Séance 1 : Variables et affectation (Cours)

Pourquoi code-ton ? Vidéo

Introduction

Au collège, vous avez déjà programmé mais en utilisant un langage de programmation *par blocs*. Ce type de langage est très utile pour *apprendre les bases* de l'algorithmique et de la programmation mais ne constitue pas un langage utilisé pour programmer réellement. Pour cela, on utilise des langages dits *textuels* qui nécessitent d'écrire tout le code avec son clavier.

Au lycée, vous allez apprendre l'un de ces langages : le langage **Python**. Vous l'utiliserez dans différentes disciplines : Mathématiques, SNT, Physique-Chimie, spécialité NSI notamment.

Différences collège/lycée

Voici à gauche, un programme écrit dans un langage par blocs et à droite le même programme écrit dans le langage Python.


```
nombre_depart = float(input("Choisir un nombre : "))
nombre_fin = -4 * nombre_depart + 5
if nombre_fin < 0:
    print("Bravo")
else:
    print("Essaie encore")
```

Variables

En informatique, il est indispensable de conserver des informations de natures diverses. Par exemple, votre smartphone enregistre dans sa mémoire votre numéro de téléphone, les numéros de téléphones de vos contacts, vos messages (SMS), vos photos, vos applications, etc. Chacune de ces informations est stockée à un endroit précis dans la mémoire dans ce qu'on appelle une **variable**.

Ainsi, de manière simplifiée, retenez la définition suivante.

Définition

Une **variable** est un espace de stockage de la mémoire (une case mémoire). Chaque variable est caractérisée par son **nom**, son **type** et sa **valeur**.

Au fait, pourquoi dit-on *variable* ?

Tout simplement car les données qu'elle contient peuvent *varier* (au cours du temps) : si un ami change de numéro de téléphone, vous êtes content de pouvoir le modifier.

Types de variables

Le **type** d'une variable est la nature de l'information qu'elle contient. Vous conviendrez que l'heure de votre réveil et votre dernier SMS reçu n'ont pas la même nature : le premier correspond à des nombres et le second à du texte.

Dans un premier temps, voici les trois types de base qui nous intéressent :

- le type **entier** : il désigne les entiers relatifs (positifs ou négatifs). En Python on parle du type **int** (pour *integer* qui signifie « entier » en anglais) ;
- le type **flottant** : il désigne les nombres décimaux (à virgule). En Python on parle du type **float** (pour *floating* qui signifie « flottant » en anglais)
- le type **chaîne de caractères** : il désigne toute suite ordonnée de caractères. En Python on parle du type **str** (pour *string* qui signifie « chaîne » en anglais).

A FAIRE : Activité 1

Noms de variables (en Python)

Chaque variable possède un **nom** qui permet d'identifier l'emplacement mémoire correspondant.

Dans le langage Python, il y a des règles à respecter pour nommmer les variables. Voici celles qui vous concernent :

- **Règle 1** : un nom ne peut contenir que des lettres (a-z, A-Z), des chiffres (0 - 9) et le caractère `_` (nommé *underscore* = « tiret du bas »).
- **Règle 2** : un nom ne peut pas commencer par un chiffre.
- **Règle 3** : les noms sont sensibles à la casse, cela signifie qu'il y a une distinction entre les minuscules et les majuscules : la variable nommée `snt` est différente de la variable `Snt`.
- **(Règle 4)** : il est préférable de toujours choisir un nom de variable représentatif : par exemple, si vous voulez stocker le nom d'une personne dans une variable, il est préférable de l'appeler `nom` plutôt que `x`.
- **(Règle 5)** : il est préférable de ne pas utiliser de caractères accentués dans le nom d'une variable (nous n'entrerons pas dans le pourquoi du comment).

