



Олексій Васильєв

Мова програмування Python

Київ 2020



Лекція 3. Списки і кортежі



- Створення списку
- Знайомство з кортежами
- Доступ до елементів
- Виконання зрізу
- Операції зі списками і кортежами



Створення списку

Список (тип list) - впорядкований набір значень (елементів)

Створення списку:

- **Перерахунок елементів у квадратних дужках:**

nums=[1,3,5]

- **За допомогою функції list():**

nums=list(1,3,5)

- **Размір списку визначається за допомогою функції len()**

- Звернення до елементу списку - за індексом
- Індиксація починається з нуля
- Від'ємний індекс - відлік з кінця списку

Наприклад:

len(nums) - розмір списку

nums[0] - перший елемент

nums[-1] - останній елемент



Створення зрізу

Два індекси:

- `список[i:j+1]` - список з елементів з індексами від i до j включно
- `список[i:]` - список з елементів з індексами від i до кінця вихідного списку
- `список[:j+1]` - список з елементів, починаючи з першого i до елемента з індексом j
- `список[:]` - копія списку

Три індекси:

- `список[i:j+1:k]` - список з елементів з індексами від i до j включно з інтервалом k
- якщо $k < 0$ - вибірка у зворотному порядку
- `список[::-1]` - копія списку в зворотному порядку



Створення списку

Програма (UsingList.py)

```
# Список з чисел:
nums=[5,10,1,60,25,3]
# Відображення вмісту списку:
print("Список з чисел:",nums)
# Довжина списку:
print("Довжина списку:",len(nums))
# Перший елемент:
print("Перший елемент:",nums[0])
# Останній елемент:
print("Останній елемент:",nums[-1])
# Найбільше значення:
print("Найбільше значення:",max(nums))
# Найменше значення:
print("Найменше значення:",min(nums))
# Сума:
print("Сума:",sum(nums))
# Список в зворотному порядку:
print("Список в зворотному порядку:",list(reversed(nums)))
# Сортвання за зростанням значень:
print("За зростанням значень:",sorted(nums))
# Сортвання за спаданням значень:
print("За спаданням значень:",sorted(nums,reverse=True))
# Вихідний список:
print("Вихідний список:",nums)
# Зміна значення елемента списку:
nums[1]="текст"
# Відображення вмісту списку:
print("Після внесення змін:",nums)
```



Створення списку

```
# Отримання зрізу:
print("Отримання зрізу:", nums[1:len(nums)-1])
# Заміна частини елементів списку:
nums[1:-1] = ["A", "B"]
# Список після заміни елементів:
print("Після заміни елементів:", nums)
# Список чисел від 5 до 10:
nums = list(range(5, 11))
print("Список чисел від 5 до 10:", nums)
# Видалення елементів зі списку:
nums[2:4] = []
print("Після видалення двох елементів:", nums)
# Видалення останнього елементу:
del nums[len(nums)-1]
print("Видалено останній елемент:", nums)
# Непарні числа:
nums = [2*k+1 for k in range(5)]
print("Непарні числа:", nums)
# Список з символів створюється на основі тексту:
syms = list("Python")
# Відображення вмісту списку:
print("Список з символів:", syms)
# Два перших символи:
print("Два перших символи:", syms[:2])
# Решта символів:
print("Решта символів:", syms[2:])
```

**Програма (UsingList.py) -
продовження**



Створення списку

Результат виконання:

Список з чисел: [5, 10, 1, 60, 25, 3]

Довжина списку: 6

Перший елемент: 5

Останній елемент: 3

Найбільше значення: 60

Найменше значення: 1

Сума: 104

Список в зворотному порядку: [3, 25, 60, 1, 10, 5]

За зростанням значені: [1, 3, 5, 10, 25, 60]

За спаданням значень: [60, 25, 10, 5, 3, 1]

Вихідний список: [5, 10, 1, 60, 25, 3]

Після внесення змін: [5, 'текст', 1, 60, 25, 3]

Отримання зрізу: ['текст', 1, 60, 25]

Після заміни елементів: [5, 'А', 'В', 3]

Список чисел від 5 до 10: [5, 6, 7, 8, 9, 10]

Після видалення двох елементів: [5, 6, 9, 10]

Видалено останній елемент: [5, 6, 9]

Непарні числа: [1, 3, 5, 7, 9]

Список з символів: ['P', 'y', 't', 'h', 'o', 'n']

Два перших символи: ['P', 'y']

