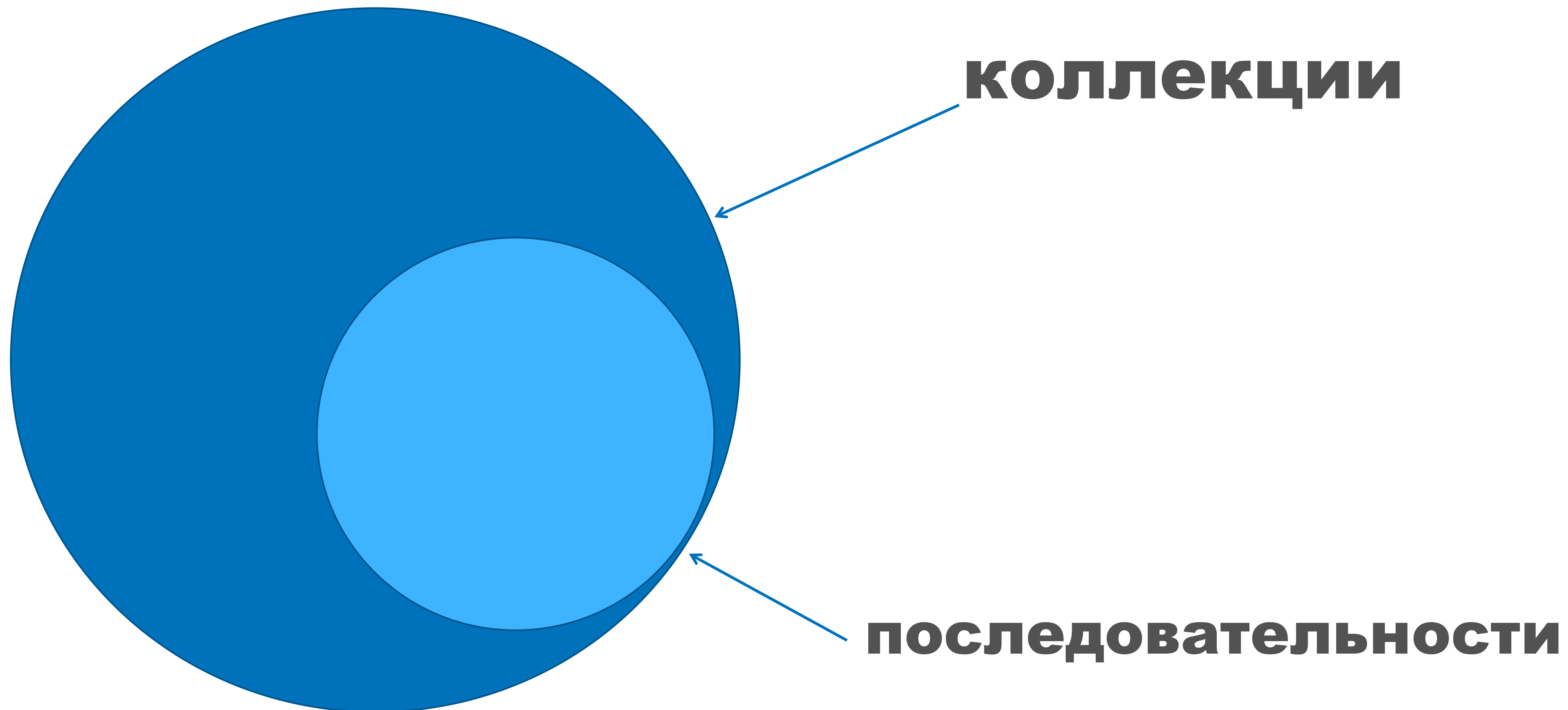


Последовательности: списки и кортежи

Докладчик: Евграфов Михаил

Последовательности

Последовательности



Конкатенация

конкатенация списков

```
>>> [1, 2, 3] + [4, 5]  
[1, 2, 3, 4, 5]
```

конкатенация кортежей

```
>>> (1, 2, 3) + (4, 5)  
(1, 2, 3, 4, 5)
```

конкатенация в составном присваивании

```
>>> array = [1, 2, 3]  
>>> array += [4, 5]          # array == [1, 2, 3, 4, 5]
```

Повторения

повторение списка

```
>>> [1, 2] * 2  
[1, 2, 1, 2]
```

повторение кортежа

```
>>> 2 * (3, 4, 5)  
(3, 4, 5, 3, 4, 5)
```

повторение в составном присваивании

```
>>> tuple_ = (1, 2)  
>>> tuple_ *= 2          # tuple_ == (1, 2, 1, 2)
```

Некорректные повторения

повторение 0 раз

```
>>> (1, 2) * 0  
()
```

повторение отрицательное число раз

```
>>> [3, 4, 5] * -1  
[]
```

повторение с некорректным операндом

```
>>> (1, 2) * 0.5
```

```
TypeError: ...
```

Оператор in

```
>>> 3 in [1, 2, 3]  
True
```

```
>>> 3 in (4, 5, 6)  
False
```

Отрицание оператора in

менее предпочтительный вариант отрицания

```
>>> not (3 in [1, 2, 3])
```

```
False
```

