

I. Avant tout

Allumer l'ordinateur et se connecter avec ses identifiants personnels.

Lancer Spyder (icône sur le bureau). Inutile de cliquer 50 fois : il est effectivement long à ouvrir, mais plus vous cliquez, plus ça sera long.

Pendant qu'il s'ouvre, dans « Mes documents », créer un dossier informatique, puis un sous dossier TP1.

Tous les documents de cours et de TP sont disponibles sur <http://icpaulconstans.free.fr>.

II. Premiers pas avec Spyder

Il existe plusieurs Environnements de Développement Intégré (EDI, ou IDE en anglais). Dans ce cours nous utiliserons principalement Spyder. C'est un logiciel libre, disponible sur <https://github.com/spyder-ide/spyder> ou plus simplement par Anaconda (<http://continuum.io/downloads#34>) qui regroupe tout ce dont on a besoin, quelque soit le système d'exploitation ou la version de Python souhaitée.

Attention : si vous voulez installer Spyder sur votre ordinateur perso par Anaconda, vérifiez bien que vous téléchargez la version Python 3.4 (et pas 2.7) adaptée à votre système d'exploitation (Windows, Mac OS ou Linux en 32 ou 64 bit).

La fenêtre de Spyder est divisée en trois :

- à gauche, l'éditeur dans lequel on tapera les programmes ;
- en haut à droite, trois onglets possibles :
 - l'inspecteur d'objets dans lequel une documentation s'affiche. On configurera Spyder pour que cette documentation s'affiche dès qu'on ouvre une parenthèse ;
 - l'explorateur de variables qui nous servira à visualiser la valeur des variables utilisées, principalement pour debugger les programmes que l'on tapera ;
 - l'explorateur de fichier pour parcourir l'arborescence des répertoires et choisir les fichiers à ouvrir.
- en bas à droite, plusieurs onglets sont possibles, mais le seul qui nous intéresse est l'onglet Console (si l'onglet ouvert est Console IPython, il faut le fermer. Normalement, il ne s'ouvrira pas au prochain lancement de Spyder). Cette console permet de taper directement des instructions Python (souvent pour faire des tests, ou de petits calculs), et c'est aussi dans cet onglet que l'on pourra visualiser l'exécution des programmes tapés dans l'éditeur.

1) Configuration

Beaucoup de choses sont configurables dans Spyder. Nous allons commencer par en fixer quelques une. Vous pourrez toujours changer ce comportement par la suite. Pour cela, dans le menu Outils, aller dans Préférences.

- Dans l'onglet Exécuter, choisir dans Interpréteur « Exécuter dans un nouvel interpréteur dédié » et dans Répertoire de travail « le répertoire du fichier à exécuter ». Décocher aussi l'option permettant de toujours afficher la fenêtre Options d'exécution lors de la première exécution d'un script.
- Dans l'onglet Répertoire de travail global, sélectionner au Démarrage le dossier info que vous avez dû créer dans votre répertoire perso sur le réseau. Pour les autres fichiers, sélectionner « le répertoire du fichier en cours d'édition ».
- Dans l'onglet Console, Options avancées, choisir l'interpréteur par défaut et le script PYTHONSTARTUP par défaut.
- Dans l'onglet Inspecteur d'objets, cocher toutes les options pour afficher l'aide dès l'ouverture d'une parenthèse.

2) La console

Dans la console, on peut taper directement des instructions Python qui s'exécutent immédiatement (à condition d'appuyer sur <Entrée> en fin de ligne). La division s'obtient avec /. On peut faire des divisions entières avec //, et calculer des modulo avec %. La puissance s'obtient par ** (attention de ne pas confondre avec ^ qui calcule un ou exclusif). Le résultat du dernier calcul est rappelé par _ (sur la touche 8) :

```
>>> 3 + 4
7
>>> 5 * 14
70
>>> _ +5
75
```

```
>>> 11 / 4
2.75
>>> 11 // 4
2
>>> 11 % 4
3
>>> 2 ** 10
1024
>>> 2 ^ 10
8
```

On peut aussi introduire des commentaires en commençant par un # :

```
>>> 3 * 5 # Un calcul très compliqué!!!
15
```

On peut aussi affecter des variables avec =, afficher du texte avec `print` :

```
>>> a = 3
>>> b = 5
>>> a + b
8
>>> a = a - 1
>>> a = b
>>> b = a
>>> print('Salut, monde!')
Salut, monde!
```

Exercice 1 : Après avoir tapé les lignes ci-dessus, quelles sont les valeurs de a et b? (Rappel : un des onglets s'appelle Explorateur de variables...) Quelle suite d'instructions faudrait-il taper pour échanger les valeurs de deux variables?

