

Python3 - Les variables

Aymc#9525 & Chall3nger#0076

1 Les variables

En programmation, les variables représentent une notion fondamentale. Une variable est, en quelque sorte, une case de la mémoire de l'ordinateur. Cela permet de stocker des données de multiples types tels que des nombres ou encore du texte, nous y viendrons. Au sein d'un programme, une variable est représentée par son nom. C'est à vous de le choisir, vous êtes totalement libre ! Toutefois, il est très vivement conseillé de suivre certaines règles et bonnes pratiques pour faciliter la relecture et la compréhension (voir le cours sur les conventions et bonnes pratiques). Essayons de mettre du sens derrière ces phrases. Pour ceux qui débutent, je vous invite à vous munir de votre console python pour exécuter ces lignes en même temps.

```
1 >>> A = 5
2 >>> A
3 5
4 >>> A + A
5 10
```

Ici, nous avons créé une variable nommée **A**, elle est initialisée à 5 à la ligne 1. En python, il est possible de donner une valeur à une variable grâce au signe "=", cela semble assez instinctif. Ensuite, nous voyons qu'il est possible d'accéder aux données contenues dans **A** en utilisant son nom; on dit qu'on appelle la variable. Ainsi, à la ligne 2, nous appelons la variable **A** et l'ordinateur nous renvoie sa valeur : 5 à la ligne 3. Finalement, la ligne 4 nous montre qu'il est possible d'effectuer des opérations avec les variables. La notion à assimiler c'est que même s'il s'agit d'une lettre, **A** n'est pas du texte, le programme la considère exactement de la même manière qu'un nombre, car justement, la variable contient un nombre.

Nous avons vu les variables qui peuvent contenir des nombres. Maintenant, allons découvrir les chaînes de caractères. On appelle chaîne de caractères, du texte, tout simplement. Par exemple, si vous voulez une variable qui contienne le nom de votre chien, vous allez nécessiter une chaîne de caractère.

```
1 >>> nom = "Milou"
```

Vous remarquerez que le nom de notre gentil toutou est encadré par des guillemets. Il s'agit d'une manière de dire à python que ce qu'on entre n'est pas le nom d'une autre variable mais bien une suite de lettres qu'il faut garder telle quelle.

Il existe finalement un dernier genre de variable : les booléens. Ce nom exotique cache en réalité une notion très simple. Il s'agit d'une variable qui ne peut contenir que deux valeurs : Vrai ou Faux. C'est un peu comme si elle ne pouvait contenir que 1 ou 0. Même si vous ne percevez pas encore leur utilité, vous vous apercevrez qu'elles sont très utiles et, en réalité, centrales en informatique. Pour le moment, retenez qu'il s'agit de variables qui contiennent soit Vrai soit Faux. En python, elles se notent comme ceci :

```
1 >>> variable_vraie = True
2 >>> variable_fausse = False
```

2 Manipulation de variables

Essayons maintenant de jouer un peu avec ces nouvelles notions.

```
1 >>> A = 5
2 >>> B = 12
3 >>> B * A
4 60
5 >>> A = A + B
6 >>> A
7 17
8 >>> 8.354/100
9 0.08353999999999999
```

Que de choses passionnantes viennent de se passer sous nos yeux ! Analysons : la ligne 1 et 2 permettent de définir deux valeurs pour nos variables **A** et **B**, rien de nouveau. La ligne 3 n'est rien d'autre qu'une multiplication, une autre

forme d'opérations entre des variables. De nouvelles choses apparaissent à la ligne 5. Les plus matheux d'entre vous doivent ahurir en voyant cela, c'est mathématiquement faux. Pour autant, comme il a été dit précédemment, le symbole "=" ne traduit pas une égalité mais une assignation de valeur à la variable. Ici, on veut donc donner $A + B$ comme nouvelle valeur pour `A`. Dans notre cas, $A + B$ vaut 17, on assigne donc ce nouveau nombre à la variable `A`. Comme le montre la ligne 6, la variable `A` a bien changé de valeur. Il est ainsi possible de changer la valeur d'une variable au cours d'un programme. Enfin, à la ligne 8, nous avons effectué une division qui va nous amener à rencontrer un nouveau type de nombres : les décimaux, appelés flottants en programmation. Ces nombres peuvent parfois poser problème et nous en avons l'exemple ici. $8,354/100 = 0,08354$: c'est évident ! Mais python ne l'entend pas de cette oreille. Cette erreur provient de la manière dont les nombres flottants sont gérés par python, nous n'entrerons pas dans les détails. Retenez simplement qu'il faut être méfiant avec les flottants.

