

Python : les variables:

1. Programmer un ordinateur, c'est quoi ?:

Programmer, c'est créer des programmes (suite d'instructions données à l'ordinateur) ! Un ordinateur sans programme ne sait rien faire. Il existe différents langages qui permettent de programmer un ordinateur, mais le seul directement utilisable par le processeur est le langage machine (suite de 1 et de 0). Aujourd'hui (presque) plus personne ne programme en langage machine (trop compliqué).

Les informaticiens utilisent des instructions (mots souvent en anglais) en lieu et place de la suite de 0 et de 1. Ces instructions, une fois écrites par le programmeur, sont traduites en langage machine. Un programme spécialisé assure cette traduction. Ce système de traduction s'appellera interpréteur ou bien compilateur, suivant la méthode utilisée pour effectuer la traduction.

Il existe 2 grandes familles de langages de programmation :

- Les langages de bas niveau sont très complexes à utiliser, car très éloignés du langage naturel, on dit que ce sont des langages proches de la machine, en contrepartie ils permettent de faire des programmes très rapides à l'exécution. L'assembleur est le langage de bas niveau. Certains "morceaux" de programmes sont écrits en assembleur encore aujourd'hui.
- Les langages de haut niveau sont eux plus "faciles" à utiliser, car plus proches du langage naturel (exemple : si $a=3$ alors $b=c$). Exemples de langages de haut niveau : C, C++ , Java, Python...

En NSI, notre langage de prédilection sera Python. Pour écrire nos programmes, nous utiliserons la sandbox basthon (<https://console.basthon.fr/>).

2. Prise en main de basthon:

Une fois basthon lancé, vous devriez obtenir quelque chose qui ressemble à cela :



basthon se divise en plusieurs fenêtres, deux fenêtres vont principalement nous intéresser : la fenêtre "éditeur" et la fenêtre "console".

Exercice 1: Dans la fenêtre "éditeur", saisissez le programme suivant :

```
0 print('Hello world!')
```

prgm.py

Cliquez sur le "Exécuter" afin d'exécuter le programme qui vient d'être saisi. Vous devez voir le message "hello world !" apparaître dans la console.

3. Notion de variable:

Définition du mot ordinateur d'après "Le Petit Larousse" :

"Machine automatique de traitement de l'information, obéissant à des programmes formés par des suites d'opérations arithmétiques et logiques."

Qui dit "traitement de l'information", dit donc données à manipuler. Un programme "passe" donc son temps à traiter des données. Pour pouvoir traiter ces données, l'ordinateur doit les ranger dans sa mémoire (RAM - Random Access Memory). La RAM se compose de cases dans lesquelles nous allons ranger ces données (une donnée dans une case). Chaque case a une adresse (ce qui permet au processeur de savoir où sont rangées les données).

Alors, qu'est-ce qu'une variable ?

Eh bien, c'est une petite information (une donnée) temporaire que l'on stocke dans une case de la RAM. On dit qu'elle est "variable", car c'est une valeur qui peut changer pendant le déroulement du programme.

Une variable est constituée de 2 choses :

- Elle a une valeur : c'est la donnée qu'elle "stocke" (par exemple le nombre entier 5).
- Elle a un nom : c'est ce qui permet de la reconnaître. Nous n'aurons pas à retenir l'adresse de mémoire, nous allons juste indiquer des noms de variables à la place.

```
0 i = 12
```

prgm.py

Grâce à cette ligne, nous avons défini une variable qui porte le nom `i` et qui "contient" le nombre entier 12. Plus précisément, nous dirons que la variable `i` référence le nombre entier 12.

Exercice 2: Dans la partie "éditeur" de basthon, saisissez le code suivant :

```
0 point_de_vie = 15
```

prgm.py

Après avoir exécuté le programme en cliquant sur le Exécuter, il est possible de connaître la valeur référencée par une variable en utilisant la partie "console" de basthon.

Dans le cas qui nous intéresse ici, tapez `point_de_vie` dans la console. Après avoir appuyé sur la touche "Entrée", vous devriez voir la valeur référencée par la variable `point_de_vie` s'afficher dans la console.

NB: Dans la suite la procédure sera toujours la même :

- Vous utiliserez la partie "éditeur" pour saisir votre programme
- vous utiliserez la partie "console" pour afficher la valeur référencée par une variable

Exercice 3: Écrire un programme dans lequel on attribue la valeur 12 à la variable `point_de_force`. La valeur référencée par la variable `point_de_force` devra ensuite être affichée dans la console.

Nous venons de voir qu'une variable peut référencer un nombre entier, mais elle peut aussi référencer un nombre à virgule :

```
0 i = 5.2
```

`prgm.py`

Prenez bien garde, nous utilisons un point à la place d'une virgule (convention anglo-saxonne).

Une variable peut donc référencer plusieurs types d'entités (pour l'instant nous n'en avons vu que deux, mais nous en verrons d'autres plus loin) : les nombres entiers ("integer" en anglais, abrégé en "int") et les nombres à virgule ("float" en anglais). Il est possible de connaître le type de l'entité référencé par une variable à l'aide de l'instruction "type".

Exercice 4: Testez le programme suivant :

```
0 a = 5.2  
  b = 12
```

`prgm.py`

tapez `type(a)` puis `type(b)` dans la console. Comme vous pouvez le constater, le type de la grandeur référencée par la variable `a` et le type de la grandeur référencée par la variable `b` s'affichent dans la console.