En Python, le symbole *underscore* (c'est-à-dire _) est très souvent utilisé pour marquer une séparation entre plusieurs mots dans un nom : si on veut utiliser une variable qui contiendra un nombre d'élèves de Seconde G, on peut la nommer `nombre_eleves_2G` ou encore `nb_eleves_2G` car c'est plus facile à lire que `nombreeleves2G`

A FAIRE : Activité 2

Valeurs des variables

Chaque variable possède une **valeur** qui est l'information qu'elle porte. Par exemple, la valeur de la variable `nombre_eleves_2G` pourrait être `34`, celle de la variable `note` pourrait être `14.5`, celle de la variable `prenom` pourrait être `"Louane"`.

Affecter une valeur à une variable

En Python, pour définir ou modifier la valeur d'une variable c'est très simple, il suffit d'utiliser le symbole égal : =

Exemples

```
note = 15
note = 15.5
prenom = "Louane"
```

Explications :

- ligne 1 : on affecte la valeur 15 à la variable `note` qui est de type **int** (entier)
- ligne 2 : on affecte une nouvelle valeur, ici 15.5, à la variable `note`, qui devient de type **float** (flottant)
- ligne 3 : on affecte la valeur `"Louane"` à la variable `prenom`, qui est de type **str** (chaîne de caractères).

Important :
- On utilise le point et non pas la virgule pour écrire des nombres décimaux.
- On utilise les guillemets (`""`) pour designer les chaînes de caractères.

Afficher la valeur d'une variable

Pour afficher la valeur d'une variable on utilise la fonction `print()`. Par exemple, pour afficher la valeur de la variable `note` on écrit simplement : `print(note)`. A l'exécution du code, la valeur s'affiche dans la console.

Opérations sur les variables de type nombre (int et float)

Voici les symboles utilisés pour effectuer des opérations avec les variables de type nombre.

Opération	Ecriture en Python
Addition	`a + b`
Soustraction	`a - b`
Multiplication	`a * b`

Division	`a / b`
Elever `a` à la puissance `n`	`a ** n` (double symbole de multiplication)

A FAIRE : Activité 3

A vous de jouer !

Pour écrire du code Python, on utilise un **éditeur de code**. C'est un logiciel qui se décompose en deux fenêtres : la **fenêtre d'édition** dans laquelle on écrit le programme dans le langage Python et la **console** dans laquelle peuvent notamment s'afficher les valeurs des variables souhaitées (si on utilise la fonction `print()`). Il existe énormément d'éditeur et nous utiliserons pour cette séance un éditeur en ligne appelé **BASTHON** (qui signifie **B**ac **À S**able Py**thon**) qui ne nécessite pas d'installation (vous pourrez donc l'utiliser chez vous rien qu'avec une connexion Internet).

Voici un lien permettant d'accéder à Basthon : https://console.basthon.fr/. Vous trouverez également un lien sur la page consacrée à Python de ce site Web.

> Les activités qui suivent n'ont pas de contexte particulier. C'est pourquoi on utilise des noms de variables peu représentatifs. Leur seul but est de faire ancrer les notions abordées dans ce document

A FAIRE : Activités 4, 5, 6, 7 et 8

Compléments : Opérations sur le chaînes de caractères

Pour les chaînes de caractères c'est plus court car il n'y a que deux opérations possibles :

- la *concaténation* qui consiste à mettre bout à bout deux chaînes pour former une unique chaîne : on utilise pour cela le symbole d'addition +.
- la *répétition* qui consiste à ajouter une chaîne un certain nombre de fois à elle-même : on utilise pour cela le symbole de multiplication *.

```
1 chaine1 = "Lou"
2 chaine2 = "ane"
3
4 chaine3 = chaine1 + chaine2  # concaténation
5 print(chaine3)
6
7 chaine4 = 3 * chaine1  # répétition
8 print(chaine4)
```

```
Python 3.8.2 (default, Dec 25 2020
Type "help", "copyright", "credits"
>>> # script executed
Louane
LouLouLou
>>>
```