3 Les différents types de données

De manière général, il existe quatre types de données pouvant être contenus dans des variables:

- Les entiers, integer en anglais, notés `int`
- Les flottants, float en anglais, notés `float`
- Les chaînes de caractères, string en anglais, notées `str`
- Les booléens, boolean en anglais, notés `bool`

Afin de connaître le type d'une variable, il existe un outil intégré dans python : `type`

```
1 >>> type(42)
2 <class 'int'>
3 >>> type(1.3)
4 <class 'float'>
5 >>> type("Bonsoir")
6 <class 'str'>
7 >>> type(True)
8 <class 'bool'>
```

Nous retrouvons ici les notations décrites plus haut, `int`, `float`, `str` et `bool`.

4 Exercices

Les corrections sont disponibles à la page suivante. Essayez de faire les exercices avant de regarder les solutions.

4.1 Créer des variables

Dans votre console python, créez une variable `age` et entrez-y votre age. Appuyez ensuite sur la touche Entrée. Bravo, vous venez de créer votre première variable ! Maintenant on va essayer de l'afficher, il vous suffit d'entrer le nom de la variable et d'appuyer sur Entrée. Une fois cela fait, multipliez la variable par 2 en faisant faire le calcul par python : interdiction de calculer le double au préalable puis de l'entrer manuellement. Finalement, créez une variable `name` contenant votre prénom. Affichez la sur la console. Vous pouvez aussi afficher `name` et `age` en même temps en les séparant par une virgule.

4.2 Manipulation des variables

Commencez par créer une variable `paris` contenant le nombre 2024, c'est à dire l'année où auront lieux les J.O. de Paris. Ensuite, créez une nouvelle variable `in_four_years` valant 4. Finalement, faites une variable `los_angeles` contenant la somme de `paris` et `in_four_years`. Cela correspondra à l'année des J.O. de Los Angeles.

Pour manipuler les variables de type `float`, vous pouvez créer une dernière variable `centuries` qui calcul, en siècles, le temps écoulé entre l'an 0 et l'année des J.O. de Los Angeles.

Aide : Divisez la variable `los_angeles` par un certain nombre, `centuries` sera du type `float`.

4.3 Les différents types de données

À l'aide de la console et de la commande `type`, déterminez le type de 45 puis de la variable `name` de l'exercice n°1.

5 Correction

5.1 Créer des variables

```
1 >>> age = 42
2 >>> age
3 42
4 >>> age * 2
5 84
6 >>> name = "Tintin"
7 >>> name
8 "Tintin"
9 >>> name, age
10 ('Tintin', 42)
```

5.2 Manipulation des variables

```
1 >>> paris = 2024
2 >>> in_four_years = 4
3 >>> los_angeles = paris + in_four_years
4 >>> los_angeles
5 2028
6 >>> centuries = los_angeles / 100
7 20.28
```

Ici nous commençons par créer une variable `paris` avec la valeur 2024, l'année où se déroulent les J.O. de Paris. Ensuite, la variable `in_four_years` qui contient le chiffre 4. Enfin, à la ligne 3, on additionne `paris` et `in_four_years` pour obtenir `los_angeles` qui vaut 2028, soit l'année des J.O. de Los Angeles. Finalement, pour calculer le nombre de siècles écoulés entre l'an 0 et 2028, il faut diviser `los_angeles` par 100, pour obtenir 20,28.

5.3 Les différents types de données

```
1 >>> type(45)
2 <class 'int'>
3 >>> type(name)
4 <class 'str'>
```

45 est considéré comme un `int`. Le type `int` signifie que le nombre est un entier, il n'a pas de valeur décimale. Ces nombres sont très faciles à manier pour les ordinateurs. Ils peuvent être manipulés en effectuant des opérations (additions, multiplications...). Ces opérations peuvent s'effectuer entre `int` et `float` car ces deux types sont des nombres. Finalement, les `str` (chaînes de caractères), c'est à dire du texte sont délimitées par des guillemets. Il existe également des opérations possibles entre elles, comme des additions, nous auront le temps d'y revenir.