3) L'éditeur

L'éditeur sert à taper des programmes (en Python, on parle aussi de scripts) plus complets, que l'on pourra sauvegarder dans des fichiers dont l'extension est .py.

Exercice 2 : Dans l'éditeur, taper le programme suivant :



```
print("Bonjour")
x = 42
print(x)
```

Observer la coloration syntaxique, l'ajout automatique de parenthèse fermante, la coloration des paires de parenthèses correspondantes.

Sauvegarder ce programme dans un répertoire adapté (rappelez-vous les répertoires créés au début de ce TP...). Essayez de donner un nom explicite à vos dossiers et fichiers, en évitant autant que possible les lettres accentuées et les espaces. Exécuter ensuite ce programme (soit par le menu Exécution, soit en cliquant sur le bouton ►, soit plus simplement en appuyant sur F5). On peut éventuellement configurer l'interpréteur dans lequel le programme est exécuté, et l'interaction avec cet interpréteur après la fin du programme en appuyant sur F6 ou en cliquant sur 🐘. L'intérêt de pouvoir choisir différents interpréteurs ou de choisir d'interagir après l'exécution est de que, tant que l'interpréteur est ouvert, on peut accéder aux variables utilisées dans le programme. Faire quelques essais en ouvrant plusieurs consoles grâce au menu Consoles, en définissant des variables différentes dans chaque console et en exécutant des scripts, et observer l'Explorateur de variables.

4) Le débogueur

Le débogueur permet d'exécuter un programme pas à pas, soit pour mieux comprendre ce qu'il fait, soit plus souvent pour trouver pourquoi il ne fait pas ce qu'on en attend (ce qu'on appelle un bogue, ou un bug en anglais). Pour cela, on

lance le débogueur soit par le menu Déboguer, soit en cliquant sur , soit en appuyant sur Ctrl+F5. On exécute ensuite chaque ligne du programme en cliquant sur  ou en appuyant sur Ctrl+F10.

Exercice 3 : Taper le programme suivant dans l'éditeur, puis utiliser le débogueur pour l'exécuter pas à pas, et surveiller la valeur des différentes variables.

```
x = 3
y = 10
z = y / (x + y)
y = y - x
z = y // x
```

III. Expressions et types simples

1) différents types

En python, les variables sont typées, autrement dit elles sont classées selon l'objet qu'elles représentent. Une expression peut ainsi être de type entier, flottant, chaîne de caractères, ...

On obtient le type d'une expression en utilisant la fonction `type`. Par exemple, taper dans la console :

```
type(42)
type(3.5)
type('Bonjour')
x = 3
type(x)
message = 'Voulez-vous rejouer ?'
type(message)
```

2) les entiers

En python, les entiers sont stockés en précision illimitée, ce qui signifie qu'on peut manipuler des entiers aussi grand que l'on veut. Les opérations sur les entiers sont l'addition (+), la soustraction ou l'opposé (-), la multiplication(*), la division (/), la division entière(/), le modulo(%) et l'exponentiation (**). Les ordres de priorité sont les mêmes qu'en mathématiques.

Exercice 4 : Sans afficher tous les chiffres, donner les deux derniers chiffres de 2015^{2015} .

3) les flottants

Les nombres à virgule flottante, ou plus simplement flottants, sont la représentation informatique des nombres décimaux (on y reviendra dans le cours sur la représentation des nombres). Les opérations sont les mêmes que sur les entiers.

Exercice 5 : Dans la console, taper `2**100` et `2.0**100`. Expliquer la différence.

