

Nota : sur la page web http://www.i3s.unice.fr/~bernot/Enseignement/GB3_Python1 vous trouverez, avec un retard de quelques jours par rapport au cours, des notes de cours et les feuilles de TD.

1 Installation de Python

Vérifiez tout d'abord que Python ne soit pas déjà installé sur votre machine. Si oui, passez cette partie du TD...

- Cherchez (avec un moteur de recherche, par exemple « startpage ») la page officielle de Python et choisissez la version stable la plus récente.
 - Ne pas utiliser les « sources » : ce sont des textes de programmes qui doivent être compilés pour fabriquer les commandes utiles pour le langage Python (python, IDLE, *etc.*). Mieux vaut utiliser les installations déjà compilées, qui sont alors dépendantes du système que vous utilisez : Windows, Linux, Mac OS 10, *etc.*
 - Chargez celle qui convient. Cela installera dans vos menus diverses commandes utiles pour Python.
- Lancez parmi ces commandes « IDLE ».

- Une fenêtre apparaît, avec « >>> » et le curseur qui attend que vous commenciez à programmer, et donnera les résultats au fur et à mesure de vos commandes.
- Dans le menu de IDLE, *File/New* permet d'accéder à un éditeur. Dans cet éditeur on peut modifier à loisir un texte de programme en Python mais, contrairement à la première fenêtre IDLE, cela ne l'exécute pas au fur et à mesure des lignes tapées. Pour exécuter le programme, il faut choisir *Run* dans le menu de l'éditeur : cela demande un nom de fichier la première fois, où le texte du programme sera sauvegardé avant d'être exécuté.
- Ultérieurement, dans le menu de IDLE, *File/Open* permet d'éditer un ancien fichier de programme de son choix. Il faut donc choisir des noms de fichier « parlants »... Mieux : prévoyez un répertoire sur votre ordinateur réservé aux TD du cours de bonnes pratiques de la programmation.

2 Premières fonctions et procédures en Python

Exercice 1 : Écrivez en python une fonction `carre` qui prend en entrée un nombre entier relatif (de type `int`) et retourne en sortie son carré.

Par exemple `carre(5)` retourne 25.

Faites des essais d'utilisation avec plusieurs valeurs entières.

Faites aussi un essai avec une valeur réelle (type `float`) : ça marche aussi! pourquoi à votre avis?

Exercice 2 : Écrivez en python une procédure `double` qui prend en entrée un nombre entier relatif `n` et imprime à l'écran "`le double de ... est ...`" qui indique à combien est égal le double de `n`.

Par exemple `double(6)` imprime à l'écran "`le double de 6 est 12`" (mais ne retourne aucun valeur en tant que fonction).

Faites 2 versions, l'une utilisant la concaténation `+`, l'autre avec l'opération de substitution `%`.

Exercice 3 : Quels résultats donneraient respectivement les expressions `1 + carre(2)` et `1 + double(2)` ?

Vérifiez vos prédictions *après* les avoir formulées...

Exercice 4 : Écrivez une fonction `triangle` de 3 arguments `a`, `b` et `c` qui indique si ces 3 réels définissent les côtés d'un triangle, en suivant le procédé suivant : le plus grand des côtés doit être inférieur à la somme des deux autres.

À l'occasion de ces premiers exercices, on découvre entre autres :

- que l'indentation (les marges) doivent impérativement être scrupuleusement respectées,
- qu'il faut une ligne vide à la fin d'un « `def ... :` » pour que l'ordinateur comprenne qu'on a fini la définition de fonction,
- que les calculs que l'on écrit doivent tous respecter une forme « arborescente » dans laquelle le type retourné par un sous-arbre de calcul doit coïncider avec celui attendu par chaque fonction en argument, en particulier il ne faut pas écrire « `a >= b and c` » qui est mal typé si `a`, `b` et `c` sont des entiers, alors que l'on voulait écrire « `a >= b and a >= c` »
- *etc.*

3 Quelques autres fonctions et procédures en Python

Exercice 5 : Écrivez en python une fonction `direAge` qui prend en entrée une chaîne de caractères `p` (qui sera en fait un prénom) et un entier `n` (sa date de naissance), et qui retourne une chaîne de caractères en suivant l'exemple suivant :

`direAge("Max",1996)` retourne la chaîne "Max a 21ans."

Si la différence entre l'année courante (2018) et la date de naissance est supérieure à 150, la fonction devra imprimer à l'écran une erreur indiquant que la date de naissance est improbable.

Exercice 6 : Écrivez en python une fonction `filtre` qui prend en entrée un entier relatif `n`, qui imprime à l'écran un message d'erreur si `n` est négatif, qui retourne `n` lui même s'il est inférieur à 1000 et la moitié (entière) de `n` s'il est supérieur à 1000.

Par exemple `filtre(-36)` imprime `Erreur!` à l'écran; `filtre(358)` retourne 358, `filtre(1050)` retourne 525 et `filtre(1051)` aussi.

Exercice 7 : Écrivez en python une procédure `ecrit_ligne` qui prend en entrée une chaîne de caractères et l'imprime à l'écran sous réserve qu'elle fasse moins de 50 caractères de long. Si elle fait plus de 50 caractères, la procédure écrit seulement la ligne « TROP LONG! ».

Par exemple, `ecrit_ligne("ligne assez courte.")` imprime à l'écran « ligne assez courte. ». Par contre, si on lui donne en entrée "012345678901234567890123456789012345678901234567890123456789glop" la procédure `ecrit_ligne` imprime à l'écran « TROP LONG! ».