

Initiation à Python (sous Spyder)

1) Présentation de Spyder

Spyder est un environnement de programmation (IDE). Plusieurs sous-fenêtres apparaissent à l'écran. Repérez celles dont nous nous servirons le plus :

- la console (Python ou IPython) : permet d'interpréter directement des commandes Python ;
- l'éditeur : on peut y rédiger des programmes (plusieurs lignes) et les exécuter ensuite dans la console ;
- l'inspecteur de variables : donne des informations sur les variables stockées en mémoire.

2) Python comme calculatrice

Placez-vous dans la console et exécutez les commandes suivantes :

```
>>> 4+6
>>> 3+2*5
>>> 2**3
>>> 5/2
>>> 5//2
>>> 5%2
```

- Q1) Python respecte-t-il les priorités dans les calculs ?
- Q2) Peut-on utiliser des parenthèses dans les calculs ? Des crochets ?
- Q3) Que renvoient les commandes `//` et `%` ?

On peut également manipuler des nombres complexes, saisissez la commande : `>>> a=1+2j`

Vous avez créé une variable `a` à laquelle vous avez affecté la valeur `1+2j` (comme en sciences physiques, `j` désigne un nombre dont le carré vaut `-1`).

Vérifiez que `a` est apparu dans le gestionnaire de variables et saisissez la commande `>>> print(a)` pour afficher le contenu de la variable `a`.

Testez les commandes suivantes :

```
>>> a.real
>>> a.imag
>>> abs(a)
```

Remarquez que le symbole `=` a été utilisé pour affecter une valeur à la variable `a`. Lorsque l'on voudra utiliser l'opérateur booléen de test d'égalité, on utilisera `==`. Testez les commandes suivantes :

```
>>> a==1+2j
>>> a==4-5j
>>> a=3
>>> print(a)
```

3) Encore des calculs

Exécutez la commande `>>> cos(0)`, vous obtenez un message d'erreur expliquant que la fonction `cos()` n'est pas définie.

Pour utiliser des fonctions plus élaborées que les opérations arithmétiques, il faut les importer depuis une bibliothèque, c'est-à-dire une collection d'outils, prêts-à l'emploi.

Saisissez les commandes

```
>>> from math import cos
>>> cos(0)
>>> cos(pi)
```

Q4) Comment lever cette difficulté ?

Notez que vous pouvez obtenir des informations sur les fonctions :

```
>>> help(cos)
```

Vous pouvez également obtenir de l'aide sur le module lui-même dès lors que vous l'avez importé :

```
>>> import math
>>> help(math)
```

Les fonctions des modules sont alors reconnues par Python mais pour les utiliser il faut se servir de la syntaxe `module.fonction` :

```
>>> math.sin(0)
```

Q5) Utilisez Python pour trouver une valeur approchée de $e^{\sqrt{2}}$.

4) Chaînes de caractères

Les chaînes de caractères sont saisies entre guillemets (simples ou doubles, les deux sont acceptés).

```
>>> m='hello world'
```

On peut accéder aux différents caractères de la chaîne à l'aide de leurs indices, attention en Python les indices commencent à 0.

```
>>> m[3]
>>> m[0]
>>> m[-1]
```

Q6) Etant donnée une chaîne de caractères `C` :

- que va produire l'appel `C+C` ?
- que va produire l'appel `5*C` ?
- que renverra `len(C)` ?
- que va renvoyer `C[-2]` ?

5) Listes

Pour saisir une liste sous Python on utilise les crochets avec les virgules comme séparateur.

```
>>> l=[1 , 2+3j, 'toto']
```

On accède aux éléments des listes de la même façon qu'aux caractères d'une chaîne, c'est-à-dire à l'aide de leur indice (0 pour premier indice).

Q7) Que va renvoyer `l[-1][1]` ?

Pour ajouter un élément à une liste `L` on utilise la méthode `append` :

```
>>> L.append(nouvelElement)
```

Q8) La commande `>>> L=[L,nouvelElement]` permet-elle de rajouter `nouvelElement` à `L` ?

Q9) Comment savoir le nombre d'éléments d'une liste `L` ?

6) Type d'une variable

Saisissez les commandes suivantes :

```
>>> c=1
>>> type(c)
>>> d=1.0
>>> type(d)
>>> e=1+0j
>>> type(e)
```

Q10) Qu'appelle-t-on *type* d'une variable ?

Q11) A votre avis, que vont renvoyer les commandes suivantes ?

```
>>> c+d+e
>>> c==d
>>> c==e
```

Python gère les types de façon sophistiquée, il tend à rendre cette notion transparente pour l'utilisateur. Sous d'autres langages (C par exemple), on ne peut additionner ou comparer des variables de types différents et il faut déclarer le type de la variable lorsqu'on la crée (on dit que Python gère le typage de façon dynamique).

Q12) Listez le maximum de types différents en essayant de décrire à quoi ils correspondent.

Q13) Dans le type `complex`, `j` désigne un nombre dont le carré vaut `-1`.

Tester la commande : `>>> j**2`

Comment expliquer le résultat obtenu ? Comment reformuler la commande pour vérifier que le carré de `j` est `-1` ?