



Prise en main de Spyder

Objectifs :

- Prendre en main Spyder
- Savoir faire un programme simple et l'exécuter

Exercice I

1. Pour taper un programme, on commence par créer un projet (qui contient le code tapé dans des modules, des images, des sons,..) : Créer un projet nommé *PriseEnMain* (menu Fichier/Nouveau Projet), à l'intérieur un package nommé *premiersprogrammes* (clic droit à gauche sur le nom du projet dans l'explorateur de projets/Nouveau/Paquet), puis un module nommé *recopie* (clic droit sur le nom du paquet dans l'explorateur de projets/Nouveau/Module)
2. Le but est de recopier le programme ci-dessous dans le module :
 - Taper les deux lettres *fr*, appuyer simultanément sur les touches CTRL+Espace, et valider le choix *from* : on vient d'utiliser l'autocomplétion de Spyder.
 - Si on attend un peu, un petit losange rouge apparaît à gauche de la ligne : Spyder analyse le code en temps réel et a détecté une erreur de syntaxe. Survoler le losange rouge avec la souris : on a un message d'erreur.
 - Finir de taper la première ligne et attendre : un petit losange orange apparaît à gauche de la ligne. En survolant le losange avec la souris, on obtient un warning. Ce n'est pas une erreur car il n'empêche pas le programme de fonctionner, mais il indique que le code pourrait être mal conçu. Le mot clé *import* permet d'utiliser les fonctions qui sont dans un module (ici le module *math*).
 - Finir de taper les trois premières lignes. Quand on valide après *:*, le code est automatiquement indenté (décalé vers la droite) : cette indentation permet de délimiter le bloc de code qui se trouve dans le "if". Il faut scrupuleusement respecter les indentations (à l'espace près).
 - Taper la quatrième ligne et valider. Pour sortir du bloc se trouvant dans le "if", il faut désindenter : taper sur la touche de suppression.
 - Taper le reste du programme.
 - Pour avoir de l'aide sur la fonction *raw_input*, cliquer dessus dans le code et appuyer simultanément sur CTRL+I : lire le résultat dans l'inspecteur d'objets (à droite, cliquer sur l'onglet correspondant si nécessaire). Deuxième méthode : taper *help(raw_input)* (dans la console interactive).

```
from math import sqrt

a = int(raw_input("Entrer un entier: "))
if a>0 :
    print sqrt(a)+1
else :
    print a+5
```

3. Exécuter le programme (flèche verte dans la barre d'outils en haut), saisir une valeur (dans la console) en validant, puis lire le résultat. Que fait ce programme ?
4. **Ne pas hésiter à relancer la console de temps en temps pour nettoyer la mémoire** contenant les variables : cliquer sur le triangle orange au-dessus de la console, puis sur le triangle vert au-dessus de la console.

Exercice II

1. Dans le même package que précédemment, créer un module nommé *suite* (clic droit sur le nom du package dans l'explorateur de projets/nouveau/module)
2. Soit (u_n) la suite telle que $u_0 = 2$ et $u_{n+1} = 2u_n - n - 3$ pour $n \in \mathbb{N}$. Compléter et taper le programme ci-dessous afin qu'il affiche u_p où l'entier p est saisi par l'utilisateur :

```
p = int(raw_input("Entrer un entier p: "))
u = 2
n = 0
while n < p :
    u =
    n =
print u
```

Combien vaut u_4 ?

3. Créer un nouveau module nommé `somme_suite`, recopier le programme précédent dedans et l'adapter afin qu'il affiche la somme $S_p = \sum_{k=0}^p u_k$. Combien vaut S_{10} ?
4. Créer un nouveau module nommé `seuil` et écrire un programme qui affiche le plus petit entier p tel que S_p est inférieur à -10^{10} .