

Variables	listes, chaînes de caractères, tuples
<p align="center">Types de variables</p> <p>int : nombre entier 10 0 -45 0b1011 float : nombre à virgule flottante 1.5 3.4e4 bool : booléen True False str : chaîne de caractères "chaîne 1" "un\ndeux" list : liste de valeurs [1,2,3] ["a","bc",4] tuple : liste non modifiable (1,2,3) 3,"test",5</p> <p align="center">Affectation : donner un nom à une valeur</p> <p>x = 3.2+4*cos(z) a,b = b,a a = b = c = 0 calcul échange a et b même valeur a,b,c = 1,4,5 x="test" a=1 b=4 c=5 chaîne de caractères</p>	<p>Accès à un élément unique</p> <p>L = [3, 5, 7, 2, 8] L[0] L[1] L[2] L[3] L[4] L[-5] L[-4] L[-3] L[-2] L[-1]</p> <p>Nombre d'éléments len(L) → 5</p> <p>Modification d'un élément Uniquement pour les listes, pas pour les tuples ni chaînes. L[3] = 5 affectation</p> <p align="center">Ajouter supprimer</p> <p>del(L[3]) suppression d'un élément L.append(5) ajouter en fin de liste L=L1+L1 concaténation</p> <p align="center">Accès à des sous-listes (slice)</p> <p>L[a:b:c] début fin pas</p> <p>L[0:2] → [3, 5, 7] L[2:] → [7, 2, 8] L[:3] → [3, 5, 7, 2] L[-3:-1] → [7,2,8] L[:2] → [3, 7, 8] L[0:2:-1] → [7, 5, 3]</p>

Python

Boucles

for	while
<p>Exécuter des instructions un nombre de fois prédéterminé</p> <p>for i in range(n): instructions exécutées n fois i prend les valeurs 0, 1, 2, ..., n-1</p> <p>exemple S=0 for i in range(100): S = S+i+1 print(S) affiche la valeur de : $S = \sum_{i=0}^{100} i$</p> <p>Exécuter des instructions pour chaque valeur d'une liste</p> <p>for e in L: instructions pour chaque valeur e de la liste L</p> <p>exemple S = 0 for v in [1,4,6,8]: S = S+i print(S) affiche la valeur de : S = 1+4+6+8</p>	<p>Exécuter des instructions tant qu'une condition logique est vraie</p> <p>while <condition logique>: instructions exécutées tant que la condition est vraie</p> <p>exemple i=S=0 while i<100: i = i+1 S = S+i print(S) affiche la valeur de : $S = \sum_{i=0}^{100} i$</p> <p>Faire varier la variable de la condition sinon boucle infinie</p> <p align="center">contrôle</p> <p>break → Sortie immédiate de la boucle continue → Passer à l'itération suivante</p>

Maths

Opérateurs
<p>+, -, *, / addition, soustraction, multiplication, division</p> <p>** puissance // division entière % modulo ex : 2**3 → 8 ex : 7//3 → 2 ex : 7%3 → 1</p> <p align="center">Fonctions mathématiques</p> <p>from math import * Importer la bibliothèque math</p> <p>sin(pi/4) → 0.707... fonctions trigo (cos, tan, asin, atan, ...) sqrt(81) → 9.0 √ abs(-3.2) → 3.2 log(e**2) → 2.0 log10(1000) → 3.0 ln log</p>

Conditionnelle

<p>if<condition logique>: Instructions exécutées si la condition est vraie</p> <p>elif<autre condition>: Instructions exécutées si la seconde condition est vraie (si la première est fausse)</p> <p>elif<autre condition>: ...</p> <p>else : Instructions exécutées si toutes les conditions sont fausses</p>	<p align="center">exemple</p> <p>if age<18 : e = "enfant" elif age>65 : e = "retraité" else : e = "actif"</p> <p align="center">Facultatifs</p>
--	--

Fichiers

<p>Stocker ou lire des données du disque dur</p> <p>f = open("c:\\docs\\fichier.txt", "w")</p> <p>variable pour manipuler le fichier nom du fichier (chemin d'accès) mode d'ouverture 'r' : lecture (read) 'w' : écriture (write) 'a' : ajout (append)</p> <p align="center">Lecture</p> <p>t = f.read(n) lit n caractères du fichier si n est absent, lit jusqu'à la fin.</p> <p>l = f.readline() lit une ligne du fichier</p> <p>ls = f.readlines() lit toutes les lignes du fichier dans une liste</p> <p align="center">Écriture</p> <p>f.write("Salut") écrit une chaîne de caractères dans le fichier</p> <p>f.close() penser à fermer le fichier sinon perte de données !</p>
--

Affichage

<p>print("v = ", 5, x+y)</p> <p>chaîne de caractères valeur expression</p> <p>les différents paramètres sont séparés par un espace à la fin, print ajoute un retour à la ligne.</p> <p>caractères spéciaux :</p> <p>'\n' : retour à la ligne '\t' : tabulation '\'' : apostrophe simple</p>
--

Fonctions

Définition
<p>nom de la fonction paramètres</p> <p>def mafonction(a,b,c): """documentation"""</p> <p><instructions></p> <p>return res</p> <p>valeur renvoyée par la fonction traitement des paramètres, calculs, ...</p> <p align="center">Appel</p> <p>r = mafonction(3, "abc", i)</p> <p>stockage de la valeur renvoyée les arguments donnent les valeurs des paramètres</p>