

Язык программирования

Python

10 Цикл for



Оператор for

Рассмотрим циклы, представленные оператором for.

В цикле for есть переменная и есть множество значений, по которому «пробегаются» эта переменная.

Это множество может быть задано списком, кортежем, строкой и вообще любой коллекцией, у которой есть доступ к элементам и возможность навигации по ним (возможность взять следующий элемент), а также арифметической прогрессией, возвращаемой функцией range.

```
chars = 'str'
for char in chars:
    print(char, end = '')
```

Вывод:

```
str
```

Функция range

Вспомним, как работает функция range.

- range(stop) создаст последовательность целых чисел от 0 до stop не включительно;
- range(start, stop) создаст последовательность целых чисел от start до stop не включительно;
- range(start, stop, step) создаст последовательность целых чисел от start до stop не включительно с шагом step.

```
for i in range(5):  
    print(i, end = '')
```

Вывод:

01234

Оператор for

В множестве значений, по которым будет пробегаться переменная, могут находиться элементы разных типов.

```
items = [1, 2.0, True, 'four']  
for item in items:  
    print(item, end = ' ')
```

Вывод:

```
1 2.0 True four
```

Вычисление факториала

Факториал — это произведение всех чисел от 1 до num включительно.

```
num = int(input())
factorial = 1
for n in range(1, num + 1):
    factorial *= n
print('Факториал числа', num, '=', factorial)
```

Вывод для n = 3:

Факториал числа 3 = 6

Вывод для n = 4:

Факториал числа 4 = 24

Оператор break

Оператор `break` позволяет нам досрочно выйти из цикла.

```
my_lies = [False, False, True, False]
for lie in my_lies:
    if lie == True:
        break
    print(lie, end = ' ')
```

Вывод:

```
False False
```

Оператор continue

Оператор continue позволяет нам перейти к началу следующей итерации, минуя оставшееся тело цикла.

```
my_lies = [False, False, True, False]
for lie in my_lies:
    if lie == True:
        continue
    print(lie, end = ' ')
```

Вывод:

```
False False False
```

Ключевое слово else

Ключевое слово `else`, стоящее за телом цикла, позволяет нам выполнить блок инструкций после завершения цикла «естественным» путем. При завершении цикла оператором `break` блок операторов внутри `else` выполняться не будет.

```
my_lies = [False, False, False]
for lie in my_lies:
    if lie == True:
        break
    print(lie, end = ' ')
else:
    print('ok')
```

Вывод:

```
False False False
ok
```


Перебор элементов кортежа

```
days = ('Monday', 'Tuesday', 'Wednesday', 'Thursday',  
'Friday', 'Saturday', 'Sunday', )
```

```
for day in days:  
    print(days, end = ' ')
```

Или:

```
for i in range(0, len(days)):  
    print(days[i], end = ' ')
```

Вывод

```
Monday Tuesday Wednesday Thursday Friday Saturday Sunday
```

Итоги

Мы познакомились с такими понятиями, как оператор `for`, оператор `break`, оператор `continue`, ключевое слово `else`, а также попрактиковались в работе с интерпретатором в файловом режиме.