**Fiche ressource sur les bases du langage Python**

Les nombres

Les nombres entiers sont de type int, et les nombres décimaux sont de type float (les nombres flottants).

La commande type () donne le type d’une expression.

Les opérations

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Opérations | Addition  | Soustraction | Multiplication | Puissance | Division |
| Symboles | + | - | \* | \*\* | / |

Remarque sur la division : la division donne toujours un résultat avec une partie décimal

Comparaison des nombres

|  |  |
| --- | --- |
| Python | Français |
| x == y | x est égal à y |
| x != y | x est différent de y |
| x > y | x est strictement supérieur à y |
| x < y | x est strictement inférieur à y |
| x >= y | x est supérieur ou égal à y |
| x <= y | x est inférieur ou égal à y |

Noms de variable et mots réservés

Les noms des variables sont des noms que vous choisissez : de préférence assez courts, mais aussi le plus explicite possible, de manière à exprimer clairement ce que la variable est censée contenir.

Pour désigner le prix d’un article par exemple, nous pouvons utiliser le mot entier prix ou PRIX mais aussi P. Alors qu’en mathématiques, nous avons l’habitude d’utiliser la lettre x pour nommer une variable.

Quelques règles à respecter :

* Un nom de variable est une suite de lettres (a → z , A → Z) et de chiffres (0 → 9), qui doit toujours commencer par une lettre ;
* Seules les lettres ordinaires sont autorisées. Les lettres accentuées, les cédilles, les espaces, les caractères spéciaux tels que $, #, @, etc. sont interdits, à l’exception du caractère \_ ;
* Les caractères majuscules et minuscules sont distingués.

On ne peut pas utiliser les « 33 mots réservés » ci-dessous :

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| and | assert | break | class | continue | def | del | elif | else | except |
| false | finally | for | from | global | if | import | in | is | lambda |
| none | nonlocal | not | or | pass | raise | return | true | try | while |
| with | yield | as |  |  |  |  |  |  |  |