Решта символів: ['t', 'h', 'o', 'n']



Завдання - 1

Напишіть програму, в якій створюється і відображається список, що містить степені двійки (числа **1**, **2**, **4**, **8**, і так далі)

Завдання - 2

Напишіть програму, в якій створюється список з чисел, котрі при діленні на **5** дають у залишку **3** (такі числа розраховуються за формулою $5k + 3$, де $k = 0, 1, 2, \dots$). Відобразити цей список в прямому і зворотному порядку



Створення кортежу

Кортеж (тип tuple) - незмінюваний впорядкований набір значень (елементів)

Створення кортежу:

- **Перерахунок елементів у круглих дужках:
A=(1,3,5) або A=() або A=(100,)**
- **Круглі дужки можна не використовувати:
A=1,3,5 або A=100,**
- **За допомогою функції tuple():
A=tuple([1,3,5]) або A=tuple("Python")**
- **Вміст кортежу змінити не можна**
- **Розмір кортежу визначається за допомогою функції len()**
- **Доступ до елементів кортежу - за допомогою індексу**
- **Для кортежу можна виконувати зріз**



Створення кортежу

```
# Різні способи створення кортежів:
Alpha=(5,10,15,"двадцять")
Bravo=100,["один","два","три"],200
Charlie=tuple([1,2,3,(4,5,6,7,8,9)])
Delta=tuple("ABCDEF")
Echo=tuple(2*k for k in range(11))

# Зчитування значень елементів і отримання зрізу:
print("Alpha:",Alpha)
print("Елементів:",len(Alpha))
print("Перший:",Alpha[0])
print("Останній:",Alpha[-1])
print("Bravo:",Bravo)
print("Елементів:",len(Bravo))
print("Bravo[1]:",Bravo[1])
print("Bravo[1][2]:",Bravo[1][2])
print("Charlie:",Charlie)
print("Елементів:",len(Charlie))
print("Charlie[3]:",Charlie[3])
print("Charlie[3][1:4]:",Charlie[3][1:4])
print("Delta:",Delta)
print("Елементів:",len(Delta))
print("Delta[-3:]:",Delta[-3:])
print("Echo:",Echo)
Foxtrot=tuple(Echo[k] for k in range(len(Echo)) if k%2==0)
print("Foxtrot:",Foxtrot)
Golf=Echo[2:5]
print("Golf:",Golf)
```

Програма (UsingTuples.py)



Створення кортежу

Результат виконання:

```
Alpha: (5, 10, 15, 'двадцять')
Елементів: 4
Перший: 5
Останній: двадцять
Bravo: (100, ['один', 'два', 'три'], 200)
Елементів: 3
Bravo[1]: ['один', 'два', 'три']
Bravo[1][2]: три
Charlie: (1, 2, 3, (4, 5, 6, 7, 8, 9))
Елементів: 4
Charlie[3]: (4, 5, 6, 7, 8, 9)
Charlie[3][1:4]: (5, 6, 7)
Delta: ('A', 'B', 'C', 'D', 'E', 'F')
Елементів: 6
Delta[-3:]: ('D', 'E', 'F')
Echo: (1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024)
Foxtrot: (1, 4, 16, 64, 256, 1024)
Golf: (4, 8, 16)
```



Операції зі списками та кортежами

Основні операції:

- зчитування значення елементу (або групи елементів);
- зміна значення елементу;
- вставка елементу (або групи елементів) в послідовність;
- видалення елементу (або групи елементів) з послідовності;
- зміна порядку елементів в послідовності.



Операції зі списками

**Програма
(UsingLists_2.py)**

```
A: [10, 20, 30]
B: ['Python', [1, 2]]
C: [10, 20, 30, 'Python', [1, 2]]
C: [10, 20, 30, 'Python', [1, 2], 100]
C: [10, 30, 'Python', [1, 2], 100]
C: [200, 10, 30, 'Python', [1, 2], 100]
C: ['A', 'B', 'Python', [1, 2], 100]
C: ['A', 'B', 8, 9, 'Python', [1, 2], 100]
C: ['A', 'B', 7, 9, 'Python', [1, 2], 100]
```

```
# Створення списків:
A=[10,20,30]
print("A:",A)
B=["Python",[1,2]]
print("B:",B)
# Розрахунок суми списків:
C=A+B
print("C:",C)
# Додавання елементу в кінець списку:
C+=[100]
print("C:",C)
# Видалення елементу списку:
C[1:2]=[]
print("C:",C)
# Додавання елементу на початок списку:
C=[200]+C
print("C:",C)
# Заміна декількох елементів у списку:
C[:3]=["A","B"]
print("C:",C)
# Вставка елементів у список:
C[2:2]=[8,9]
print("C:",C)
# Присвоювання значення елементу списку:
C[2:3]=[7]
print("C:",C)
```



