

Самое важное

Оператор for

Оператор for позволяет нам создавать новый вид цикла. Он, в отличие от while, работает строго ограниченное количество раз.

В этом его особенность — там, где мы заранее знаем количество итераций, лучше использовать цикл for. Там, где мы заранее не уверены в количестве итераций, лучше использовать цикл while.

Синтаксис

```
for <переменная_цикла> in <последовательность>:  
    <тело цикла>
```

Особенности

1. Важно отступом показать Python, какой код мы хотим выполнять внутри цикла.
2. Последовательность мы можем задавать совершенно любым способом (но пока основной способ — range(<число>).
3. Переменная цикла — такая же переменная, как и остальные. Это, например, значит, что имя, занятое переменной цикла, может конфликтовать с таким же именем обычной переменной!
4. В цикле while перед началом итерации выполнялась проверка условия. Здесь же выполняется операция перезаписи переменной:

```
for i in range(3):  
    print(i)
```

Мы этого не увидим, но перед началом каждой итерации будет выполняться операция: `i = <число из последовательности>`

1-я итерация: `i = 0`

2-я итерация: `i = 1`

3-я итерация: `i = 2`

Функция range()

Эта функция позволяет быстро создать нужную последовательность чисел.

Синтаксис

range(<start>, <stop>) — start и stop здесь указывают на начало и конец последовательности, причём start по умолчанию равен 0, то есть если мы напишем range(3), то получим последовательность 0, 1, 2 (от 0 до 3, не включая 3).

range(10, 15) позволит нам начать не с 0, а с 10: 10, 11, 12, 13, 14.

Не допускай следующих ошибок!

Не забывайте, что переменная цикла — такая же переменная, как и другие.

```
for x in range(10, 15):
```

`x += 1` — если вы прибавите +1 к значению переменной цикла, это не повлияет на количество итераций.

Как это происходит «под капотом» (иначе говоря, как Python выполняет эти инструкции изнутри): запускается цикл, инициализируется последовательность — 10, 11, 12, 13, 14, создаётся переменная цикла `x = 10`, выполняется тело цикла — `x += 1` (значение переменной увеличилось), тело цикла закончено, выполняется следующая итерация, и начинается она с `x = 11`!

Python не обращает внимание на то, что переменная уже содержит число 11, он просто берёт следующее значение последовательности, которая была создана до начала цикла, а там стоит число 11.

Важно: пользоваться переменной цикла можно, но нужно помнить, что она перезаписывается заново на каждой итерации цикла!