide mais très peu coûteux en mémoire. C'est cette méthode qui est privilégiée dans les sujets.

- 4- Taper à la console `import numpy as np` puis évaluez dans la console, de manière successive, les instructions suivantes :

`pi` `e` `sqrt(4)` `log(1)` `log10(10)`

- 5- Évaluez dans la console, de manière successive, les instructions suivantes :

`np.pi` `np.e` `np.sqrt(4)` `np.log(1)` `np.log10(10)`

- 6- Évaluez à la main l'instruction `np.sqrt(3)**2` puis vérifiez votre réponse en tapant l'expression à la console. Comment expliquer cette différence?

- 7- Taper à la console `import numpy as lala` puis évaluez dans la console, de manière successive, les instructions suivantes :

`pi` `e` `sqrt(4)` `log(1)`



Vous ne devez pas faire preuve d'originalité mais utilisez les acronymes du programme. Ainsi numpy sera toujours noté np

Exercice 4 : Pour vous entraîner

Calculer les expressions suivantes : noter les instructions et les résultats.

a) 2^{12}

b) $\sqrt{2}$

c) e^2 de 2 manières

d) $\ln(e)$

e) le nombre d'or $\frac{1+\sqrt{5}}{2}$