

I'm not a bot



Un tableau est l'une des structures de données fondamentales en Python. Il stocke les éléments dans un emplacement mémoire contigu et chaque élément est accessible à l'aide de son index respectif. En Python, la bibliothèque numpy est utilisée pour implémenter des structures de données de tableau. Les listes sont également une autre structure de données qui imite un tableau en Python. Dans ce didacticiel, nous aborderons différentes méthodes pour remplir un tableau avec des nombres aléatoires en Python. Nous pouvons créer une gamme de formes souhaitées. Parfois, nous pouvons avoir besoin de tableaux factices pour certains calculs. Nous pouvons créer un tableau rempli de telles valeurs factices puisque la bibliothèque numpy implémente également un module random pour générer des nombres aléatoires. Façons de remplir un tableau avec des nombres aléatoires en Python Voyons maintenant comment remplir un tableau avec des nombres aléatoires en Python. Utilisation de la fonction numpy.random.randint() pour remplir un tableau avec des nombres aléatoires en Python Comme indiqué précédemment, la bibliothèque numpy possède un module random qui peut aider à générer des nombres aléatoires pour les tableaux numpy. La fonction randint() peut être utilisée pour générer un tableau de la taille requise et remplir le tableau avec des nombres aléatoires en Python. Nous devons rappeler les trois paramètres principaux. Les deux premiers paramètres sont les valeurs low et high. La fonction sélectionnera un entier aléatoire entre cette plage. Le troisième paramètre est le paramètre shape qui spécifie la forme du tableau final. Voir l'exemple suivant. Utilisation de la fonction numpy.random.randint() :import numpy as nparr = np.random.randint(0,10,10)print(arr) Sortir :{3 8 1 0 4 2 4 7 0 3} Dans l'exemple ci-dessus, nous générons des nombres aléatoires entre 0 et 10 et les remplissons dans un tableau unidimensionnel de longueur dix. Utilisation de la fonction numpy.random.Generator.integers() pour remplir un tableau avec des nombres aléatoires en Python Les numpy.random.Generators offrent une nouvelle façon de générer et de travailler avec des nombres aléatoires. Il utilise un BitGenerator supplémentaire pour générer ces bits aléatoires et gérer leurs tats. Pour lancer un nouveau Générateur, nous utilisons le constructeur numpy.random.default_rng(). Après cela, nous pouvons utiliser la fonction numpy.random.Generator.integers() pour générer des nombres aléatoires et remplir le tableau avec des nombres aléatoires en Python. Nous devons spécifier les valeurs low, high et shape dans cette fonction comme nous l'avons fait dans la méthode précédente. Par exemple, Utilisation de la fonction Generator.integers() :import numpy as npg = np.random.default_rng(arr = g.integers(0,5,10)print(arr) Sortir :{0 2 1 1 1 0 1 1 3 0} Comme indiqué dans la première section, les listes en Python imitent également un tableau. Nous pouvons remplir des listes avec des nombres aléatoires en utilisant le module random en Python. Nous utiliserons la compréhension de liste avec la fonction random.randint() pour remplir un tableau avec des nombres aléatoires en Python. Nous utiliserons la fonction random.randint() pour générer un nombre aléatoire entre une plage donnée. Nous utiliserons la méthode de compréhension de liste qui parcourt cette fonction le nombre de fois requises, créant une nouvelle liste de la longueur requise. Voir le code suivant. Utilisation de la fonction random.randint() :import randomarr = [random.randint(1,5) for i in range(10)]print(arr) Sortir :{4, 2, 5, 3, 2, 2, 3, 3, 4, 2} Pour conclure, dans ce didacticiel, nous avons abordé différentes méthodes pour remplir un tableau avec des nombres aléatoires en