Exercice 6 : Taper `1-1/3-1/3-1/3` ou `1-0.2-0.2-0.2-0.2-0.2` ou encore `2.2+1.1`. Que faudra-t-il retenir?

On peut convertir des expressions d'un type à un autre. Par exemple `float(42)` donne `42.0`. On peut aussi taper directement `42.0`, ou même `42.` sans écrire le zéro. Réciproquement, `int` renvoie la troncature en direction de zéro.

Exercice 7 : Prévoir le résultat de `8.5 / 2.5`, de `int(8.5) / int(2.5)` et de `int(8.5 / 2.5)`. Vérifier.

Exercice 8 : Que renvoie `float(7//2)` ?

Exercice 9 : Étudier la fonction `round` (on a parlé d'un onglet inspecteur d'objet non?). Taper `round(1.05, 1)`, puis `round(10.05, 1)` et `round(100.05, 1)`. On expliquera ces résultats en cours en détaillant la façon dont les flottants sont stockés en mémoire.

4) les booléens

Un booléen ne peut prendre que deux valeurs : `True` et `False`. Il permet de représenter le résultat d'une expression logique. Par exemple, l'expression `x % 2 == 0` sera `True` (vraie) si `x` est pair, et `False` (fausse) sinon.

Les opérations sur les booléens sont la négation (le non logique) `not`, la conjonction (le et logique) `and` et la disjonction (le ou logique) `or`.

On peut aussi tester des inégalités, voir des encadrements. Par exemple :

```
1 < 3 #inégalité stricte
1 <= 3 #inégalité large
1 >= 3
1 + 1 != 3 # non égal
1 <= 3 < 5 #encadrement
1 < 3 < 9 == 3 * 3 <= 10 #voire des choses plus compliquées
```

Exercice 10 : Prévoir la valeur des expressions booléennes suivantes, lorsqu'elle existe. Vérifier les réponses dans une console.

```
3 * 3.5 > 10
3. * 7 == 21
3 - 1 => 1
0 < 10**-300 == 100**-150
not (2-1 == 1 == 4+3)
not (True and False)
not (not True)
(5.5 * 2 == 11. or 1/2 != .5) and (3 % 2 == 0)
```

Exercice 11 : Écrire des expressions booléennes traduisant les conditions suivantes. Les nombres mentionnés sont tous des flottants.



- Le point de coordonnées (x, y) est à l'intérieur du cercle de centre (z, t) et de rayon r .
- Les points de coordonnées (x, y) et (z, t) sont situés sur une droite parallèle à l'un des axes du repère.
- Les points de coordonnées (x, y) et (z, t) sont les sommets opposés d'un carré dont les côtés sont parallèles aux axes du repère.
- Il existe un triangle dont les côtés mesurent respectivement a , b et c .

Exercice 12 : Écrire des expressions booléennes traduisant les conditions suivantes. Les nombres mentionnés sont tous des entiers.

- L'entier n est divisible par 5.
- Les entiers m et n sont de même signe.
- n est le plus petit multiple de 7 supérieur à 10^{100} .
- Les entiers m et n sont tels que l'un est multiple de l'autre.
- Les trois entiers m , n et p sont de même signe.
- Les trois entiers m , n et p sont distincts deux à deux.

IV. Pour ceux qui ont fini :

Lancer un navigateur internet (un vrai, pas MS...) et aller sur <http://www.france-ioi.org>

Dans la colonne de gauche, dans le menu Connexion, inscrivez-vous soit en utilisant  ou  si vous avez un compte Google ou bien un compte Facebook, soit directement en cliquant sur Inscription.

Il faut ensuite compléter votre nom et prénom dans la fiche publique, avant d'aller dans le menu **Plan du site**, **Enseigner**, **Groupes et classes**, rejoindre le groupe **PTSI Paul Constans 2016-2017**. Le mot de passe est « taupe ».

Vous pouvez alors aller dans **Cours et problème**, et soit faire des exercices niveau 1 ou 2, soit si vous connaissez suffisamment bien Python, débloquent les exercices niveau 3, soit si vous êtes vraiment fort en Python et en algorithmique essayer de débloquent directement les exercices du niveau 4.