

## Самое важное

### Функция `range()` и шаг

С этой функцией вы уже знакомы, она позволяет создавать последовательности чисел. Ранее вы научились задавать начало и конец этой последовательности. В этой теме рассматривается ещё один параметр этой функции — `step`, или шаг.

Этот параметр по умолчанию равен 1.

`range(9)` — по сути, это `range(0, 9, 1)`, то есть числа от 0 до 9 с шагом 1.

Теперь можно настраивать функцию и получать более гибкие последовательности под поставленные задачи.

### Синтаксис

`range(<start>, <stop>, <step>)`

### Пример

```
for i in range(10, 20, 2):  
    print(i)
```

В итоге в консоль мы выведем числа от 10 до 20 (не включая 20) с шагом 2: 10, 12, 14, 16, 18.

Шаг может быть любым (кроме 0). Например, можно использовать отрицательный шаг:

```
for i in range(20, 10, -2):  
    print(i)
```

Важно: в этом случае первый элемент должен быть больше второго. Надо понимать, как «дойти» от начала до конца при помощи указанного шага. Если бы был следующий вариант — `range(10, 20, -2)`, — то мы бы просто «пошли» в другую сторону:

$10 - 2 = 8$

$8 - 2 = 6$

...

...и до 20 мы бы никогда не дошли, что вызвало бы ошибку.

```
for i in range(20, 10, -2):  
    print(i)
```

В правильном же виде это алгоритм выведет: 20, 18, 16, 14, 12. Число 10, опять же, не будет включено в последовательность, так как оно является её концом!

**Начало, конец и шаг можно задавать не только явными цифрами, но и переменными!**

```
x = 20
y = 10
z = -2
```

```
for i in range(x, y, z):
    print(i)
```

Мы получим точно такой же результат, как выше. Однако использование переменных даёт нам возможность изменять нашу последовательность чисел «на ходу».

Мы можем задавать числа не вручную, а например через input():

```
x = int(input("Начало"))
y = int(input("Конец"))
z = int(input("Шаг"))
```

```
for i in range(x, y, z):
    print(i)
```

**Не допускай следующих ошибок!**

1. Не забывайте, что конец последовательности не учитывается в ней!

for i in range(5, 10, 5) — такой цикл будет иметь только одну итерацию, в которой переменная цикла будет равна 5.

for i in range(5, 11, 5) — такой цикл будет иметь уже две итерации (с числами 5 и 10).

2. Не забывайте про правильные границы и их связь с шагом:
  - если шаг положительный, то первое число должно быть меньше второго;
  - если шаг отрицательный, то первое число должно быть больше второго.