

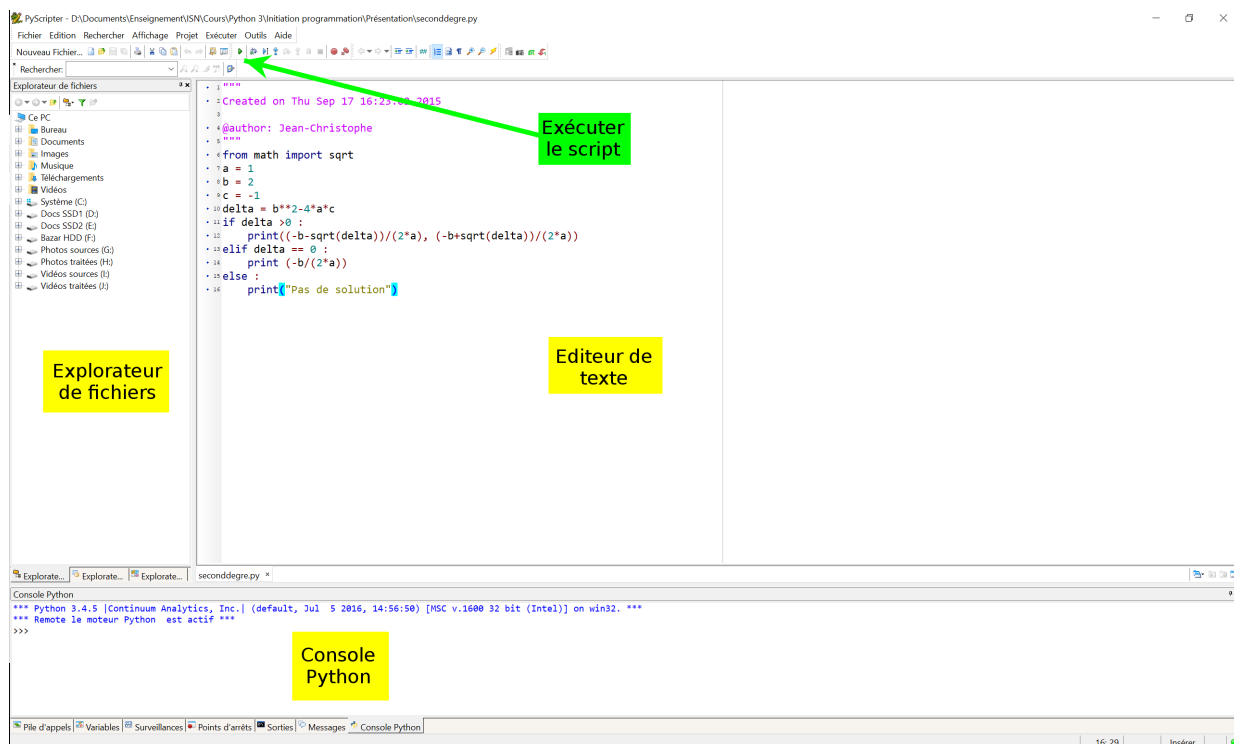
Utilisation de Pyscripter

Objectifs :

- Connaître une brève description de Pyscripter
- Installer Pyscripter avec Edupython
- Démarrer avec Pyscripter
- Connaître la structure d'un programme Python type

1 Description de Pyscripter

Un simple éditeur de texte pourrait suffire pour taper des programmes Python mais ça ne serait pas pratique. Pyscripter est un IDE (integrated development environment) dont le but est de créer des programmes en Python de manière plus efficace.



- Il comporte un éditeur de texte qui permet de taper les scripts en Python. Les principales caractéristiques de l'éditeur sont :
 - La coloration automatique du code.
 - L'indentation automatique des blocs.
 - L'autocomplétion : on tape le début d'un mot et une liste apparaît.
 - L'affichage des erreurs de syntaxe en temps réel (elles sont soulignées).
 - L'exécution du script (bouton Exécuter le script, qui lance l'interpréteur Python).
- Une console Python :
 - C'est dans cette console qu'on affiche les résultats avec la fonction *print*.

