

Выполнить качественный
конспект
срок (18.11.22-21.00)

Тема 2.6:

Списки в Python

Большинство программ работает не с отдельными переменными, а с набором переменных. Для хранения таких данных можно использовать структуру данных, называемую в Python **список** (в большинстве же языков программирования используется другой термин «массив»).

**Список (list) — тип данных,
предназначенный для
хранения набора или
последовательности разных
элементов.**

Список представляет собой
последовательность элементов:

- заключенных в квадратные скобки [],
- отделяющиеся друг от друга с помощью запятой,
- пронумерованных от 0, как символы в строке.

Создать список можно несколькими способами:

1. Получение списка через присваивание конкретных значений, используя конструкцию []

```
st = ["яблоко", "банан", "вишня"]  
print(st)
```

```
# ["яблоко", "банан", "вишня"]
```

<code>l = []</code>	<code># это пустой список</code>
<code>l = [25, 755, -40, 57, -41]</code>	<code># список целых чисел</code>
<code>l = [1.13, 5.34, 12.63, 4.6, 34.0, 12.8]</code>	<code># список из дробных чисел</code>
<code>l = ["Sveta", "Sergei", "Ivan", "Dasha"]</code>	<code># список из строк</code>
<code>l = ["Москва", "Иванов", 12, 124]</code>	<code># смешанный список</code>
<code>l = [[0, 0, 0], [1, 0, 1], [1, 1, 0]]</code>	<code># список, состоящий из списков</code>
<code>l = ['s', 'p', ['isok'], 2]</code>	<code># список из значений и списка</code>

Чтобы обратиться к элементу списка, достаточно указать его индекс:

`elements[3]`



Нумерация элементов списка в Python начинается с нуля

Существует также отрицательный индекс

```
st = ["яблоко", "банан", "вишня"]  
print(st[-3]) # яблоко
```


Ввод элементов списка:

Значения элементов списка можно также ввести в программу с помощью инструкций присваивания:

```
a[0] = ...  
a[1] = ...  
...
```

Можно использовать множественное присваивание:

```
a[0], a[1], ... = ...
```

2. Создание списка при помощи функции **List()**

```
l = list()           # пустой список  
l = list ('spisok') # 'spisok' - строка  
print(l)           #['s', 'p', 'i', 's', 'o', 'k'] - результат - список
```

Пример 2:

```
list(range(5, 160, 10))
```

```
# [5, 15, 25, 35, 45, 55, 65, 75, 85, 95, 105, 115, 125, 135, 145, 155]
```

3. Создание списка при помощи функции **Split()**

```
stroka = "Hello, world"      # stroka - строка  
l = stroka.split(",")      # l - список  
print(l)                   # ['Hello', ' world']
```

Функция `split` сканирует всю строку и разделяет ее в случае нахождения разделителя.

В строке должен быть как минимум один разделитель.

Им может выступать в том числе и символ пробела.

Пробел — разделитель по умолчанию.

Если параметр не задать, то разделение будет выполнено именно по символу пробела.

Синтаксис функции следующий:

`string.split(sep*, maxsplit*)`

`<string>` – любая допустимая строка в Python

`sep` – необязательный аргумент. По умолчанию метод `split()` разбивает строки по пробелам.

`maxsplit` – еще один опциональный аргумент, указывающий, сколько раз вы хотите разделить исходную строку `<string>`.

По умолчанию `maxsplit` имеет значение `-1`. При таком значении метод разбивает строку по всем вхождениям параметра `sep`.

Пример 1:

```
my_string = "Я программирую по 2 часа в  
день"
```

```
my_string.split()
```

```
#
```

```
['Я', ' программирую ', ' по ', '2', ' часа ', ' в ',  
' день ']
```

Давайте разделим `my_string` по запятым. Для этого нужно установить `sep = ","` или просто передать в метод `","` при вызове.

Пример 2:

```
my_string = " Яблоки, Апельсины, Груши, Бананы, Ягоды"  
my_string.split(",")  
# ['Яблоки', 'Апельсины', 'Груши', 'Бананы', 'Ягоды']
```

Пример 3:

```
my_string = " Яблоки, Апельсины,  
Груши, Бананы, Ягоды"
```

```
my_string.split(", ", 2)
```

```
# ['Яблоки', 'Апельсины', 'Груши,  
Бананы, Ягоды']
```

Длина списка?

4. Генераторы списков

список из 10 элементов, заполненный единицами

*l = [1]*10*

список l = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

l = [i for i in range(10)]

список l = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

*c = [c * 3 for c in 'list']*

print (c) # ['lll', 'iii', 'sss', 'ttt']

```
from random import randint  
l = [randint(10,80) for x in range(10)]  
# 10 чисел, сгенерированных случайным образом в диапазоне (10,80)
```

Срезы(slice) списка. Срезы хороши для получения подмножества значений списка.

```
z = [3, 7, 4, 2] # создаем список  
print(z[0:2]) # Вывод элементов с индексом от 0 до 2 (не включая 2)  
[3, 7]  
print(z[:3]) # Все, кроме индекса 3  
[3, 7, 4]  
print(z[1:]) # начиная с индекса 1 до конца списка  
[7, 4, 2]
```