более предпочтительный вариант отрицания

```
>>> 3 not in [1, 2, 3]
```

```
False
```


Индексация

индексация списков

```
>>> array = [1, 2, 3]
```

```
>>> array[0]
```

1

индексация кортежей

```
>>> tuple_ = (4, 5, 6)
```

```
>>> tuple_[2]
```

6

Отрицательная индексация

индексация списков

```
>>> array = [1, 2, 3]
```

```
>>> array[-3]
```

1

индексация кортежей

```
>>> tuple_ = (4, 5, 6)
```

```
>>> tuple_[-1]
```

6

Некорректная индексация

```
>>> array = [1, 2, 3]
```

```
>>> array[0.5]
```

```
TypeError: ...
```

```
>>> tuple_ = (4, 5, 6)
```

```
>>> tuple_[42]
```

```
IndexError: ...
```


Срезка

с помощью вызова `slice()`

```
my_slice = slice(1, 10, 2)
```

в контексте индексации

с мгновенным использованием

```
array[1:10:2]
```

Использование срезов

```
arr = [1, 2, 3, 4, 5]
```

```
# все числа с позиций [1, 3)
```

```
arr[1:3]          # == [2, 3]
```

```
# все числа с позиций [0, 3]
```

```
arr[:3]           # == [1, 2, 3]
```

```
# все числа с позиций [1, 4]
```

```
arr[1:]           # == [2, 3, 4, 5]
```

```
# все числа с четными индексами
```

```
arr[::2]          # == [1, 3, 5]
```

```
# все числа в обратном порядке
```

```
arr[::-1]         # == [5, 4, 3, 2, 1]
```

```
# числа из диапазона [1, 3] в обратном порядке
```

```
arr[-2:-5:-1]     # == [4, 3, 2]
```

Проверка количества элементов

проверка количества элементов в списке

```
>>> len([1, 2, 3])  
3
```

проверка количества элементов в кортеже

```
>>> len(())  
0
```

проверка количества элементов в строке

```
>>> len("abcdef")  
6
```


Повторения с изменяемыми элементами

повторение с изменяемыми элементами

```
>>> seq = [[3]] * 3
```

```
>>> seq
```

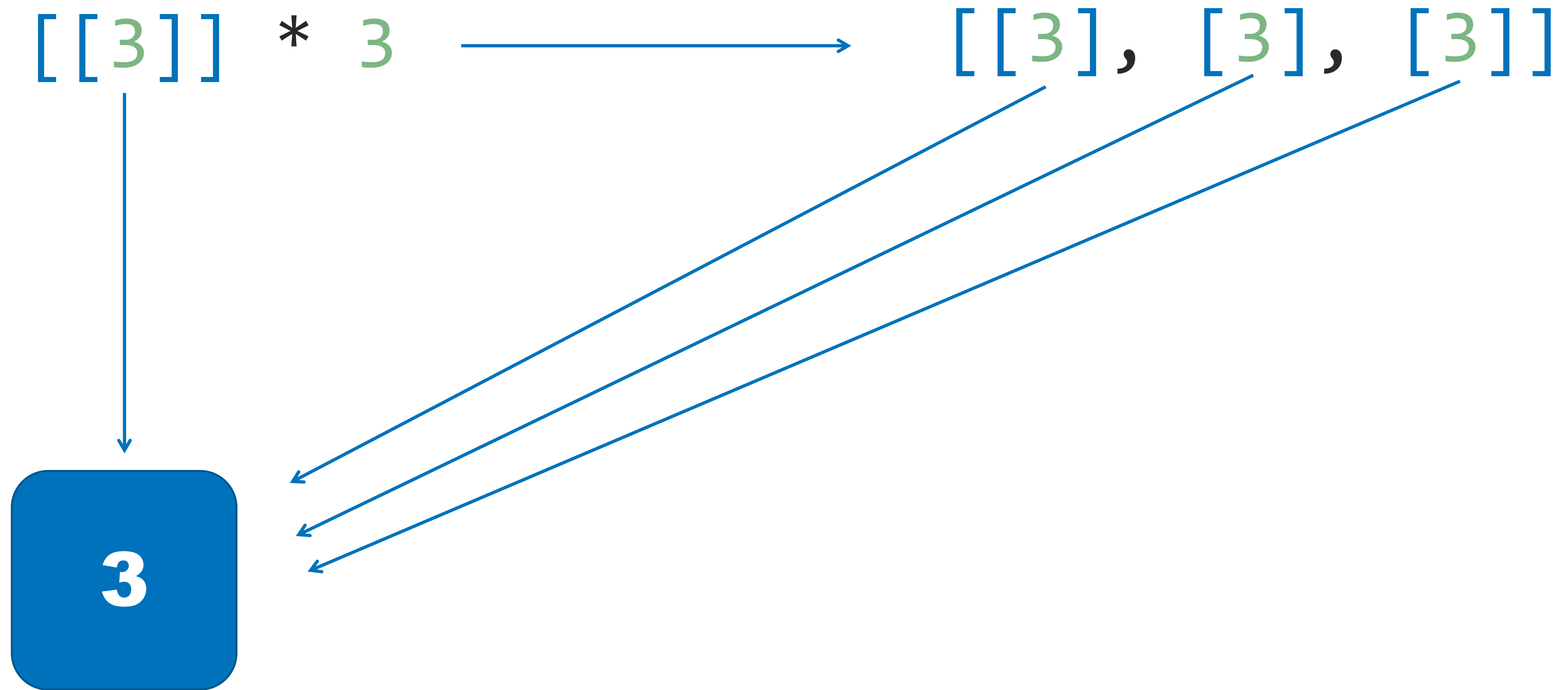
```
[[3], [3], [3]]
```

```
>>> seq[0][0] = 1
```

```
>>> seq
```

```
[[1], [1], [1]]
```

