

# Affectation des variables

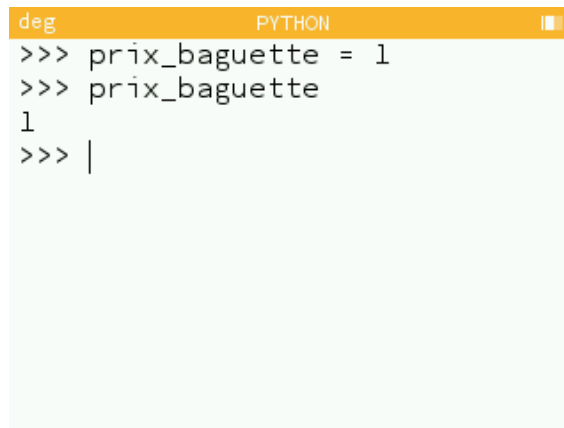
## NUMWORKS

Une variable est un symbole dans lequel il est possible de stocker une valeur. L'affectation de variable est l'action de stocker une valeur dans une variable.

Pour définir le nom d'une variable en Python, vous pouvez utiliser des lettres, minuscules ou majuscules, et des chiffres. Il est aussi possible d'utiliser le caractère underscore : `_`. Cependant, le nom de la variable ne doit pas commencer par un chiffre. Ainsi `variable_1` est accepté par Python. Mais `1ere_variable` ne l'est pas.

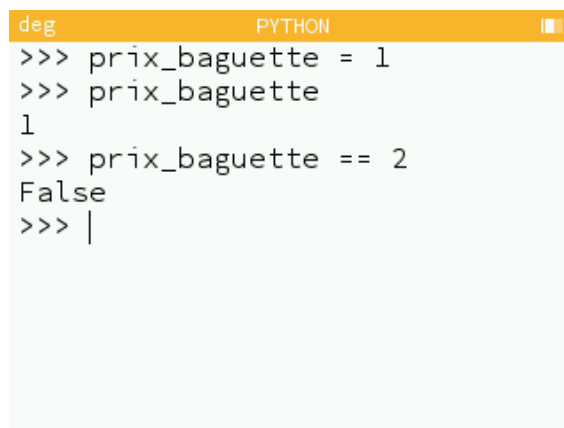
L'affectation de variable se fait avec le symbole `=`. Ainsi, si vous souhaitez stocker la valeur 1 dans `prix_baguette`, vous écrirez l'instruction :

```
prix_baguette = 1
```



```
deg PYTHON
>>> prix_baguette = 1
>>> prix_baguette
1
>>> |
```

Dans la console, vous pouvez demander à Python le contenu d'une variable en écrivant le nom de cette variable et en appuyant sur EXE, comme sur l'exemple ci-contre.



```
deg PYTHON
>>> prix_baguette = 1
>>> prix_baguette
1
>>> prix_baguette == 2
False
>>> |
```

Le symbole `=` étant utilisé pour l'affectation de variable, il n'est plus possible de l'utiliser pour le

test d'égalité. Par exemple pour savoir si `prix_baguette` vaut 2, on devra utiliser un double signe égale `==`. Dans la console on lira le résultat ci-contre.

On comprendra

Est-ce que la variable `prix_baguette` contient la valeur 2?

Non!

### ATTENTION

Il ne faut pas confondre variable mathématique et variable informatique. En mathématiques on peut par exemple avoir une équation :  $x - 3 = 5$ . L'inconnue de cette équation est la variable  $x$ . On lit "x moins 3 égale 5".

En informatique, on peut être amené à écrire des choses comme `x=x+3`.

Cette instruction ne se lit plus comme "x égale x plus 3" mais doit être comprise comme "je stocke x plus 3 dans x". Ou bien "j'augmente x de 3".

### Remarque

Vous pouvez stocker une chaîne de caractères dans une variable. Une chaîne de caractères se délimite avec des guillemets. Par exemple :

```
texte = "Ceci est une chaîne de caractères"
```

Ici nous avons stocké la chaîne de caractères "Ceci est une chaîne de caractères" dans la variable `texte`.

## 1 Exemple

Je peux écrire la suite d'instructions suivante. Quelle sera la réponse de la console après l'instruction `print(a)` ?

```
deg PYTHON
a = 4
a = a-1
a = 2*a
a = a+4
print(a)
```

On regarde la première instruction : "je diminue `a` de 1". La variable contenait 4, elle contient maintenant 3. La deuxième instruction : "je stocke `2*a` dans `a`". La variable `a` contient 3, je stocke donc 6 dans `a`.

La dernière instruction augmente `a` de 4. La variable `a` qui contenait la valeur 6, contient désormais la valeur 10. La console répond donc 10!

## 2 Exercice

Écrire une suite d'instructions qui échange le contenu de deux variables.

On pourra par exemple imaginer une enveloppe bleue contenant un billet de 20 euros et une enveloppe rouge contenant un billet de 50 euros. Le problème revient à échanger le contenu des deux enveloppes de telle sorte qu'au début on ait :

```
>>> bleue
20
>>> rouge
50
```

Et à la fin :

```
>>> bleue
50
>>> rouge
20
```

## 3 Un autre exercice

Écrire une fonction `moyenne` qui prend une liste de valeurs en argument et qui renvoie la moyenne arithmétique de ces valeurs.