

Тема урока: цикл for

`for` **название переменной** `in range` (**количество повторений**):

блок кода

Функция `range()` с одним параметром

Рассмотрим программный код:

```
for i in range(10):  
    print('Привет', i)
```

Значение, которое мы указываем в скобках у функции `range()` обозначает количество итераций цикла, при этом переменная `i` принимает последовательно значения: `0, 1, 2, 3, 4, 5, 6, 7, 8, 9`.

Если быть более точным, то мы говорим, что функция `range(n)` генерирует **последовательность** чисел от `0` до `n-1`, а цикл `for` последовательно перебирает эту последовательность.

Перегрузка `range()` с двумя параметрами

Если мы хотим начинать последовательность не с `0`, а с какого-то другого числа, то мы можем использовать **перегрузку функции** `range()` принимающую два параметра. Например, вызов функции `range(1, 5)` сгенерирует последовательность чисел `1, 2, 3, 4` (будьте внимательны, правая граница не включительна). Если нам нужны числа от `1` до `5` включительно, то мы используем `range(1, 6)`.

Таким образом:

- `range(n)`: создает последовательность чисел `0, 1, 2, 3, ..., n - 1`;
- `range(n, m)`: создает последовательность чисел `n, n + 1, n + 2, ..., m - 2, m - 1`.

Напишем программу, которая выводит те числа из промежутка `[100;999]`, которые оканчиваются на `7`.

Используя функцию `range()` с двумя параметрами, получаем:

```
for i in range(100, 1000): # перебираем числа от 100 до 999  
    if i % 10 == 7: # используем остаток от деления на 10, для получения последней цифры  
        print(i)
```

Обратите внимание, в качестве второго параметра мы передали число `1000`.

Важно! Если первый параметр больше второго, то функция `range()` генерирует пустую последовательность. Например, вызов функции `range(10, 1)` приводит к генерации пустой последовательности.

Перегрузка `range()` с 3 параметрами

Передавая два параметра в функцию `range()` мы можем генерировать любую последовательность целых чисел с шагом `1`. Но, что делать если нужно поменять шаг? Как быть если мы хотим сгенерировать последовательность чисел `5, 10, 15, 20, 25`? В этом случае существует еще одна перегрузка функции `range()`, принимающая три параметра: `range(n, m, k)`. Первый параметр задает **старт последовательности**, второй параметр задает **стоп последовательности** и третий – **шаг генерации** чисел.

Например, вызов функции `range(1, 10, 2)` создаст последовательность чисел `1, 3, 5, 7, 9`, а вызов функции `range(5, 30, 5)` сгенерирует последовательность `5, 10, 15, 20, 25`.

Напишем программу, которая выводит все четные числа из промежутка `[56;170]`.

Используя функцию `range()` с тремя параметрами, получаем:

```
for i in range(56, 171, 2):  
    print(i)
```

Обратите внимание, мы можем использовать функцию `range()` с двумя параметрами:

```
for i in range(56, 171):
    if i % 2 == 0:
        print(i)
```

однако такой код получается менее эффективным.

Отрицательный шаг генерации

Если шаг генерации является положительным числом, то генерируемая последовательность будет **возрастать**. Мы можем указать отрицательный шаг генерации (третий параметр), что приведет к генерированию **убывающей** последовательности.

В случае отрицательного шага, мы должны гарантировать, что старт последовательности (первый параметр) больше чем конец последовательности (второй параметр).

Например, вызов функции `range(20, 16, -1)` создаст последовательность чисел 20, 19, 18, 17, а вызов функции `range(20, 10, -3)` сгенерирует последовательность 20, 17, 14, 11.

Напишем программу, которая отсчитывает от 5 до 1, а затем выводит текст `Взлетаем!!!`:

```
for i in range(5, 0, -1):
    print(i)
print('Взлетаем!!!')
```

Результатом будет:

```
5
4
3
2
1
Взлетаем!!!
```

Важно! Если величина шага отрицательна и первый параметр меньше второго, то функция `range()` генерирует пустую последовательность. Например, вызов функции `range(1, 10, -1)` приводит к генерации пустой последовательности.

Примеры использования функции `range()`

Вызов функции	Последовательность чисел
<code>range(10)</code>	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
<code>range(1, 10)</code>	1, 2, 3, 4, 5, 6, 7, 8, 9
<code>range(3, 7)</code>	3, 4, 5, 6
<code>range(7, 3)</code>	пустая последовательность
<code>range(2, 15, 3)</code>	2, 5, 8, 11, 14
<code>range(9, 2, -1)</code>	9, 8, 7, 6, 5, 4, 3
<code>range(3, 10, -2)</code>	пустая последовательность

Примечания

Примечание 1. Функция `range()` может принимать от одного до трех параметров: `range(n)`, `range(n, m)`, `range(n, m, k)`

- первый параметр – это старт последовательности (включительно);
- второй параметр – это стоп последовательности (не включительно);
- третий параметр – это величина шага.

Примечание 2. Функция `range()` может генерировать только целые числа, включая отрицательные.

Примечание 3. Величина шага не может равняться нулю. Следующий код:

```
for i in range(1, 10, 0):
    print(i)
```

приведет к ошибке **ValueError: range() arg 3 must not be zero.**

Вопросы:

Какую последовательность чисел даст вам вызов функции `range(8)` ?

Какую последовательность чисел даст вам вызов функции `range(1, 8)` ?

Какую последовательность чисел даст вам вызов функции `range(3, 11, 2)` ?

Какую последовательность чисел даст вам вызов функции `range(10, 0, -2)` ?

Решение задач

Последовательность чисел 1

Даны два целых числа $m \leq n$. Напишите программу, которая выводит все числа от m до n включительно.

Формат входных данных

На вход программе подаются два целых числа m и n , каждое на отдельной строке.

Формат выходных данных

Программа должна вывести числа в соответствии с условием задачи.

Входные данные	Выходные данные
1	1
9	2
	3
	4
	5
	6
	7
	8
	9

Последовательность чисел 2

Даны два целых числа m и n . Напишите программу, которая выводит все числа от m до n включительно в порядке возрастания, если $m < n$, или в порядке убывания в противном случае.

Формат входных данных

На вход программе подаются два целых числа m и n , каждое на отдельной строке.

Формат выходных данных

Программа должна вывести числа в соответствии с условием задачи.

Входные данные	Выходные данные
1	1
5	2
	3
	4
	5
7	7
1	6
	5
	4
	3
	2
	1
100	100
100	

Последовательность чисел 3

Даны два целых числа m и n ($m > n$). Напишите программу, которая выводит все нечетные числа от m до n включительно в порядке убывания.

Формат входных данных

На вход программе подаются два целых числа m и n , каждое на отдельной строке.

Формат выходных данных

Программа должна вывести числа в соответствии с условием задачи.

Входные данные	Выходные данные
77 62	77 75 73 71 69 67 65 63
6 -2	5 3 1 -1
9980 9972	9979 9977 9975 9973

Последовательность чисел 4

Даны два натуральных числа m и n ($m \leq n$). Напишите программу, которая выводит все числа от m до n включительно удовлетворяющие хотя бы одному из условий:

- число кратно 17;
- число оканчивается на 9;
- число кратно 3 и 5 одновременно.

Формат входных данных

На вход программе подаются два натуральных числа m и n ($m \leq n$), каждое на отдельной строке.

Формат выходных данных

Программа должна вывести числа в соответствии с условием задачи.

Примечание. Если чисел удовлетворяющих условию нет, выводить ничего не надо.

Входные данные	Выходные данные
1 20	9 15 17 19
17 17	17
19 19	19