

Mémento Python



Opérations

Affecter la valeur b à la variable a (la variable a reçoit la valeur b).	<code>a = b</code>
Multiplier a par b.	<code>a * b</code>
Diviser a par b.	<code>a / b</code>
Élever a à la puissance n.	<code>a ** n</code>
Calculer le quotient de la division euclidienne de a par b.	<code>a // b</code>
Calculer le reste de la division euclidienne de a par b.	<code>a % b</code>

Boucles, tests et fonction

Exécuter en boucle des instructions tant qu'une condition est vérifiée.	<pre>while condition: instructions</pre>
Exécuter en boucle des instructions pour une variable k allant de 0 à n - 1.	<pre>for k in range(n): instructions</pre>
Exécuter en boucle des instructions pour une variable k allant de n à m - 1.	<pre>for k in range(n, m): instructions</pre>
Exécuter des instructions selon une condition.	<pre>if condition: instructions1 else: instructions2</pre>
Exécuter des instructions selon deux conditions (elif remplace else if).	<pre>if condition1: instructions1 elif condition2: instructions2 else: instructions3</pre>
Tester si a égale b.	<code>a == b</code>
Tester si a est différent de b	<code>a != b</code>
Tester si a est strictement inférieur à b	<code>a < b</code>
Tester si a est supérieur ou égal à b.	<code>a >= b</code>
Créer une fonction nom avec comme paramètres n et m et renvoyant un résultat.	<pre>def nom(n, m): instructions return resultat</pre>

Des fonctions utiles

Renvoyer une valeur arrondie de a avec n chiffres après la virgule.	<code>round(a, n)</code>
Renvoyer le plus petit entre a et b.	<code>min(a, b)</code>
Renvoyer le plus grand entre a et b.	<code>max(a, b)</code>

Listes	
Créer la liste des entiers de 0 à $n - 1$.	<code>range(n)</code>
Créer la liste des entiers de n à $m - 1$.	<code>range(n, m)</code>
Ajouter l'élément x à la liste L .	<code>L.append(x)</code>
Donner le nombre d'éléments de la liste L .	<code>len(L)</code>
Rechercher l'indice de l'élément x de la liste L .	<code>L.index(x)</code>
Obtenir l'élément d'indice n de la liste L .	<code>L[n]</code>
Supprimer l'élément d'indice n de la liste L .	<code>del L[n]</code>
Parcourir les éléments x d'une liste L .	<code>for x in L:</code>
Modifier la liste L pour trier ses éléments dans l'ordre croissant.	<code>L.sort()</code>
Modifier la liste L pour trier ses éléments dans l'ordre décroissant.	<code>L.sort(reverse = True)</code>

Bibliothèque math	
Importer la bibliothèque math.	<code>import math</code>
Prendre la racine carrée de a .	<code>math.sqrt(a)</code>
Importer la valeur de π .	<code>math.pi</code>
Renvoyer le cosinus, le sinus ou la tangente de x (en radian).	<code>math.cos(x)</code> <code>math.sin(x)</code> <code>math.tan(x)</code>

Bibliothèque random	
Importer la bibliothèque random.	<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 a et b inclus.	<code>random.randint(a, b)</code>

Bibliothèques numpy et matplotlib.pyplot	
Importer la bibliothèque numpy avec l'alias <code>np</code> .	<code>import numpy as np</code>
Importer la bibliothèque matplotlib.pyplot avec l'alias <code>plt</code> .	<code>import matplotlib.pyplot as plt</code>
Définir la liste des valeurs de x de l'intervalle $[a, b]$ et le nombre n de points calculés pour le tracer d'une courbe.	<code>x = np.linspace(a, b, n)</code>
Définir la liste des valeurs de x de l'intervalle $[a, b]$ avec un pas p non nécessairement entier pour le tracé d'une courbe.	<code>X = np.arange(a, b, p)</code>
Définir les dimensions des axes entre a et b pour les abscisses et entre c et d pour les ordonnées.	<code>plt.axis([a, b, c, d])</code>
Tracer une ligne reliant les points de coordonnées x et y où x et y sont deux listes (courbe d'une fonction).	<code>plt.plot(x, y)</code>
Tracer les points de coordonnées x et y où x et y sont deux listes (représentation d'une suite).	<code>plt.scatter(x, y)</code>
Afficher un quadrillage.	<code>plt.grid()</code>
Ouvrir une fenêtre et afficher un graphique.	<code>plt.show()</code>
Tracer un histogramme des données de la liste L , regroupées en n classes.	<code>plt.hist(L, bins = n)</code>
Tracer un histogramme des données de la liste L , regroupées en n classes, avec une aire totale valant 1 (histogramme normalisé).	<code>plt.hist(L, bins = n, normed = True)</code>
Tracer un diagramme en barres (ou bâtons) dont les abscisses correspondent à la liste x et les hauteurs à la liste y .	<code>plt.bar(x, y)</code>
Tracer un diagramme circulaire (camembert) correspondant aux valeurs de la liste L et aux étiquettes de la liste E .	<code>plt.pie(L, labels = E)</code>

