

Noms de variables

Complément - niveau basique

Revenons un peu sur les noms de variables autorisés ou non.

Les noms les plus simples sont constitués de lettres. Par exemple:

```
factoriel = 1
```

On peut utiliser aussi les majuscules, mais attention cela définit une variable différente. Ainsi:

```
Factoriel = 100
factoriel == Factoriel
```

Le signe == permet de tester si deux variables ont la même valeur. Si les variables ont la même valeur, le test retournera True et False sinon.

Conventions habituelles

En règle générale on utilise **uniquement des minuscules** pour désigner les variables simples (ainsi d'ailleurs que pour les noms de fonctions); les majuscules sont réservées en principe pour d'autres sortes de variables, comme les noms de classe, que nous verrons ultérieurement.

NB. Il s'agit uniquement d'une convention, ceci n'est pas imposé par le langage lui-même.

Pour des raisons de lisibilité, il est également possible d'utiliser le tiret bas _ dans les noms de variables. On préférera ainsi

```
age_moyen = 75 # oui
```

plutôt que ceci (bien qu'autorisé par le langage)

```
AgeMoyen = 75 # non
```

On peut également utiliser des chiffres dans les noms de variables comme par exemple

```
age_moyen_dept75 = 80
```

avec la restriction toutefois que le premier caractère ne peut pas être un chiffre; cette affectation est donc refusée:

```
[-] 75_age_moyen = 80 # erreur de syntaxe
```

Le tiret bas comme premier caractère

Il est par contre, en théorie au moins, possible de faire commencer un nom de variable par un tiret bas comme premier caractère; toutefois à ce stade nous vous déconseillons d'utiliser cette pratique, qui est réservée à des conventions de nommage bien spécifiques.

```
[-] _autorise_mais_deconseille = 'Voir le PEP 008'
```

Et en tous cas, il est **fortement déconseillé** d'utiliser des noms de la forme `__variable__` qui sont réservés au langage. Nous reviendrons sur ce point dans le futur, mais considérez par exemple cette variable que nous n'avons définie nulle part mais qui pourtant existe bel et bien:

```
[-] __name__ # ne définissez pas vous-même de variables de ce genre
```

Autres caractères

Il n'est **pas possible** d'utiliser d'autres caractères que les caractères alphanumériques et le tiret bas. Notamment le tiret haut - est interprété comme l'opération de soustraction. Attention à cette erreur fréquente:

```
[-] age-moyen = 75 # erreur : en fait python lit 'age - moyen = 75'
```

Pour en savoir plus

Pour les esprits curieux, Guido van Rossum, le fondateur de python, est le co-auteur d'un document qui décrit les conventions de codage à utiliser dans les bibliothèques python standard. Ces règles sont plus restrictives que ce que le langage permet de faire, mais constituent une lecture intéressante si vous projetez d'écrire beaucoup de python.

Voir La section consacrée aux règles de nommage – (en anglais)

(<http://legacy.python.org/dev/peps/pep-0008/#descriptive-naming-styles>)