

TP de Langages de script n° 1 : premiers pas en PYTHON

Coordonnées des enseignants

Peter Habermel <Peter.Habermehl@liafa.univ-paris-diderot.fr>

Sophie Laplante <Sophie.Laplante@liafa.univ-paris-diderot.fr>

Anne Micheli <Anne.Micheli@liafa.univ-paris-diderot.fr>

Christine Tasson <Christine.Tasson@pps.univ-paris-diderot.fr>

Documents à télécharger

Tous les énoncés et les documents à télécharger seront disponibles sur didel

<http://didel.script.univ-paris-diderot.fr> (le code du cours est LS42013). Inscrivez-vous dès la première semaine pour faciliter la communication avec l'équipe enseignante.

Version de Python Nous utiliserons la version 3 de python qui n'est pas compatible avec les versions antérieures. De ce fait beaucoup d'exemples que vous pourrez trouver sur internet demanderont de petites modifications pour pouvoir être interprétés avec la version utilisée.

Exercice 1 : Premier programme.

Ce premier exercice a pour but de découvrir les différents moyens d'utiliser Python.

1. Lancez la version 3 de l'interpréteur Python (mode interactif) grâce à la commande `python3` depuis un terminal. Écrivez et exécutez une commande pour afficher la ligne de texte **Hello World!**. Quittez l'interpréteur en tapant `^D` (touches **Control** et **D** simultanément) ou en appelant la fonction `exit()`.
2. Créez un fichier `hello.py` contenant comme unique ligne de texte la commande précédente. Sauvegardez ce fichier, et tapez dans un terminal la commande `python3 hello.py`. Observez le résultat.
3. Lancez à nouveau l'interpréteur Python, et tapez la commande `import hello`. Que se passe-t-il? Grâce à la commande `dir()`, affichez la liste des identificateurs connus. Que constatez-vous?
4. Rendez le fichier `hello.py` exécutable et tentez de l'exécuter. Que se passe-t-il? Rajoutez la ligne `#!/usr/bin/env python3` au début du fichier, puis réessayez.

Exercice 2 : Mode calculette.

Placez-vous dans un interpréteur Python.

1. Faites les calculs $3/4$, $4/2$. Que constatez-vous? Essayez encore $3//4$, $4//2$. Quelle est la différence?
2. Calculez le reste de la division de la somme de 19875 et 77569 par 7. Si vous trouvez un nombre supérieur à 6, c'est évidemment qu'il y a une erreur... Que pouvez-vous en déduire sur les opérateurs `+` et `%`? Qu'en est-il des autres opérateurs arithmétiques (faites des tests)?
3. Faites le calcul $\sqrt{3} + 56/9.0 \times |-1/4|$, soit en une seule fois, soit en plusieurs fois en utilisant des variables (**note** : consultez l'aide sur les nombres entiers (`help(int)`) et sur l'opérateur "puissance" (`help("POWER")`) pour trouver la syntaxe nécessaire).
4. Affichez la valeur de la variable `_` (tiret bas). À quoi sert-elle à votre avis? Expérimentez.
5. Testez les instructions `1j**2`, `(1+2j).imag` et `(1+2j).real`. Que représente `j`?

Par la suite, vous ne vous servirez du mode interactif uniquement pour tester des petites commandes et consulter l'aide en ligne. Vous éditerez vos programmes dans un fichier que vous exécuterez via le terminal.

Exercice 3 : Chaînes de caractères.

1. Créez une variable `h` contenant la chaîne de caractères "Hello" et une variable `w` contenant "World". En utilisant la concaténation de chaînes de caractères et ces deux variables (entre autres), créez une nouvelle variable `hw` contenant la chaîne "Hello World!". En utilisant la fonction `print`, affichez le contenu de la variable `hw`.
2. Testez la fonction `len(chaine)`. A quoi sert-elle ?
3. En utilisant la syntaxe `chaine[debut:fin]` et la variable `hw`, affichez les chaînes "Hell", "orld!" et "llo Wo". Testez ensuite les instructions `print(hw[:4])`, `print(hw[-4:])`, `print(hw[:])` et `print(hw[:4] + hw[4:])`. Qu'observez-vous ?
4. Créez une variable contenant la chaîne de caractères suivante (en respectant les retours à la ligne) :
 Dans le vieil étang,
 Une grenouille saute,
 Bruit dans l'eau.
Utilisez d'abord des guillemets triples (''' ou """), puis des guillemets simples.
5. Consultez l'utilisateur à l'aide de la fonction `input`, puis utilisez cette fonction pour lire un message donné par l'utilisateur, enfin affichez ce message à l'écran. Testez ensuite la commande
 `print(input("Ca va ?"))`
6. La fonction `int(objet)` permet de traduire un objet (par exemple une chaîne) en valeur entière (lorsque c'est possible). Écrivez une suite d'instructions permettant de lire un entier puis d'afficher le **double** de sa valeur.

Exercice 4 : Première boucle, première fonction

1. Créez une variable `x` de valeur 100. Calculez la somme S_x de tous les entiers de 1 à x (en les additionnant tous). Vérifiez que la formule $S_x = x(x+1)/2$ pour cette somme est satisfaite.
2. Que signifient `range(10)` et `range(3, 40, 5)` ? Confirmez votre opinion.
3. Programmez deux fonctions : `somme_bete(x)` et `somme_rapide(x)` qui calculent la somme de tous les entiers de 1 à x avec les deux méthodes vues dans le premier point.
4. Testez vos fonctions pour quelques valeurs de `x`.
5. Documentez vos fonctions. Placez vous dans un interpréteur et importez votre fichier. Qu'affiche l'aide en ligne `help(somme_bete)` ?