Операції з кортежами

**Програма
(UsingTuples_2.py)**

```
A: (10, 20, 30)
B: ('Python', (1, 2))
C: (10, 20, 30, 'Python', (1, 2))
C: (10, 20, 30, 'Python', (1, 2), 100)
C: (10, 30, 'Python', (1, 2), 100)
C: (200, 10, 30, 'Python', (1, 2), 100)
C: ('A', 'B', 'Python', (1, 2), 100)
C: ('A', 'B', 8, 9, 'Python', (1, 2), 100)
C: ('A', 'B', 7, 9, 'Python', (1, 2), 100)
```

```
# Створення кортежів:
A=(10,20,30)
print("A:",A)
B=("Python", (1,2))
print("B:",B)
# Розрахунок суми кортежів:
C=A+B
print("C:",C)
# "Додавання" елементу в кінець кортежу:
C+=(100,)
print("C:",C)
# "Видалення" елементу кортежу:
C=C[:1]+C[2:]
print("C:",C)
# "Додавання" елементу у початок:
C=(200,)+C
print("C:",C)
# "Заміна" декількох елементів:
C=("A","B")+C[3:]
print("C:",C)
# "Вставка" елементів у кортеж:
C=C[:2]+(8,9)+C[2:]
print("C:",C)
# "Присвоєння" значення елементу:
C=C[:2]+(7,)+C[3:]
print("C:",C)
```



Створення вибірки

Генератори послідовностей:

вираз **for** змінна **in** діапазон **if** умова

знач **if** умова **else** знач **for** змінна **in** діапазон

знач **if** умова **else** знач **for** змінна **in** діап **if** умова

Також можна використовувати зрізи

При множенні списку/кортежу на число створюється новий список/кортеж кратним повторенням вмісту вихідного списку/кортежу



Створення вибірки

Кортеж чисел:

```
A=tuple(k for k in range(1,21) if k%3!=0)
```

```
print(A)
```

Список чисел:

```
B=[2**(k//2) if k%2==0 else 3**(k//2) for k in range(15)]
```

```
print(B)
```

Список чисел:

```
C=[0 if k==0 or k==1 else k**2 for k in range(13) if not k in [2,5,7]]
```

```
print(C)
```

Кортеж у зворотному порядку:

```
Alpha=A[::-1]
```

```
print(Alpha)
```

Елементи вибираються "через один", починаючи з першого:

```
Bravo=B[::2]
```

```
print(Bravo)
```

Елементи вибираються "через один", починаючи з другого:

```
Charlie=B[1::2]
```

```
print(Charlie)
```

**Програма
(ListsAndTuples.py)**

```
(1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 19, 20)
[1, 1, 2, 3, 4, 9, 8, 27, 16, 81, 32, 243, 64, 729, 128]
[0, 0, 9, 16, 36, 64, 81, 100, 121, 144]
(20, 19, 17, 16, 14, 13, 11, 10, 8, 7, 5, 4, 2, 1)
[1, 2, 4, 8, 16, 32, 64, 128]
[1, 3, 9, 27, 81, 243, 729]
```




Множення на число

```
Alpha=5*[0]
print(Alpha)
Bravo=(1,)*3
print(Bravo)
Charlie=[1,2]*3
print(Charlie)
Delta=[[1,2]]*3
print(Delta)
Echo=4*(1,[2,3])
print(Echo)
Foxtrot=([1]*2)*3
print(Foxtrot)
Golf=([1]*2,)*3
print(Golf)
```

Програма
(ListsAndTuples_2.py)

```
[0, 0, 0, 0, 0]
(1, 1, 1)
[1, 2, 1, 2, 1, 2]
[[1, 2], [1, 2], [1, 2]]
(1, [2, 3], 1, [2, 3], 1, [2, 3], 1, [2, 3])
[1, 1, 1, 1, 1, 1]
([1, 1], [1, 1], [1, 1])
```



