

Opérations arithmétiques	Python
Ajouter $a$ et $b$ .	$a + b$
Soustraire $b$ de $a$ .	$a - b$
Multiplier $a$ par $b$ .	$a * b$
Diviser $a$ par $b$ .	$a / b$
Dans la division euclidienne de $a$ par $b$ : - calculer le quotient (entier) - calculer le reste.	$a // b$ $a \% b$
Élever $a$ à la puissance $n$ .	$a ** n$
Calculer $4 \times 10^3$ .	$4e3$

Pour tester	Python
$a$ est inférieur à $b$	$a < b$
$a$ est inférieur ou égal à $b$	$a <= b$
$a$ est supérieur à $b$	$a > b$
$a$ est supérieur ou égal à $b$	$a >= b$
$a$ est égal à $b$	$a == b$
$a$ est différent de $b$	$a != b$
$a$ est dans la liste $L$	$a \text{ in } L$
$a$ est dans $\text{texte}$	$a \text{ in } \text{texte}$
ET/OU	and / or

Variables	Python	Algorithmme
Affecter à $a$ le nombre 5 ( $a$ reçoit 5).	$a = 5$	$a \leftarrow 5$
Affecter à $a$ le contenu de la variable $b$ ( $a$ reçoit $b$ ).	$a = b$	$a \leftarrow b$
Affecter à $a$ la chaîne de caractères « blabla ».	$a = \text{"blabla"}$	$a \leftarrow \text{"blabla"}$
Afficher le contenu de la variable $a$ .	$\text{print}(a)$	Afficher $a$
Afficher le type de la variable $a$ .	$\text{type}(a)$	

Fonction – Structure conditionnelle – Boucles	Python	Algorithmme
Créer une fonction avec comme arguments (ou paramètres) $a$ et $b$ et renvoyant un résultat nommé $rep$ .	<pre>def nom(a, b):     instructions     return(rep)</pre>	fonction $nom(n, m)$ instructions renvoyer( $rep$ )
Exécuter en boucle des instructions Tant que $condition$ est vérifiée.	<pre>while condition:     instructions</pre>	Tant que $condition$ instructions Fin Tant que
Exécuter en boucle des instructions pour une variable $k$ allant de $n$ à $m$ (voir aussi page 12).	<pre>for k in range(n, m+1):     instructions</pre>	Pour $k$ allant de $n$ à $m$ instructions Fin Pour
Exécuter des instructions selon une condition.	<pre>if condition:     instructions1 else:     instructions2</pre>	Si $condition$ Alors instructions1 Sinon instructions2 Fin Si

Des listes et des ensembles	
Créer les entiers de 0 à $n - 1$ ( $n$ entier naturel non nul).	$\text{range}(n)$
Créer les entiers de $n$ à $m$ ( $n$ et $m$ entiers, $n \leq m$ ).	$\text{range}(n, m+1)$
Créer les entiers de $a$ à $b$ avec un pas $p$ ( $a \leq b$ ).	$\text{range}(a, b+1, p)$
Affecter à $L$ la liste $[1, 2, -3, 10]$ / Affecter à $L$ la liste vide.	$L = [1, 2, -3, 10] / L = []$
Affecter à $L$ la liste des carrés des entiers de 0 à 50.	$L = [t**2 \text{ for } t \text{ in } \text{range}(51)]$
Ajouter l'élément $x$ à la liste $L$ .	$L.append(x)$
Donner le nombre d'éléments de la liste $L$ / leur somme.	$\text{len}(L) / \text{sum}(L)$
Rechercher l'indice de l'élément $x$ de la liste $L$ .	$L.index(x)$
Obtenir l'élément d'indice $n$ de la liste $L$ .	$L[n]$
Supprimer l'élément d'indice $n$ de la liste $L$ .	$\text{del } L[n]$
Parcourir la liste des éléments de $L$ .	$\text{for } x \text{ in } L:$
Modifier la liste $L$ : - en triant ses éléments dans l'ordre croissant - en triant ses éléments dans l'ordre décroissant.	$L.sort()$ $L.sort(\text{reverse} = \text{True})$



# DES BIBLIOTHÈQUES (OU MODULES) EN PYTHON

Certaines fonctions sont accessibles directement en Python, mais beaucoup d'autres sont rangées dans des bibliothèques (ou modules) et il faut les importer avant de les utiliser.