Bibliothèque statistics	
Importer la bibliothèque statistics avec l'alias sta.	<code>import statistics as sta</code>
Calculer la moyenne d'une liste L.	<code>sta.mean(L)</code>
Calculer l'écart type d'une liste L (population standard deviation en anglais).	<code>sta.pstdev(L)</code>
Calculer la médiane d'une liste L.	<code>sta.median(L)</code>

Bibliothèque pandas	
Installation de pandas et lecture d'un fichier csv	
Installer pandas à partir d'Anaconda : saisir dans la console l'une des instructions ci-contre.	<pre>conda install pandas pip install pandas</pre>
Importer le module pandas	<code>import pandas</code>
Lire le fichier nom.csv 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 df. Les tableaux de données manipulés par pandas sont des dataframe.	<pre>df = pandas.read_csv('nom.csv', sep = ';') sep = ',' si le séparateur du fichier csv est une virgule. sep = '\t' si le séparateur du fichier csv est une tabulation. Ajouter l'argument encoding = 'latin-1' si le fichier csv comporte des accents. Ajouter l'argument decimal = ',' si les nombres décimaux du fichier csv sont séparés par une virgule.</pre>
Opérations de base sur un dataframe pandas et ses colonnes	
Dimension du dataframe df : nombre de lignes et de colonnes (c'est-à-dire de variables).	<code>df.shape</code>
(Nom) et type des variables (ou colonnes).	<code>df.dtypes</code>
Afficher les 5 premières lignes de df.	<code>df.head(5)</code>
Afficher les 5 dernières lignes de df.	<code>df.tail(5)</code>
Afficher des indicateurs statistiques pour df.	<code>df.describe()</code>
Afficher des indicateurs statistiques pour la variable (colonne) nommée C.	<code>df['C'].describe()</code>
Calculer la moyenne de la variable C.	<code>df['C'].mean()</code>
Calculer la médiane de la variable C.	<code>df['C'].median()</code>
Calculer la somme de la variable C.	<code>df['C'].sum()</code>
Compter le nombre de valeurs de la variable quantitative C.	<code>df['C'].count()</code>
Compter le nombre de valeurs de la variable qualitative C.	<code>df['C'].value_counts()</code>
Trier le dataframe df par ordre croissant selon la variable C.	<code>df.sort_values(by = 'C')</code>
Trier le dataframe df par ordre décroissant selon la variable C.	<code>df.sort_values(by = 'C', ascending = False)</code>
Extraction d'un sous-tableau sous condition, tableau croisé avec pandas	
Créer un dataframe df2 correspondant aux lignes de df où la variable C est non nulle (trois écritures possibles).	<pre>df2 = df.query('C != 0') df2 = df[df.C != 0] df2 = df[df['C'] != 0]</pre>
Créer un dataframe df2 correspondant aux lignes de df où la variable A est supérieure à 2 et la variable B inférieure à 6 (pour « ou » utiliser).	<code>df2 = df[(df.A > 2) & (df.B < 6)]</code>
Créer un dataframe df2 échantillon aléatoire sans remise de taille 100 de df.	<pre>import random df2 = df.sample(n = 100)</pre>
Regroupement des données selon les valeurs de la variable C.	<code>df.groupby('C')</code>
Créer un tableau croisé en effectifs des variables C1 et C2 d'un dataframe df.	<code>pandas.crosstab(df['C1'], df['C2'], margins = True)</code>