Повторения с изменяемыми элементами



Конкатенация с изменяемыми элементами

```
>>> seq1 = [[1]]
>>> seq2 = [[2]]
>>> seq3 = seq1 + seq2
>>> seq3
[[1], [2]]
```

```
>>> seq3[0][0] = 42
>>> seq3
[[42], [2]]
>>> seq1
[[42]]
```


Кортежи

tuple

создание кортежа из списка

```
>>> tuple([1, 2, 3])  
(1, 2, 3)
```

создание кортежа из другого кортежа

```
>>> tuple((3, 4, 5))  
(3, 4, 5)
```

создание кортежа из строки

```
>>> tuple("hello")  
( 'h', 'e', 'l', 'l', 'o' )
```

создание пустого кортежа

```
>>> tuple()  
( )
```

Литералы кортежей

(1, 2, 3)

1, 2, 3

(3.14,)

3.14,

()

== (1, 2, 3)

== (1, 2, 3)

== (3.14,)

== (3.14,)

пустой кортеж

Изменение значений кортежа

```
>>> tuple_ = (1, 2, [])
```

```
>>> tuple_  
(1, 2, [])
```

```
>>> tuple_[-1].append(5)
```

```
>>> tuple_  
(1, 2, [5])
```

Попытки явных изменений

```
>>> tuple_ = (1, 2, 3, 4, 5)
```

```
# попытка привязать ссылку к другому объекту
```

```
>>> tuple_[1] = 42
```

```
TypeError: ...
```

```
# попытка удалить ссылку
```

```
>>> del tuple_[1]
```

```
TypeError: ...
```

Логические операции

```
>>> (1, 2, 3) == (4, 5, 6)
```

```
False
```

```
>>> (1, 2, 3) != (4, 5, 6)
```

```
True
```

```
>>> (1, 2, 3) < (4, 5)
```

```
True
```

```
>>> (1, 2, 3) <= (4, 5)
```

```
True
```

```
>>> (1, 2, 3) > (4, 5)
```

```
False
```

```
>>> (1, 2, 3) >= (4, 5)
```

```
False
```

index

```
>>> tuple_ = (1, 2, 3, 4, 5)
```

```
# элемент со значением 3 имеет индекс 2
```

```
>>> tuple_.index(3)  
2
```

```
# в данном кортеже нет элемента "a"
```

```
>>> tuple_.index("a")  
ValueError: ...
```

count

```
>>> tuple_ = (1, 2, 3, 4, 3, 3)
```

```
>>> tuple_.count(3)
```

```
3
```

```
>>> tuple_.count("a")
```

```
0
```


Списки

list

создание списка из списка

```
>>> list([1, 2, 3])  
[1, 2, 3]
```

создание списка из кортежа

```
>>> list((3, 4, 5))  
[3, 4, 5]
```

создание списка из строки

```
>>> list("hello")  
['h', 'e', 'l', 'l', 'o']
```

создание пустого списка

```
>>> list()  
[]
```

Списковые литералы

```
[42, 3.14,
```

```
'string']
```

```
[100]
```

```
[]
```

Изменения списков

```
nums = [1, 2, 3, 4, 5]
```

```
nums[1] = 42
```

```
# [1, 42, 3, 4, 5]
```

```
nums[2:4] = [5, 6, 7, 8]
```

```
# [1, 42, 5, 6, 7, 8, 5]
```

```
nums[2:2] = [3, 3, 3]
```

```
# [1, 42, 3, 3, 3, 5, 6, 7, 8, 5]
```

```
nums[2:7] = []
```

```
# [1, 42, 7, 8, 5]
```

```
nums[:] = [1]
```

```
# [1]
```

```
del nums[0]
```

```
# []
```

Модифицирующие методы

```
>>> list.append(x)
>>> list.clear()
>>> list.extend(i)
>>> list.insert(i, x)
>>> list.pop(i=-1)
>>> list.remove(x)
>>> list.reverse()
>>> list.sort(key=None, reverse=False)
```


Цикл for

переменная цикла

итерируемый объект

for i in iterable:

...

тело цикла

nums = [1, 2, 3, 4, 5]

```
for num in nums:  
    print(num)
```

range

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

```
for i in range(1, 5, 2):  
    print(i)
```

map

```
nums_string = input()
nums = list(
    map(
        int, nums_string.split()
    )
)
```


Практическая часть