Вкладені списки

```
# Імпорт функцій для генерування випадкових чисел:
from random import seed
from random import randint
# Створення вкладеного списку:
A=[[j+1)*10+i+1 for i in range(2)] for j in range(3)]
print("Список А:",A)
# Ініціалізація генератора випадкових чисел:
seed(2019)
# Список випадкових чисел:
B=[[randint(0,9) for i in range(3)] for j in range(2)]
print("Список В:",B)
# Список з буквами:
val='A'
m=2
n=3
C=[[' ' for i in range(m)] for j in range(n)]
for i in range(n):
    for j in range(m):
        C[i][j]=val
        val=chr(ord(val)+1)
print("Список С:",C)
# Список визначає кількість рядків у вкладеному списку:
size=[1,3,2,4]
# Створення вкладеного списку:
D=[['*' for k in range(s)] for s in size]
print("Список D:")
for s in D:
    print(s)
```

Програма
(ListsAndTuples_3.py)

```
Список А: [[11, 12], [21, 22], [31, 32]]
Список В: [[2, 3, 7], [2, 3, 3]]
Список С: [['A', 'B'], ['C', 'D'], ['E', 'F']]
Список D:
['*']
['*', '*', '*']
['*', '*']
['*', '*', '*', '*']
```



Копіювання списків

```
# Вихідний список:
A=[1,3,5]
# Присвоювання списків:
B=A
# Зміна значень елементів:
B[1]="Python"
A[2]=(10,20)
# Перевірка результату:
print(A)
print(B)
```

**Програма
(ListsAndTuples_4.py)**

```
[1, 'Python', (10, 20)]
[1, 'Python', (10, 20)]
```

```
# Вихідний список:
A=[1,3,[10,20],"Python",[40,50]]
# Створення поверхневої копії списку:
B=A[:]
C=A.copy()
print("Вихідні значення:")
print("A:",A)
print("B:",B)
print("C:",C)
# Внесення змін у вихідний список:
A[0]=[100,200]
A[2][1]=300
A[3]="Java"
A[4]=90
C[4][1]="C++"
print("Після внесення змін:")
print("A:",A)
print("B:",B)
print("C:",C)
```

**Програма
(ListsAndTuples_5.py)**

```
Вихідні значення:
A: [1, 3, [10, 20], 'Python', [40, 50]]
B: [1, 3, [10, 20], 'Python', [40, 50]]
C: [1, 3, [10, 20], 'Python', [40, 50]]
Після внесення змін:
A: [[100, 200], 3, [10, 300], 'Java', 90]
B: [1, 3, [10, 300], 'Python', [40, 'C++']]
C: [1, 3, [10, 300], 'Python', [40, 'C++']]
```



Повна копія

```
# Імпорт функції для створення повної копії:
from copy import deepcopy
# Вихідний список:
A=[1,3,[10,20],"Python",[40,50]]
# Створення повної копії списку:
B=deepcopy(A)
print("Вихідні значення:")
print("A:",A)
print("B:",B)
# Внесення змін у вихідний список:
A[0]=[100,200]
A[2][1]=300
A[3]="Java"
A[4]=90
print("Після внесення змін:")
print("A:",A)
print("B:",B)
```

**Програма
(ListsAndTuples_6.py)**

Вихідні значення:

A: [1, 3, [10, 20], 'Python', [40, 50]]

B: [1, 3, [10, 20], 'Python', [40, 50]]

Після внесення змін:

A: [[100, 200], 3, [10, 300], 'Java', 90]

B: [1, 3, [10, 20], 'Python', [40, 50]]



Функції і методи

- ❑ Метод **insert()** - вставка у список нового елементу
- ❑ Додавання елементу в кінець списку - метод **append()**
- ❑ Метод **extend()** добавляє в кінець списку елементи списку, вказаного аргументом метода
- ❑ Метод **pop()** дозволяє видалити елемент зі списку. Результат методу — значення елементу що видаляється
- ❑ Для видалення елементу використовують метод **remove()**
- ❑ Щоб змінити (на обернений) порядок елементів використовують метод **reverse()** або функцію **reversed()**
- ❑ Метод **index()** дозволяє визначити індекс елементу у списку
- ❑ Метод **count()** - кількість елементів з даним значенням
- ❑ Сортування - метод **sort()** (опція **reverse** зі значенням **True** або **False**) або функція **sorted()** (опція **reversed** зі значенням **True** або **False**)