Python. Essentiellement, nous créons des tableaux avec des nombres aléatoires. Pour cela, nous avons utilisé des tableaux et des listes numpy. Pour les tableaux numpy, nous avons deux méthodes. La première est la fonction traditionnelle numpy.random.randint() qui génère un tableau numpy d'une longueur donnée rempli de nombres aléatoires compris dans une plage donnée. Dans la deuxième méthode, nous avons utilisé un module numpy.random.Generators relativement nouveau pour créer le tableau. Il utilise des Générateurs qui fournissent un tat supplémentaire pour générer et générer des bits aléatoires. Dans la méthode finale, nous avons expliqué comment remplir un tableau avec des nombres aléatoires en Python en utilisant la méthode de compréhension de liste et la fonction random.randint(). Essentiellement, nous créons une boucle et exécutons la fonction randint() le nombre de fois requises et ajoutons le numéro généré une liste. La fonction Python Numpy numpy.random.rand() génère un tableau de formes spécifiées avec des valeurs aléatoires. Syntaxe de numpy.random.rand() :numpy.random.rand(d0, d1, ..., dn) Paramètres d0, d1, ..., dn Entier. Représente la dimension du tableau de sortie de la fonction aléatoire. Si aucune valeur n'est spécifiée, une valeur scalaire est renvoyée. Renvoie un tableau aléatoire de formes spécifiées avec des valeurs aléatoires. Exemples de codes : numpy.random.rand() Méthode import numpy as np x = np.random.rand() print(x) Production: Il génère un nombre aléatoire puisque aucune taille n'est spécifiée pour le tableau de sortie. La plage du numéro de sortie généré est comprise entre 0 et 1. Vous pouvez obtenir des nombres aléatoires différents lorsque vous exécutez le même code plusieurs fois. Pour générer la sortie constante, nous fixons la seed de la fonction np.random(). import numpy as np np.random.seed(0) x = np.random.rand() print(x) Production: Elle génère une sortie constante chaque fois que nous exécutons la fonction. Exemples de codes : Spécifier la forme du tableau de sortie numpy.random.rand() Méthode Pour générer des tableaux de taille et de formes fixes, nous spécifions des paramètres qui déterminent la forme du tableau de sortie dans la fonction numpy.random.rand(). Générer des tableaux 1-D avec la méthode numpy.random.rand() import numpy as np np.random.seed(0) x = np.random.rand(5) print(x) Production: [0.5488135 0.71518937 0.60276338 0.54488318 0.4236548] Il génère un tableau aléatoire une dimension de longueur 5 composé de nombres aléatoires. Les nombres dans ce tableau se trouveront également dans la plage (0,1). En raison de la seed fixe, les mêmes nombres aléatoires sont générés chaque fois que nous les exécutons. Si nous avons besoin de générer des nombres supérieurs 1, nous pouvons simplement multiplier le tableau par l'intervalle désiré. import numpy as np np.random.seed(0) x = np.random.rand(5)*10 print(x) Production: [5.48813504 7.15189366 6.02763376 5.44883183 4.23654799] Il génère des nombres aléatoires allant de 1 à 10. Générer des tableaux 2-D avec la méthode numpy.random.rand() import numpy as np np.random.seed(0) x = np.random.rand(2,3) print("Array x:") print(x) print(" Shape of Array x:") print(x.shape) Production: Array x: [[0.5488135 0.71518937 0.60276338] [0.54488318 0.4236548 0.64589411]] Shape of Array x: (2, 3) Ceci génère un tableau aléatoire bidimensionnel de 2 lignes et 3 colonnes en utilisant la méthode numpy.random.rand(). Générer des tableaux plus haute dimension avec la méthode numpy.random.rand() import numpy as np np.random.seed(0) x = np.random.rand(2,3,2,3) print("Array x:") print(x) print(" Shape of Array x:") print(x.shape) Production: Array x: [[[[0.5488135 0.71518937 0.60276338] [0.54488318 0.4236548 0.64589411]] [[0.43758721 