

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 décimale

Comparaison des nombres

Python	Français
<code>x == y</code>	x est égal à y
<code>x != y</code>	x est différent de y
<code>x > y</code>	x est strictement supérieur à y
<code>x < y</code>	x est strictement inférieur à y
<code>x >= y</code>	x est supérieur ou égal à y
<code>x <= y</code>	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`), qu'il 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 :

<code>and</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>nonlocal</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>	<code>as</code>							