Вставка/видалення

```
# Розмір списку:
n=10
# Початкове значення для списку:
A=[1,1]
# Заповнення списку:
for k in range(n-2):
    A.append(A[-1]+A[-2])
# Перевірка вмісту списку:
print("A:",A)
# Зміна порядку елементів:
for k in range(len(A)-1):
    A.append(A.pop(-k-2))
# Перевірка вмісту списку:
print("A:",A)
# Видалення найбільшого елементу:
A.remove(max(A))
# Перевірка вмісту списку:
print("A:",A)
# Видалення найменшого елементу:
A.remove(min(A))
# Перевірка вмісту списку:
print("A:",A)
# Додавання елементу в початок:
A.insert(0,A[0]+A[1])
```

```
# Перевірка вмісту списку:
print("A:",A)
# Пустий список:
B=[]
# Елементи списку переносяться в інший:
for k in range(len(A)//2):
    B.insert(0,A.pop(-1))
# Перевірка вмісту списків:
print("A:",A)
print("B:",B)
# Додавання елементу в кінець списку:
A.append(B)
# Перевірка вмісту списку:
print("A:",A)
# Видалення останнього елементу в списку:
A.pop(-1)
# Перевірка вмісту списку:
print("A:",A)
# Додавання елементів у список:
A.extend(B)
# Перевірка вмісту списку:
print("A:",A)
```

Програма (ListsAndTuples_7.py)



Вставка/видалення

Програма (ListsAndTuples_7.py) - результат

```
A: [1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
A: [55, 34, 21, 13, 8, 5, 3, 2, 1, 1]
A: [34, 21, 13, 8, 5, 3, 2, 1, 1]
A: [34, 21, 13, 8, 5, 3, 2, 1]
A: [55, 34, 21, 13, 8, 5, 3, 2, 1]
A: [55, 34, 21, 13, 8]
B: [5, 3, 2, 1]
A: [55, 34, 21, 13, 8, [5, 3, 2, 1]]
A: [55, 34, 21, 13, 8]
A: [55, 34, 21, 13, 8, 5, 3, 2, 1]
```



Пошук/підрахунок

```
# Підключення функцій для генерування випадкових чисел:
from random import *
# Ініціалізація генератора випадкових чисел:
seed(2019)
# Створення списку з випадкових чисел:
A=[randint(10,20) for k in range(15)]
# Вміст списку:
print("A:",A)
# Підрахунок елементів з різними значеннями:
for a in range(min(A),max(A)+1):
    print(a,"-",A.count(a))
# Найбільше, найменше і середнє значення:
print("Найменший:")
print("A[",A.index(min(A)),"]=" ,min(A), sep="")
print("Найбільший:")
print("A[",A.index(max(A)),"]=" ,max(A), sep="")
print("Середнє:" ,sum(A)/len(A))
# Сорткування списку:
B=sorted(A)
A.sort(reverse=True)
# Перевірка вмісту списків:
print("A:",A)
print("B:",B)
```

Програма (ListsAndTuples_8.py)



Пошук/підрахунок

A: [12, 13, 17, 12, 13, 20, 13, 20, 15, 20, 19, 20, 14, 16, 19]

12 - 2

13 - 3

14 - 1

15 - 1

16 - 1

17 - 1

18 - 0

19 - 2

20 - 4

Найменший:

A[0]=12

Найбільший:

A[5]=20

Середнє: 16.2

A: [20, 20, 20, 20, 19, 19, 17, 16, 15, 14, 13, 13, 13, 12, 12]

B: [12, 12, 13, 13, 13, 14, 15, 16, 17, 19, 19, 20, 20, 20, 20]

Програма (ListsAndTuples_8.py) - результат



Завдання - 3

Напишіть програму, в якій створюється числовий список. Список заповнюється випадковими числами. Потім елементи з парними індексами сортуються в порядку зростання, а елементи з непарними індексами сортуються в порядку спадання.

Завдання - 4

Напишіть програму, в якій створюється числовий список. Список заповнюється випадковими числами. Потім між кожною парою елементів цього списку вставляється новий елемент, який дорівнює сумі значень сусідніх елементів.



Домашнє завдання

[1] Напишіть програму, в якій створюється вкладений список із випадкових чисел. В матриці, котра реалізується даним вкладеним списком, видаляється рядок и стовпець. Номер рядка і стовпця, які треба видалити, вводиться користувачем.

[2] Напишіть програму для сортування списку методом бульбашки: послідовно порівнюються значення сусідніх елементів, і якщо значення елемента зліва більше за значення елемента справа, вони обмінюються значеннями. За один повний перебір елементів в списку саме велике значення опиняється на самій останній позиції у списку. За другий прохід передостаннім виявляється друге за величиною значення, і так далі.