- Il est possible de taper directement du Python dans la console notamment pour obtenir de l'aide : `help(nom_fonction)`, `help(nom_module)` ou `dir(nom_objet)` (pour obtenir la liste des méthodes applicables à un objet). La console peut s'utiliser comme une calculatrice.
- Les messages d'erreur sont aussi affichés dans la console (voir aussi l'onglet Messages en bas de l'écran).
- Un débogueur (possibilité d'effectuer l'exécution ligne par ligne en observant l'évolution des valeurs des variables).
Pour utiliser le débogueur :
 - On crée un point d'arrêt sur une ligne (clic sur le numéro de la ligne)
 - On lance le débogage (bouton Debugger avec un insecte) ce qui exécute le script jusqu'au point d'arrêt
 - On exécute le script pas à pas tout en inspectant l'évolution des variables dans la fenêtre Surveillances (onglet Surveillances au bas de l'écran). Pour ajouter une variable à surveiller, cliquer droit dans la fenêtre Surveillances, ajouter un point de surveillance, taper le nom de la variable.

2 Installation, utilisation de Pyscripter

En ISN, on installe en une seule fois Python (version 3.4) et Pyscripter grâce à la distribution *Edu-python* qui comporte aussi de nombreuses bibliothèques (voir le site de *Edupython*). Cette distribution peut-être installée sur une clé USB.

Pour commencer à taper du code Python :

- On crée un dossier principal nommé ISN qui contiendra tous les projets. Puis dans l'explorateur de fichiers, cliquer droit sur ISN/Explorateur de fichiers/Explorer ici.
- Pour chaque projet, on crée un dossier (qui va regrouper tous les fichiers liés au projet : code source, images, sons...) : dans l'explorateur de fichiers, cliquer sur le dossier ISN puis sur Créer un nouveau dossier en haut de la fenêtre. Choisir un nom explicite pour le dossier.
- On crée un (des) module(s) dans le dossier précédent et on y ajoute les fichiers images, sons...

Il vaut mieux choisir des noms explicites pour les dossiers et les modules. **Par convention les noms de modules sont en minuscules. Les noms de modules pourront contenir des `_` (underscores) pour séparer les mots.**

3 Structure d'un programme Python type

Résumé : structure d'un programme Python type

```
# -*- coding: latin-1 -*-
```

```
#####  
# Programme Python type  
# auteur : G.Swinnen, Liège, 2009  
# licence : GPL  
#####
```

```
#####  
# Importation de fonctions externes :
```

```
from math import sqrt
```

```
#####  
# Définition locale de fonctions :
```

```
def occurrences(car, ch):  
    "Cette fonction renvoie le \  
    nombre de caractères <car> \  
    contenus dans la chaîne <ch>"
```

```
    nc = 0
```

```
    i = 0
```

```
    while i < len(ch):
```

```
        if ch[i] == car:  
            nc = nc + 1
```

```
        i = i + 1
```

```
    return nc
```

```
#####  
# Corps principal du programme :
```

```
print "Veuillez entrer un nombre :"  
nbr = input()
```

```
print "Veuillez entrer une phrase :"  
phr = raw_input()  
print "Entrez le caractère à compter :"  
cch = raw_input()
```

```
no = occurrences(cch, phr)  
rc = sqrt(nbr**3)
```

```
print "La racine carrée du cube",  
print "du nombre fourni vaut",  
print rc
```

```
print "La phrase contient",  
print no, "caractères", cch
```

Un programme Python contient en général les blocs suivants, dans l'ordre :

- Quelques instructions d'initialisation (importation de fonctions et/ou de classes, définition éventuelle de variables globales).
- Les définitions locales de fonctions et/ou de classes.
- Le corps principal du programme.

Le programme peut utiliser un nombre quelconque de fonctions, lesquelles sont définies localement ou importées depuis des modules externes.

Vous pouvez vous-même définir de tels modules.

La définition d'une fonction comporte souvent une liste de PARAMÈTRES.

Ce sont toujours des VARIABLES, qui recevront leur valeur lorsque la fonction sera appelée.

Une boucle de répétition de type 'while' doit toujours inclure au moins quatre éléments :

- l'initialisation d'une variable 'compteur' ;
- l'instruction while proprement dite, dans laquelle on exprime la condition de répétition des instructions qui suivent ;
- le bloc d'instructions à répéter ;
- une instruction d'incréméntation du compteur.

La fonction "renvoie" toujours une valeur bien déterminée au programme appelant.

Si l'instruction 'return' n'est pas utilisée, ou si elle est utilisée sans argument, la fonction renvoie un objet vide : 'None'.

Le programme qui fait appel à une fonction lui transmet d'habitude une série d'ARGUMENTS, lesquels peuvent être des valeurs, des variables, ou même des expressions.