Variables

Types de variables

int: nombre entier 10 0 -45 0b1011 **float**: nombre à virgule flottante 1.5 3.4e4

bool : booléen True False

str : chaîne de caractères "chaine 1" "un\ndeux"

list: liste de valeurs [1,2,3] ["a","bc",4] **tuple**: liste non modifiable (1,2,3) 3,"test",5

Affectation : donner un nom à une valeur

$$x = 3.2 + 4 + \cos(z)$$

$$a,b,c = 1,4,5$$

 $a=1$ $b=4$ $c=5$

listes, chaînes de caractères, tuples

Accès à un élément unique

$$L = [3, 5, 7, 2, 8]$$

Nombre d'éléments

$$len(L) \rightarrow 5$$

Modification d'un élément

Uniquement pours les listes, pas pour les tuples ni chaînes.

$$L[3] = 5$$
 affectation

Ajouter supprimer

del(L[3]) suppression d'un élément L.append(5) ajouter en fin de liste L=L1+L1 concaténation

Accès à des sous-listes (slice)

début fin pas

$$L[0:2] \rightarrow [3, 5, 7]$$

 $L[2:] \rightarrow [7, 2, 8]$
 $L[:3] \rightarrow [3, 5, 7, 2]$

$$L[:3] \longrightarrow [3, 5, 7,$$

$$L[-3:-1] \longrightarrow [7,2,8]$$

$$L[::2] \rightarrow [3, 7, 8]$$

 $L[0:2:-1] \rightarrow [7, 5, 3]$

$$L[0:2:-1] \rightarrow [7, 5, 3]$$

thon

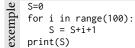
Boucles

Exécuter des instructions un nombre de fois prédéterminé

for i in range(n):

instructions exécutées n fois

i prend les valeurs $0, 1, 2, \ldots, n-1$

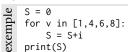


affiche la valeur de :

Exécuter des instructions pour chaque valeur d'une liste

for e in L:

instructions pour chaque valeur e de la liste L



affiche la valeur de : S = 1+4+6+8

while

Exécuter des instructions tant qu'une condition logique est vraie

while <condition logique>:

instructions exécutées tant que la condition est vraie



Faire varier la variable de la condition sinon boucle infinie

contrôle

break

Sortie immédiate de la boucle

affiche la

valeur de :

 $S = \sum_{i} i$

Passer à l'itération continue → suivante

Maths

Opérateurs

addition, soustraction, multiplication, division

** puissance ex: 2**3 → 8

// division entière $ex: 7//3 \rightarrow 2$

% modulo ex: 7%3 → 1

Fonctions mathématiques

from math import * Importer la bibliothèque math

 $sin(pi/4) \longrightarrow 0.707...$ fonctions urgo (cos, tan, asin, atan, ...) $sqrt(81) \rightarrow 9.0 \sqrt{}$ $abs(-3.2) \rightarrow 3.2$

 $log(e**2) \rightarrow 2.0$

 $log10(1000) \longrightarrow 3.0$

Conditionnelle

if<condition logique>:

Instructions exécutées si la condition est vraie

elif<autre condition>:

Instructions exécutées si la seconde condition est vraie (si la première est fausse)

elif<autre condition>:

--else :

> Instructions exécutées si toutes les conditions sont fausses

— exemple —

if age<18 :

e = "enfant"

elif age>65 : e = "retraité"

else :

e = "actif"

Fichiers

Stocker ou lire des données du disque dur

f = open("c:\\docs\\fichier.txt", "w")

{Lecture}

variable pour

t = f.read(n)

lit n caractères du fichier si n est absent, lit jusqu'à la fin.

1 = f.readline() lit une ligne du fichier

ls = f.readlines() ltoutes les lignes du fichier dans une liste

nom du fichier manipuler le fichier (chemin d'accès)

mode d'ouverture 'r': lecture (read)
'w': écriture (write) 'a': ajout (append)

Écriture

f.write("Salut")

écrit une chaîne de caractères dans le fichier

f.close()

penser à fermer le fichier sinon perte de données!

Affichage

print("v = ", 5, x+y)

chaine de caractères

valeur expression

les différents paramètres sont séparés par un espace

à la fin, print ajoute un retour à la ligne.

caractères spéciaux :

'\n': retour à la ligne '\t': tabulation

'\'': apostrophe simple

Fonctions

nom de la paramètres def mafonction(a,b,c): """documentation""" <instructions> return res traitement des valeur renvoyée paramètres,

Définition

par la fonction Appel

"abc", i)

calculs....

r = mafonction(3,

stockage de la valeur renvoyée les arguments donnent les valeurs des paramètres