

# Informatique I

## Introduction à la programmation

### TP 01

Le but de ce TP est de vous montrer comment vous pouvez installer Python sur votre ordinateur personnel et de vous familiariser de manière informelle, par des exemples et des exercices intuitifs, à la programmation en langage Python.

#### Installation de Python sur votre ordinateur personnel

Veuillez accéder dans un navigateur web au site [www.python.org](http://www.python.org), cliquez sur “Download” dans le menu de la navigation à gauche de la page, puis

- **sur Mac OSX:**  
téléchargez le package d'installation [Python 2.7.3 Mac OS X 32-bit i386/PPC Installer](#), lancez-le et suivez les instructions.
- **sur Windows:**  
téléchargez le package d'installation [Python 2.7.3 Windows Installer](#), lancez-le, suivez les instructions.

**Attention:** pour cela, vous aurez besoin de droits d'administrateur.

#### Introduction

Dans les exemples suivants, les entrées (au clavier) et les sorties (à l'écran) se distinguent par la présence, ou respectivement l'absence, des invites ">>>": afin de reproduire l'exemple, vous devez taper tout ce qui se trouve après l'invite et avant le caractère délimitant un commentaire (#). Toutes les commandes se terminent par un retour de chariot (return). Les lignes qui ne commencent pas par une invite sont une réponse (sortie) de l'interpréteur Python.

La plupart des exemples contiennent des commentaires. Les commentaires en Python commencent par le caractère dièse, "#", et s'étendent à la fin de la ligne. Un commentaire peut commencer au début d'une ligne ou à la suite d'espaces ou de code, mais pas à l'intérieur d'une chaîne littérale (chaîne de caractères ou STRING) délimitée par des guillemets. Un dièse au sein de la chaîne littérale est juste un dièse.

Un exemple:

```
# Ceci est le premier commentaire
>>> spam = 1      # et c'est le deuxième commentaire
                  # ... et maintenant un troisième!
>>> string =     "# Ceci n'est pas un commentaire."
```

## Exercice 1: Lancement de Python

Le package d'installation contient les outils nécessaires pour commencer à apprendre Python. Il s'agit de l'interpréteur du langage et d'un éditeur de code qui forment un environnement de développement IDLE.

Pour lancer l'interpréteur, cliquez sur l'icône IDLE qui se trouve

- **sur Mac OSX (y compris les ordinateurs de la salle du tapis vert):**  
dans le dossier *Applications* → *Python2.7*.
- **sur Windows Vista/7:**  
dans le menu *Démarrer* → *Tous les programmes* → *Python2.7*.



Une fois l'IDLE chargé, vous aurez l'invite ">>>". Tous ce que vous taperez dès lors sera interprété par Python. Vous pouvez quitter l'interpréteur Python avec la commande "exit()" ou en appuyant Ctrl+D.

Vous pouvez placer l'icône de l'IDLE sur le *bureau* ou dans le *dock* afin qu'il soit plus aisé de le trouver pour les semaines à venir.

```
Python Shell
Python 2.7.2 (v2.7.2:8527427914a2, Jun 11 2011, 14:13:39)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> print "salut!"
salut!
>>> |
```

## Exercice 2: Nombres entiers et décimaux

L'interpréteur peut également agir comme une simple calculatrice: vous pouvez taper une expression et l'interpréteur vous retournera la valeur résultante. La syntaxe est simple: les opérateurs +, -, \* et / fonctionnent comme dans la plupart des calculatrices, des parenthèses peuvent être utilisées pour le groupement. Rappelez-vous de ne taper que les lignes **commençant par une invite** et de les **exécuter avec un retour de chariot**.

Essayez les exemples suivants:

```
>>> 2 + 2
4
>>> 2+2      # sans les espaces
4
>>> 2 +2     # et un commentaire sur la même ligne que le code
4
>>> (50-5 * 6) / 4
5
```

**Attention:** La division de 2 entiers retourne un entier, arrondi à l'entier inférieur ;

```
>>> 7 / 3 # la division entière retourne:
2
>>> 7/-3
-3
```

Le reste de la division entière est obtenu avec l'opérateur modulo "%":

```
>>> 7 % 3
1
```

Si vous voulez faire une division décimale, au moins l'un des 2 membres de l'expression doit être décimal.

```
>>> 1 / 4.0
0.25
>>> 3.0 / 10
0.3
>>> 10.0/3.0      # soyez attentif aux arrondis
3.3333333333333335
```

Pour travailler avec des arrondis, vous pouvez utiliser la fonction `round`.

```
>>> round(10.0/3)
3.0
>>> round(10.0/3,2)
3.33
```

Le signe égal ("=") est utilisé pour assigner une valeur (numérique ou chaîne de caractères) à une **variable**. Notez qu'aucun résultat n'est affiché avant la prochaine interaction:

```
>>> largeur = 20
>>> hauteur = 5 * 9
>>> largeur * hauteur
900
```

Une valeur peut être affectée à plusieurs variables simultanément. Vous pouvez afficher la valeur d'une variable simplement en tapant son nom suivi d'un return:

```
>>> x = y = z = 0      # zéro est affecté à x, y et z
>>> x
0
>>> y
0
>>> z
0
```

Notez que les noms de variables sont sensibles à la casse (majuscule):

```
>>> Age=18
>>> age=25
>>> Age
18
>>> age
25
```

Vous pouvez évidemment assigner le résultat d'une opération utilisant des variables à une autre variable (ou à une variable existante):

```
>>> x = 1.25
>>> y = 3 * 5 / 2.5
>>> z = x * y
>>> y
6.0
>>> z
7.5
>>> x = x * 2
>>> x
2.5
```

### Exercice 3: Chaînes de caractères (string)

En plus des nombres, Python peut aussi manipuler des chaînes de caractères, qui peuvent s'exprimer de plusieurs façons. Elles peuvent être délimitées par des apostrophes ou des guillemets:

```
>>> 'spam eggs'          # des apostrophes
'spam eggs'
>>> 'doesn\'t'           # introduire une apostrophe dans la chaîne
"doesn't"
>>> "doesn't"           # ou comme ceci
"doesn't"
>>> '"Yes," he said.'    # introduire des guillemets dans une chaîne
'"Yes," he said.'
>>> "\"Yes,\" he said."  # ou comme ça
'"Yes," he said.'
>>> 'Isn\'t," she said.' # et mélanger le tout...
'Isn\'t," she said.'
```

Une chaîne de caractère littérale peut contenir des retours de chariots (pour être sur plusieurs lignes) sous la forme d'un "\n". Ceux-ci ne seront interprétés qu'avec la fonction print.

```
>>> hello = "bonjour \n tout le monde"
>>> hello          # montre le contenu de hello
'bonjour \n tout le monde'
>>> print(hello)   # affiche le contenu de hello
bonjour
tout le monde
```

Les chaînes de caractères peuvent être concaténées (collées ensemble) avec l'opérateur "+" et répétées avec "\*":

```
>>> word = 'Help' + 'A'
>>> word
'HelpA'
>>> '<' + word*5 + '>'
'<HelpAHelpAHelpAHelpAHelpA>'
```

Pour concaténer des éléments qui ne sont pas (tous) des chaînes de caractères, vous devez utiliser la fonction str() qui permet de convertir toutes sortes de types de données en chaînes de caractères. Ainsi:

```
>>> word = "HelpA"
>>> x = 2.5
>>> word + " " + str(x)
'HelpA 2.5'
```