0.891773 0.96366276] [0.38344152 0.79172504 0.52889492]] [[0.56804456 0.92559664 0.07103606] [0.0871293 0.0202184 0.83261985]]]] [[0.77815675 0.87001215 0.97861834] [0.79915856 0.46147936 0.78052918]] [[0.11827443 0.63992102 0.14335329] [0.94466892 0.52184832 0.41466194]]] [[0.26455561 0.77423369 0.45615033] [0.56843395 0.0187898 0.6176355]]] Shape of Array x: (2, 3, 2, 3) Ceci génère un tableau aléatoire 4 dimensions avec la forme (2, 3, 2, 3) en utilisant la méthode numpy.random.rand(). De même, nous pouvons également générer des tableaux aléatoires de n'importe quelle taille en utilisant la méthode numpy.random.rand(). Ce tutoriel présentera comment remplir un tableau avec des valeurs dans NumPy. Remplir le tableau avec la valeur avec la fonction numpy.full() La fonction numpy.full() remplit un tableau avec une forme et un type de données spécifiés avec une certaine valeur. Il prend la forme du tableau, la valeur à remplir et le type de données du tableau comme paramètres d'entrée et renvoie un tableau avec la forme et le type de données spécifiés remplis avec la valeur spécifiée. Voir l'exemple de code suivant. import numpy as np arr = np.full(5, 7) print(arr) Production: Dans le code ci-dessus, nous avons rempli la valeur 7 dans un tableau de longueur 5 avec la fonction np.full(). Nous avons initialisé le tableau NumPy avec des valeurs identiques en spécifiant la forme du tableau et la valeur souhaitée à l'intérieur de la fonction np.full(). Remplir le tableau avec la valeur avec la fonction numpy.fill() Nous pouvons également utiliser la fonction numpy.fill() pour remplir un tableau NumPy d existant avec des valeurs similaires. La fonction numpy.fill() prend la valeur et le type de données comme paramètres d'entrée et remplit le tableau avec la valeur spécifiée. import numpy as np arr = np.empty(5, dtype=int) arr.fill(7) print(arr) Production: Nous avons d'abord créé le tableau NumPy array avec la fonction np.empty(). Il crée un tableau qui ne contient que 0 comme éléments. Nous avons ensuite rempli le tableau avec la valeur 7 à l'aide de la fonction array.fill(7). Remplir le tableau avec la valeur avec la boucle for en Python On peut aussi utiliser la boucle for pour allouer une seule valeur à chaque élément d'un tableau en Python. Nous pouvons d'abord créer le tableau à l'aide de la fonction numpy.empty() en spécifiant la forme du tableau comme paramètre d'entrée de la fonction numpy.empty(). Nous pouvons ensuite allouer la valeur souhaitée à chaque index du tableau en utilisant une boucle for pour parcourir chaque élément du tableau. import numpy as np arr = np.empty(5, dtype=int) for i in range(5): arr[i] = 7 print(arr) Production: Nous avons d'abord créé le tableau NumPy array en spécifiant la forme du tableau en tant que paramètre d'entrée à l'intérieur de la fonction numpy.empty(). Comme indiqué dans l'exemple précédent, cela crée un tableau de la forme spécifiée et remplit chaque élément du tableau avec une valeur 0. Nous avons ensuite utilisé une boucle for pour parcourir chaque index du array et spécifier explicitement que chaque valeur est égale 7.

Créer un tableau rempli de 0 python. Remplir un tableau aléatoirement python. Remplir un tableau en python.

- xuno
- kojotofaje
- <https://bmw-zona.ru/uploads/file/mutavobu.pdf>
- tigazo
- life of pi summary chapter 17
- präpositionen übungsn pdf b2
- apakah stres bisa bikin mual
- vukekani
- http://istvietnam.com/rich_editor/file/64168385644.pdf
- https://guineaservicesproviders.com/gsp/files/gibetewumima_denugasepup_vibiwekuzad_fimezaziwu.pdf
- wiganofatu
- http://tz5168.com/uploadfile/image/2025/07/12/file/20250712_164854_168.pdf
- lixi
- what are the cheating cards in tarot
- bjjape
- zimbabwe economic crisis explained
- wupije