

Python

Listes Python

Yannick CHISTEL

Lycée Dumont d'Urville - CAEN

Décembre 2019

Parcourir un tableau

Tableaux et indices négatifs

On peut accéder aux derniers éléments d'un tableau avec des indices négatifs :

- `liste[0]` : première valeur du tableau
- `liste[-1]` : dernière valeur du tableau
- `liste[-2]` : avant-dernière valeur du tableau

Itérer sur les éléments

On peut accéder à un élément d'une liste avec une boucle `for` et la syntaxe suivante :

for `e` **in** liste :

instruction avec `e`

Exemple

Soit `t=[1,2,3,4,5,6]`

- ➊ `print(t[-1])` affiche la valeur 6
- ➋ `print(t[-2])` affiche la valeur 5
- ➌ `for k in t :`
 `print(k, end=' ')` affiche 1 2 3 4 5 6

Créer un tableau par **compréhension**

Création d'un tableau

La construction d'un tableau par compréhension introduit la boucle **for** à l'intérieur des crochets du tableau à construire. La syntaxe est de la forme :

[valeur **for** i **in** range(dimension du tableau)]

Exemple

- ❶ Construire par compréhension une liste ordonnée de nombres :
 - a) [i **for** i **in** range(5)] construit le tableau [0, 1, 2, 3, 4]
 - b) [i**2 **for** i **in** range(5)] construit le tableau [0, 1, 4, 9, 16]
 - c) [2*i+1 **for** i **in** range(5)] construit le tableau [1, 3, 5, 7, 9]
- ❷ Construire un tableau à partir des valeurs d'un autre tableau :
t=[2,3,5,7,11,13]
p=[x**2 **for** x **in** t]
Le tableau p a pour valeur [4, 9, 25, 49, 121, 169]
- ❸ Construire un tableau dont les valeurs sont des chaînes de caractères :
direction=['nord','sud','est','ouest']
DIRECTION=[e.upper() **for** e **in** direction]
Le tableau DIRECTION a pour valeur ['NORD', 'SUD', 'EST', 'OUEST']

Copier un tableau : **list**

Méthode

On a vu que la copie d'un tableau ne se fait pas par affectation puisque les variables vont au final désigner le même tableau.

Pour déclarer une nouvelle variable en lui affectant les valeurs d'un tableau existant, on utilise la fonction **list** avec la syntaxe :

```
tableau2 = list(tableau1)
```

Exemple

```
t=[10,20,30,40,50]
```

```
u=list(t)
```

```
print(u) affiche [10,20,30,40,50]
```

le tableau est copié dans la variable u.

```
for i in range(5) :
```

```
    u[i]=u[i]/10
```

ici, chaque valeur de u est divisée par 10

```
print(t) affiche [10,20,30,40,50]
```

le tableau t n'a pas changé

```
print(u) affiche [1,2,3,4,5]
```

le tableau u a ses valeurs divisées par 10

Modifier les tableaux

Ajouter des valeurs

Un tableau est de dimension fixée. Mais, en python, il est possible d'agrandir un tableau en ajoutant des valeurs. Deux méthodes sont possibles :

- 1 Par concaténation de deux tableaux existants avec l'opérateur `+` ;
- 2 En utilisant la fonction **append** qui permet d'ajouter une valeur en fin de tableau.

Exemple

- 1 Par concaténation de tableaux :
`t=[0,1,2]`
`u=[3,4,5]`
`s=t+u` *le tableau u a pour valeur [0,1,2,3,4,5]*
- 2 Avec la fonction **append** : `t=[0,1,2]`
`t.append(3)` *le tableau t a pour valeur [0,1,2,3]*
`t.append(4)` *le tableau t a pour valeur [0,1,2,3,4]*
- 3 Avec la fonction **append** et une boucle **for** : `t=[0,1,2]`
`u=[3,4,5]`
for e in u :
 `t.append(e)` *le tableau t a pour valeur [0, 1, 2, 3, 4, 5]*

Présentation

Il existe des fonctions que l'on peut appliquer sur un tableau, pour ajouter des éléments (append) et pour connaître le nombre d'éléments (len). En voici d'autres fonctions (liste non exhaustive) :

- La fonction copy qui recopie les valeurs d'un tableau dans un autre ;
- La fonction pop qui supprime la dernière valeur d'un tableau ;
- La fonction insert qui insère une valeur pour un indice donné du tableau ;
- La fonction remove qui supprime une valeur pour un indice donné du tableau ;
- La fonction sort trie le tableau dans l'ordre croissant.

Exemple

Soit un tableau : $t=[0,1,2,3]$

- 1 $u=t.copy()$ *le tableau u est créé et a les mêmes valeurs que t*
- 2 $u.pop()$ *la dernière valeur du tableau u est supprimée ; u vaut $[0,1,2]$*
- 3 $u.insert(1,4)$ *la valeur 4 est insérée à l'indice 1 ; u vaut $[0,4,1,2]$*
- 4 $u.remove(1)$ *si elle existe, la valeur indiquée est supprimée ; u vaut $[0,4,2]$*
- 5 $u.sort()$ *le tableau u est trié ; u vaut $[0,2,4]$*

Présentation

Un tableau peut contenir tout type de valeurs : des entiers (int), des chaînes de caractères (string), des nombres réels (float) et aussi des tableaux !

Pour accéder à une valeur de ce tableau, on utilise un premier indice pour sélectionner le tableau où se trouve la valeur puis un second indice pour obtenir la valeur dans le tableau sélectionné.

Exemple

❶ `t=[[4,5],[6,7],[8,9]]`

Le tableau `t` contient 3 tableaux de dimension 2 ; `t` est un tableau de dimension 3×2 ;

le tableau `[4,5]` est d'indice 0, le tableau `[6,7]` d'indice 1 et le tableau `[8,9]` d'indice 2 ;
les valeurs ont pour indice 0 et 1 pour chacun des trois tableaux ;

`print(t[1][0])` *tableau d'indice 1, valeur d'indice 0 ; affiche la valeur 6*

❷ On initialise un tableau de tableaux par compréhension :

`t=[[0]*3 for i in range(3)]` le tableau `t` vaut `[[0,0,0], [0,0,0], [0,0,0]]`