

Les variables

i

Il est souvent intéressant de pouvoir en conserver et réutiliser une valeur, par exemple le résultat d'un calcul. C'est le principe des variables : mémoriser une valeur.

•Affectation

En Python, les variables fonctionnent comme des étiquettes. Quand on assigne une valeur à une variable, on pose une étiquette avec un nom sur une valeur.

C'est-à-dire que la valeur existe quelque part en mémoire et qu'on vient lui attacher une étiquette. On peut aisément placer plusieurs étiquettes sur une même valeur, mais aussi retirer une étiquette pour la placer sur une autre valeur.

L'affectation se fait à l'aide du signe =

Affectation

Réaffectation

Double affectation

1

```
a = 1
```

1

```
a = 2
```

1

```
a = b
```



bandeau1

Source : <http://foobarnbaz.com/2012/07/08/understanding-python-variables/>

!

L'utilisation d'une variable avant de l'avoir défini lève une erreur dans l'interpréteur Python

Exemple

L'exécution du code ci-dessous :

```
1 premier = 4 + deuxieme
```

Erreur

```
1 Traceback (most recent call last):
2 File "tmp.py", line 1, in <module>
3     premier = 4 + deuxieme
4 NameError: name 'deuxieme' is not defined
```

• Les types de variables

Il existe plusieurs type de variables. Parmi les plus courants, on retrouve :

Variable	Type	Remarques
<code>a = 2</code>	<code>int</code> : Les nombres entiers relatifs	
<code>a = 3.5</code>	<code>float</code> : Les nombres réels	Le séparateur est le <code>.</code> et non la <code>,</code>
<code>a = "Bonjour"</code>	<code>str</code> : Les chaines de caractères	Entourer par des guillemets (<code>"</code>) ou des apostrophes (<code>'</code>) ou des triples guillemets (<code>"""</code>) ou des triples apostrophes (<code>'''</code>)
<code>a = True</code>	<code>bool</code> : Les booléens True ou False	

i Le résultat est un nombre décimal, mais en Python les nombres décimaux s'écrivent avec un point et non une virgule comme en français.

i Lorsque le nombre de décimales est infini, la console est obligée d'arrondir le résultat.

• Le nom des variables

•• Obligations :

- Ne pas contenir un signe opératoire `+` `-` `*` `/` `%` ni le caractère `#` (Il est réservé aux commentaires)
- Doit commencer par une lettre
- Ne doit pas être un mot réservé Python (cf. le tableau des mots réservés)

•• Préconisations :

- le nom décrit le contenu variable et est constitué d'au moins 3 caractères.
- le nom est entièrement écrit en minuscules et les mots sont séparés par des espaces soulignés (underscores `_`). Exemple : `ma_variable`
- le nom doit permettre de comprendre le rôle de la variable, éviter les caractères spéciaux et accentuer
- il est déconseillé d'utiliser une abréviation en fonction (exemple : `print`)

i Le nom des variables est sensible à la case, la variable `age` est différente de la variable `Age`

•• Mots réservés

<code>and</code>	<code>as</code>	<code>assert</code>	<code>break</code>	<code>class</code>	<code>continue</code>	<code>def</code>	<code>del</code>
<code>elif</code>	<code>else</code>	<code>except</code>	<code>False</code>	<code>finally</code>	<code>for</code>	<code>from</code>	<code>global</code>
<code>if</code>	<code>import</code>	<code>in</code>	<code>is</code>	<code>lambda</code>	<code>none</code>	<code>not</code>	<code>or</code>
<code>pass</code>	<code>raise</code>	<code>return</code>	<code>True</code>	<code>try</code>	<code>while</code>	<code>with</code>	<code>yield</code>

● Opérations arithmétiques

Opérateur	Équivalent
$a + b$	Somme de a et b
$a - b$	Différence de a et b
$a * b$	$a = a * b$
$a ** b$	a puissance b : a^b
a / b	Quotient de a par b
$a // b$	Dividende de a par b
$a \% b$	Reste de la division euclidienne de a par b

● Sucre syntaxique

Comme dans d'autres langages certains opérateurs sont introduits pour raccourcir les processus de réaffectation des variables.

Exemple

Plutôt que d'écrire

```
1 i = i + 1
```

on préférera le raccourci suivant :

```
1 i += 1
```

Opérateur	Équivalent
$a += b$	$a = a + b$
$a -= b$	$a = a - b$
$a *= b$	$a = a * b$
$a **= b$	$a = a ** b$
$a /= b$	$a = a / b$
$a //= b$	$a = a // b$
$a \% = b$	$a = a \% b$

● Exercices

Exercice 1 ★ Affectation

On considère le programme suivant

```
1 >>> a = 3
2 >>> b = 7
3 >>> c = b + a + 2
```

Compléter le tableau d'avancement des variables.

ligne	a	b	c
1			
2			
3			

Exercice 2 ★ Affectation

On considère le programme suivant :

```
1 >>> a = 5
2 >>> b = 2
3 >>> a = b * 3
4 >>> b = a ** 2
5 >>> c = a + b
```

Compléter le tableau d'avancement des variables.

ligne	a	b	c
1			
2			
3			
4			
5			

Exercice 3 ★ Affectation

On considère le programme suivant :

```
1 >>> a = 5
2 >>> a + 2
3 >>> b = a * 3
4 >>> b = b - 6
5 >>> c = a - b
```

Compléter le tableau d'avancement des variables

ligne	a	b	c
1			
2			
3			
4			
5			

Exercice 4 ★ Bien nommer ces variables

Parmi les noms suivants, quels sont ceux qui ne peuvent être utilisés pour nommer des variables en Python. Justifier.

- ma variable
- 1_nombre
- ma_variable
- %V4ri4ble\$
- nombre1
- ma-variable
- maVariable
- ma#variable