**EXEMPLE** Pour affecter à la variable *h* la racine carrée de 20, l'instruction à exécuter est `h = sqrt(20)`  
La fonction `sqrt` (`sqrt` = square root) est rangée dans la bibliothèque `math`.

Trois façons d'importer une bibliothèque (ou module) et d'utiliser une de ses fonctions.

① 

```
from math import*  
h = sqrt(20)
```

On importe toutes les fonctions de la bibliothèque `math`.

② 


```
import math  
h = math.sqrt(20)
```

Le nom de la bibliothèque doit précéder le nom de la fonction utilisée

③ 

```
import math as m  
h = m.sqrt(20)
```

Comme la méthode ② mais on a donné un nouveau nom (ou alias), plus court, à la bibliothèque

Fonctions et constantes mathématiques	
Renvoyer une valeur arrondie de <i>a</i> avec <i>n</i> décimales.	<code>round(a, n)</code>
Renvoyer le plus petit entre <i>a</i> et <i>b</i> .	<code>min(a, b)</code>
Renvoyer le plus grand entre <i>a</i> et <i>b</i> .	<code>max(a, b)</code>
Prendre la valeur absolue du nombre <i>a</i> .	<code>abs(a)</code>
Des outils de la bibliothèque <code>math</code> importée par <code>from import math*</code>	
Prendre la racine carrée de <i>a</i> .	<code>sqrt(a)</code>
Prendre la partie entière de <i>a</i> .	<code>floor(a)</code>
Utiliser $\pi$ dans un calcul (valeur approchée).	<code>pi</code>
Calculer <code>cos(angle)</code> où <i>angle</i> est en degrés.	<code>cos(radians(angle))</code>
Calculer l'angle en degrés qui a pour sinus <i>a</i> .	<code>degrees(asin(a))</code>
Bibliothèque <code>random</code> importée par <code>import random</code>	
Renvoyer un nombre décimal pseudo aléatoire entre 0 et 1.	<code>random.random()</code>
Renvoyer un nombre entier pseudo aléatoire compris entre les deux entiers <i>a</i> et <i>b</i> inclus.	<code>random.randint(a, b)</code>
Renvoyer une liste formée d'un nombre <i>nb</i> d'éléments : – pris au hasard avec remise dans la liste <i>L</i> – pris au hasard sans remise dans la liste <i>L</i> .	<code>random.choices(L, k = nb)</code> <code>random.sample(L, nb)</code>
Bibliothèque <code>statistics</code> importée par <code>import statistics as sta</code>	
Calculer la moyenne d'une liste <i>L</i> .	<code>sta.mean(L)</code>
Calculer l'écart type d'une liste <i>L</i> (population standard deviation en anglais).	<code>sta.pstdev(L)</code>
Calculer la médiane d'une liste <i>L</i> .	<code>sta.median(L)</code>
Bibliothèque <code>pandas</code> importé par <code>import pandas</code> <span style="float: right;">compléments sur </span>	
Installer <code>pandas</code> à partir d'Anaconda : saisir dans la console l'une des instructions ci-contre.	<code>conda install pandas</code> <code>pip install pandas</code>
Importer le module <code>pandas</code> .	<code>import pandas</code>
Lire le fichier <code>nom.csv</code> figurant dans le même dossier que le programme Python et affecter les données à un « dataframe » (table de données dont chaque colonne correspond à une variable statistique) nommé ici <i>df</i> . Les tableaux de données manipulés par <code>pandas</code> sont des dataframe.	<code>df = pandas.read_csv('nom.csv', sep = ';')</code> <code>sep = ','</code> si le séparateur du fichier csv est une virgule et <code>sep = '\t'</code> si c'est une tabulation. Ajouter l'argument <code>encoding = 'latin-1'</code> si le fichier csv comporte des accents, et l'argument <code>decimal = ','</code> si les nombres décimaux du fichier csv sont séparés par une virgule.