

## Циклы

Программирование предназначено для автоматизации рутинных действий. Зачастую рутинность заключается в многократном повторении набора действий либо вообще без изменений, либо с минимальными изменениями. Для этих целей предназначены циклы.

В языках программирования обычно существует три вида циклов. Помимо встречавшихся нам ранее циклов с предусловием и постусловием выделяют т.н. цикл со счетчиком. Технически это цикл с предусловием, управление которым осуществляется при помощи специальной переменной-счетчика. Такой цикл обозначается ключевым словом **for**.

### Цикл for

В Python цикл for выполняет роль цикла, называющегося в других языках foreach или range-based-for (for для диапазонов), но на данный момент нас интересует его функциональность как цикла со счетчиком. Эта функциональность обеспечивается при помощи генератора range().

Блок-схема цикла for:



Мы видим, что данная блок-схема содержит в себе цикл с предусловием, дополненный действиями, связанными с управлением переменной-счетчика. Этот счетчик играет как минимум две роли в цикле `for`:

1. Считает сколько раз выполнилось тело цикла, и как следствие – позволяет ограничить количество повторений цикла заданным числом раз.
2. Позволяет телу цикла узнать на каком повторении сейчас находится цикл, и в связи с этим внести небольшие изменения в повторяемые действия.

В Python цикл `for` со счетчиком имеет следующую структуру:

```
for переменная in range (параметры) :  
    тело
```

Здесь *переменная* это то, на что мы выше ссылались как переменная-счетчик. Ее не требуется объявлять предварительно, если она не существовала до этого, то в цикле `for` она будет объявлена.

*Параметры* управляют способом изменения переменной-счетчика. Все параметры – целые числа и перечисляются через запятую. Их может быть один, два или три.

Если указан только один параметр, он означает количество повторений цикла. При этом переменная-счетчик изменяется от 0 до этого значения не включительно.

Если указано два параметра, первый означает начальное значение переменной-счетчика, а второй – конечное значение не включительно.

Если указано три параметра, первые два значат то же, что и в предыдущем случае, а третий указывает значение шага: на сколько изменяется значение переменной счетчика после каждого повторения.

Программисты считают с нуля, поэтому если не указано иное, следует предпочитать циклы с заданием начального значения переменной-счетчика, равного нулю.

Переменную-счетчик можно назвать как угодно. Здесь применимы те же правила именования переменных, что и везде. Тем не менее, наиболее распространенные имена переменных-счетчиков это `i`, `j`, `k`.

Считается плохим тоном использовать значение переменной-счетчика за пределами цикла, в котором она использовалась.

По этой причине можно переиспользовать имя переменной счетчика, если циклы `for` идут друг за другом:

```
# первый цикл
for i in range(N):
    # действия
```

```
# второй цикл
for i in range(M):
    # действия
```

В случае же если циклы находятся внутри друг друга, нужно использовать переменную с другим именем:

```
for i in range(N):
    for j in range(M):
        # действия
```

Повторения цикла еще называют *итерациями*, а сам процесс повторения – *итерированием*.

### Примеры

Здесь представлены описания типовых действий, включающих в себя цикл `for`, на русском языке и запись этих действий на Python. В качестве действий везде указан вывод на экран значения переменной-счетчика, однако стоит помнить, что в блоке действий допустимы любые корректные алгоритмические конструкции языка программирования.

Задача: повторить <действия> 10 раз:

```
for i in range(10):
    print(i)
```

Задача: повторить <действия> 10 раз, начиная с 1:

```
for i in range(1, 11):
    print(i)
```

Задача: перебрать значения от -5 до 10 включительно и выполнить над ними <действия>:

```
for i in range(-5, 11):
    print(i)
```

Задача: перебрать значения от 10 до -5 не включительно и выполнить над ними <действия>:

```
for i in range(10, -5, -1):  
    print(i)
```

Задача: перебрать все нечетные значения от 1 до 10 и выполнить над ними <действия>:

```
for i in range(1, 10, 2):  
    print(i)
```