Initiation à Python - leçon 3.1.1

s1
Dans cette séquence, nous allons découvrir les notions de variable et d'objet et celle de typage dynamique. Dans les leçons suivantes, nous introduirons les types de base.

---
s2

Python est un langage de programmation orienté objet. Revenons d'abord sur la notion d'objet, qui si elle est bien comprise facilitera votre apprentissage de Python.
Nous avons dit (dans la partie 2.1) qu'en informatique, un objet représente un concept. Reprenons l'exemple d'un avion. Un objet avion possède des caractéristiques qu'on appelle attributs, tels que par exemple sa vitesse et le nombre de sièges passagers ; un avion a d'autres caractéristiques que l'on appelle méthodes et que l'on peut comprendre comme des fonctions permettant de modifier ses attributs, comme ici la fonction accélère que l'on doit bien entendu définir et qui on le devine va changer l'attribut vitesse.
Pour l'ordinateur, cet objet représente un programme qui s'exécute. Ce programme contient toutes les informations liées à cet objet, attributs et méthodes.

De base, Python est capable de gérer plusieurs types d'objets simples. Nous allons les découvrir dans la leçon suivante. Prenons l'exemple d'une chaîne de caractères, la chaîne "bonjour" que l'on délimite en Python par des guillemets. Saisir cette chaîne puis faire un retour chariot lance l'interpréteur. Python reconnaît l'instruction comme une chaîne de caractères - c'est ce qu'on appelle le typage dynamique - et crée dans la mémoire de l'ordinateur un objet de type chaîne de caractères. Cet objet contient d'emblée plusieurs informations : d'abord un attribut qui contient l'information "bonjour", pour simplifier appelons cet attribut valeur. La valeur de ce nouvel objet est "bonjour". Mais également, puisque cet objet est de type chaîne de caractères, Python lui attribue toutes les méthodes associées aux objets de type chaîne de caractères - c'est-à-dire un jeu de fonctions qui va pouvoir s'appliquer à cet objet.
Parmi ces méthodes, on trouve par exemple la méthode upper() qui si elle est appliquée à la chaîne "bonjour" va réécrire sa valeur en majuscules. Comment utilise-t-on ces méthodes ? Tout simplement en les accolant à un objet et en les faisant précéder d'un point, comme dans l'exemple présenté. Le point est une manière générique d'appeler une méthode sur un objet.
Je vous propose de reproduire cette ligne de commande.

---
s3

Dans la mémoire de l'ordinateur, un objet est référencé par une adresse mémoire. Une adresse mémoire est un nombre entier naturel codé sur un certain nombre de bits qui désigne une zone particulière de la mémoire.
Heureusement, lorsque l'on programme, on ne manipule pas les objets à travers ces adresses, mais grâce à des variables. Pour affecter un objet à une variable, on utilise le signe égal. Dans cet exemple présenté, Python effectue trois choses : il crée une variable a dans l'espace des variables ; il crée un objet de type chaîne de caractères ; il référence l'objet dont la valeur est 'bonjour' grâce à la variable a. À chaque fois que l'on utilisera la variable a dans une instruction Python, c'est de l'objet référencé par a qu'il s'agira.

La variable référence un objet de type chaîne de caractères. On peut appliquer à cet objet la méthode upper() par le biais de la variable a qui référence cet objet.
Je vous propose de reproduire cette ligne de commande.

Les variables n'ont pas de type ; elles ne contiennent pas d'objet, mais référencent des objets qui eux ont un type.

Si l'on écrit maintenant a = 1, Python effectue plusieurs choses : il crée en mémoire un objet de type entier ; il supprime la référence entre a et l'objet dont la valeur est 'bonjour' et crée une nouvelle référence entre a et l'objet entier 1. La mémoire nécessaire pour stocker 'bonjour' est récupérée.

L'instruction b=a, crée une autre variable b, mais ne recrée par l'objet 1. b et a référencent le même objet 1.

Un petit mot sur la fonction print(). La fonction print() d'une variable affiche à l'écran la valeur d'un objet référencé par cette variable.
Je vous propose de reproduire cette ligne de commande.

---
s4

Quelques règles concernant les noms que l'on peut utiliser pour les variables. Un nom de variable peut être une suite quelconque de caractères alphanumériques, c'est-à-dire les lettres de l'alphabet en minuscules, en majuscules et les chiffres Romains. Un nom de variable peut contenir aussi le tiret bas. Un nom de variable doit commencer par une lettre ou par le tiret bas.
Attention : certains noms sont réservés : ils correspondent à des mots-clés utilisés par Python. En voici la liste. Nous découvrirons peu à peu leur signification.

Les noms de variable sont sensibles à la casse : les deux noms suivants ne sont pas équivalents ; un conseil : il est bon de choisir des noms de variable explicites : pour des raisons de lisibilité, préférer par exemple nombreDeLivres à nbl.

---
s5

Dans le module suivant, nous allons découvrir les types de base.