

*** Variables ***

Une variable est une zone de mémoire de l'ordinateur dans laquelle une valeur est stockée. Elle est définie (déclarée) par un nom.

Le simple fait d'écrire le nom d'une variable dans Python suffit à la déclarer. Une variable doit impérativement être initialisée, c'est-à-dire qu'une valeur initiale doit lui être affectée en même temps que la déclaration. L'initialisation s'effectue en insérant un opérateur d'affectation (=) directement à droite du nom de la variable. La valeur à affecter est écrite directement à droite de l'opérateur d'affectation. Des espaces peuvent séparer le nom de la variable, l'opérateur d'affectation et la valeur à affecter.

Pour nommer une variable, sont autorisés : les minuscules, les majuscules, les chiffres et le caractère `_`. Sont strictement défendus : les espaces, un chiffre au début d'un nom, les noms réservés.

Les noms des fonctions et des type utilisés par Python sont des noms réservés. S'ils sont utilisés dans un code pour nommer des variables, cela entraînera des erreurs à l'exécution du code.

Si une erreur intervient lors de l'écriture d'une variable (on a tapé 'petot' au lieu de 'petit' qu'on a déclaré plus haut dans son code), Python affiche un message d'erreur et si une variable récemment utilisée se rapproche de l'écriture erronée Python propose ce nom de variable.

Il vaut mieux utiliser des noms explicites pour ses variables plutôt que des lettres uniques, sauf dans quelques cas.

Python est sensible à la casse.

- ****decla.py**** : déclarer une variable

Il est possible d'affecter une variable à une variable ($x = y$) et d'écraser une valeur initialement ou précédemment affectée. Pour écraser la valeur d'une variable, il suffit d'affecter la nouvelle valeur par l'opérateur d'affectation.

- ****écras.py**** : affecter une variable à une variable et écraser une variable

Les types de variables les plus fréquents sont les entiers (integer), les nombres décimaux (float) et du texte ou plus précisément des chaînes de caractères (string). Python interprète directement le type d'une variable par la valeur qui lui est affectée.

Les décimales d'un nombre float sont séparées par un point (.) et non pas par une virgule. Pour faire comprendre à Python que du texte doit être affecté à une variable, il faut l'encadrer par de guillemets (""") ou deux apostrophes (").

Il existe aussi des variables de type booléen. Ces variables n'ont que deux valeurs possibles, vrai ou faux (True ou False dans Python). Ces valeurs (True et False) sont des noms réservés par Python.

Pour connaître le type d'une variable, inscrire la variable entre les parenthèses de la fonction `type()`.

- ****type.py**** : afficher le type d'une variable

L'écriture (notation) scientifique dans Python s'effectue à l'aide de la lettre 'e'. 10.000.000, s'écrit dans Python 1e7. De cette notation Python calcule : 1×10^7 . 7.000.000 s'écrit dans Python 7e6. Python calcule : 7×10^6 .

Le résultat d'une notation scientifique dans Python est toujours un nombre de type float.

Pour mieux visualiser les nombres à plusieurs chiffres, Python autorise de les séparer par un _ ou plusieurs.

Le fichier Nota_Scient.pdf explique la notation scientifique.

- ****notaScien.py**** : afficher des nombres en utilisant la notation scientifique

Les calculs dans Python sont effectués à l'aide des quatre opérations arithmétiques (+, -, /, *). Python respecte la règle de priorité en mathématique entre les opérandes et tient compte de la priorité des opérations entre parenthèses.

Une opération faisant intervenir un nombre int et float donne un résultat de type float.

Pour élever un nombre ou une variable à une puissance, utiliser ** entre le nombre à élever et le degré de la puissance. Par exemple, x^{**3} (x au cube).

Pour obtenir le quotient d'une division, utiliser // au lieu de /. Pour obtenir son reste, %.

Il est possible d'incrémenter une variable en lui ajoutant une valeur au lieu de l'écraser.

- ****arith.py**** : opérations arithmétiques avec parenthèses
- ****puiss.py**** : opérations sur les puissances
- ****modulo.py**** : opérations sur les quotiens et les restes
- ****increm.py**** : incrémenter une variable

Python permet de concaténer des chaînes de caractères à l'aide de l'opérateur d'addition. Pour répéter ou dupliquer des chaînes de caractères, utiliser l'opérateur de multiplication.

Il n'est pas prévu dans Python de dupliquer du texte par un nombre de type float, ni de concaténer du texte avec un nombre.

- ****concat.py**** : opérations sur les chaînes de caractères

Un type de valeur (str, int ou float) dans Python peut être converti en un autre de ces trois types. Il n'est bien sûr pas possible de convertir l'un de ces trois types en type booléen (bool).

- ****conv.py**** : conversions de types

Les fonctions sont reconnaissables aux parenthèses qui suivent directement un nom. Ces parenthèses servent à introduire un argument à passer dans la fonction.

Deux fonctions très utiles, mis à part celles déjà vues (`print()`, `input()`, `type()`) sont `max()` et `min()`.

Les fonctions `min()` et `max()` contiennent toujours plusieurs arguments, au minimum deux. Chaque argument est séparé par une virgule.

- **`**maxMin.py**`** : trouver le minimum et le maximum de deux nombres