

Списки

Создать список

```
animals = ["deer", "bear", "rabbit", "butterfly"]
numbers = [5, 4, 3, 2, 1, 0]
flags = [True, True, False, True]
```

Распечатать список

```
print(animals)
>>> ["deer", "bear", "rabbit", "butterfly"]
```

Как нумеруются элементы

```
0      1      2      3      4      5
['P', 'y', 't', 'h', 'o', 'n']
-6     -5     -4     -3     -2     -1
```

Получить элемент по индексу

```
hp_books = ["Философский Камень", "Тайная Комната", "Узник Азкабана"]
print(hp_books[2])
>>> Узник Азкабана
```

Как использовать переменную в качестве индекса

```
kings = ['Генрих', 'Людовик', 'Фридрих', 'Ричард']
# Сохраним нужный нам индекс в переменную
pos = 3
# Используем переменную в качестве индекса
print(kings[pos])
>>> Ричард
```

Получить элементы с конца списка

```
mylist = ['P', 'y', 't', 'h', 'o', 'n']

print(mylist[-1])
# выведет последний элемент - n
>>> n

print(mylist[-2])
# выведет предпоследний элемент - o
>>> o
```

Получить длину списка

```
mylist = ['P', 'y', 't', 'h', 'o', 'n']

list_len = len(mylist)
print(list_len)
>>> 6
```

Добавление элементов в список

Добавить один элемент в конец списка

```
brhd_ring = ["Арагорн", "Фродо", "Гэндальф", "Гимли"]

# Добавит элемент в конец списка
brhd_ring.append("Леголас")

print(brhd_ring)
>>> ["Арагорн", "Фродо", "Гэндальф", "Гимли", "Леголас"]
```

Добавить список элементов в конец списка

```
brhd_ring = ["Арагорн", "Фродо"]
brhd_ring_2 = ["Гэндальф", "Гимли", "Леголас"]

# Добавит список в конец списка
brhd_ring.extend(brhd_ring_2)

print(brhd_ring)
>>> ["Арагорн", "Фродо", "Гэндальф", "Гимли", "Леголас"]
```

Получение подписка с помощью срезов

Срез	Расшифровка	Вхождение	Результат
letters[:]	Все элементы	ABCDEFGH	['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H']
letters[3:6]	С элемента 3 до 6 (не включая его)	ABCDEFH	['D', 'E', 'F']
letters[3:]	Начиная с элемента 3	ABCDEFH	['D', 'E', 'F', 'G', 'H']
letters[0:6]	До элемента 6 (не включая его)	ABCDEFH	['A', 'B', 'C', 'D', 'E', 'F']
letters[:6]	До элемента 6 (короткая запись)	ABCDEFH	['A', 'B', 'C', 'D', 'E', 'F']
letters[0:-1]	Все, кроме последнего	ABCDEFGH	['A', 'B', 'C', 'D', 'E', 'F', 'G']
letters[0:-3]	Все, кроме 3 последних	ABCDEFH	['A', 'B', 'C', 'D', 'E']
letters[-3:]	Три последних	ABCDEFH	['F', 'G', 'H']
letters[::2]	Каждый второй	ABCDEFH	['A', 'C', 'E', 'G']
letters[::-2]	Каждый второй с конца	ABCDEFH	['H', 'F', 'D', 'B']

Как заменить элемент, если мы знаем индекс

```
flock = ['sheep', 'sheep', 'sheep', 'sheep']  
  
flock[2] = 'wolf'  
  
print(flock)  
  
>>> ['sheep', 'sheep', 'wolf', 'sheep']
```

Удалить элемент в списке по индексу

```
mylist = ['P', 'y', 't', 'h', 'o', 'n']

del mylist[0]
# Удалит элемент с индексом 0

print(mylist)
>>> ['y', 't', 'h', 'o', 'n']
```

Удалить элемент в списке по значению

```
flock = ['sheep', 'wolf', 'sheep', 'sheep']
# Если одинаковых элементов в списке несколько,
# удалит только первое вхождение
flock.remove('sheep')

print(flock)

>>> ['wolf', 'sheep', 'sheep']
```

Получить элемент с удалением

Если аргумент не передается:

```
mylist = ['P', 'y', 't', 'h', 'o', 'n']
element = mylist.pop()

print(element)
>>> 'n'

print(mylist)
>>> ['P', 'y', 't', 'h', 'o']
```

Если аргумент задан:

```
mylist = ['P', 'y', 't', 'h', 'o', 'n']
element = mylist.pop(0)

print(element)
>>> P

print(mylist)
>>> ['y', 't', 'h', 'o', 'n']
```

Выполнить проверку нахождения элемента в списке

```
'p' in ['p', 'y', 't']
>>> True

'p' in ['h', 'o', 'n']
>>> False
```

Объединить два списка

```
list_1 = ['P', 'y', 't']
list_2 = ['h', 'o', 'n']

list_united = list_1 + list_2

print(list_united)
>>> ['P', 'y', 't', 'h', 'o', 'n']
```

Повторить список несколько раз

```
mylist = ['P', 'y', 't']

list_repeated = mylist * 3

print(list_repeated)
>>> ['P', 'y', 't', 'P', 'y', 't', 'P', 'y', 't']
```

Вставить элемент в любое место списка по индексу

Пример

```
my_list = ['P', 'y', 't', 'h', 'o', 'n']

# Вставляем букву A по индексу 2
my_list.insert(2, 'A')

print(my_list)
>>> ['P', 'y', 'A', 't', 'h', 'o', 'n']
```

Пример с отрицательным индексом

```
my_list = ['P', 'y', 't', 'h', 'o', 'n']

# Вставляем букву A по индексу -1
my_list.insert(-1, 'A')

print(my_list)
>>> ['P', 'y', 't', 'h', 'o', 'A', 'n']
```

Определить индекс элемента в списке

```
my_list = ['P', 'y', 't', 'h', 'o', 'n']

# Находим индекс элемента y
index = my_list.index('y')

print(index)
>>> 1
```

Посчитать количество одинаковых элементов в списке

```
my_list = ['P', 'y', 't', 'h', 'o', 'n']

# Считаем количество вхождений элемента y
y_quantity = my_list.count('y')

print(y_quantity)
>>> 1
```

Отсортировать список по возрастанию

Метод `sort` без аргументов

```
my_list = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5]

# Сортируем список по возрастанию методом sort
my_list.sort()

print(my_list)
>>> [1, 1, 2, 3, 3, 4, 5, 5, 5, 6, 9]
```

Функция `sorted` без аргументов

```
my_list = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5]

# Создаем отсортированную копию списка функцией sorted
sorted_list = sorted(my_list)

print(sorted_list)
>>> [1, 1, 2, 3, 3, 4, 5, 5, 5, 6, 9]
```

Как отсортировать список по убыванию

Методом `sort` с аргументом `reverse=True`

```
my_list = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5]

# Сортируем список по убыванию методом sort
my_list.sort(reverse=True)

print(my_list)
>>> [9, 6, 5, 5, 5, 4, 3, 3, 2, 1, 1]
```

Функция `sorted` с аргументом `reverse=True`

```
my_list = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5]

# Создаем отсортированную по убыванию копию списка функцией sorted
sorted_list = sorted(my_list, reverse=True)
print(sorted_list)
>>> [9, 6, 5, 5, 5, 4, 3, 3, 2, 1, 1]
# Исходный список остается неизменным
print(my_list)
>>> [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5]
```