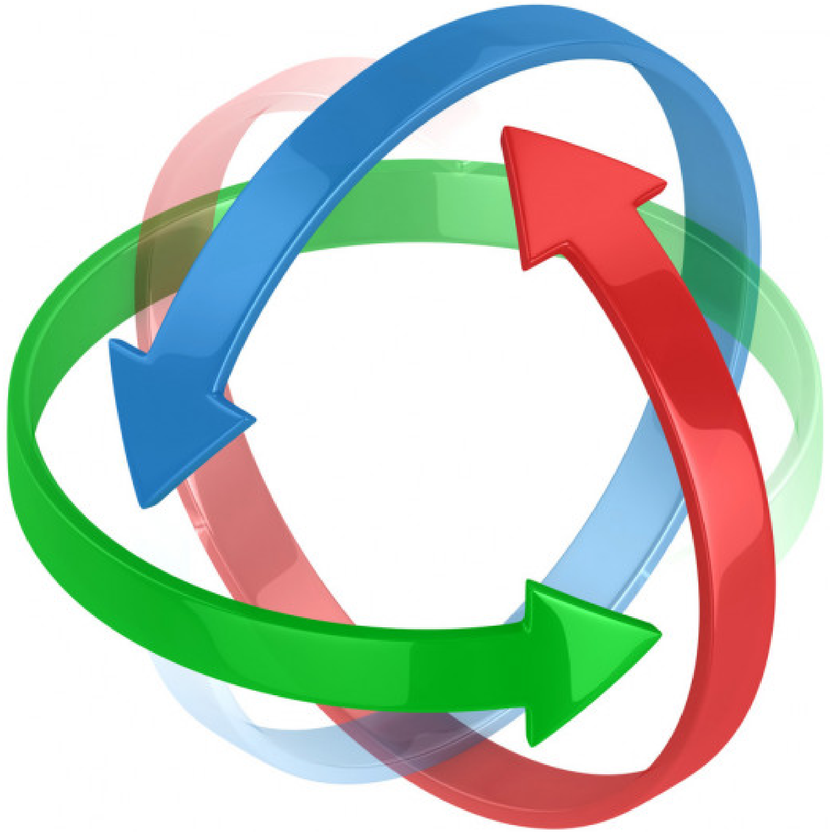
Python

7 урок. цикл For

Цикл For

Цикл while – это основной цикл в Python, но не единственный.

Существует также цикл for. Он перебирает все элементы объекта, который ему передали. Циклу for не нужен счётчик. Он сам знает, когда необходимо остановиться и закончить работу. Такая магия возможна, благодаря особому устройству объектов, по которым совершается итерация.



Циклы в Python

Посмотрим реализацию переворота строки через цил for:

В переменную char кладётся очередное значение на каждом шаге итерации. Имя этой переменной может быть любым.

```
string = input()
result = ''
for char in string:
    result = char + result
print(result)
#decode
#edoced
```

В определении цикла for нет ни счётчика ни условия выхода из него. Он не знает заранее, какое количество итераций будет совершено.

Функция range

Как правило, циклы for используются либо для повторения какой-либо последовательности действий заданное число раз, либо для изменения значения переменной в цикле от некоторого начального значения до некоторого конечного.

Для повторения цикла некоторое заданное число раз n можно использовать цикл for вместе с функцией range:

```
for i in range(4):  # равносильно инструкции for i in 0, 1, 2, 3:
    # здесь можно выполнять циклические действия
    print(i)
    print(i ** 2)
# цикл закончился, поскольку закончился блок с отступом
print('Конец цикла')
```

В качестве n может использоваться числовая константа, переменная или произвольное арифметическое выражение (например, 2 ** 10). Если значение n равно нулю или отрицательное, то тело цикла не выполнится ни разу.

Функция range

Функция range может также принимать не один, а два параметра. Вызов range(a, b) означает, что индексная переменная будеть принимать значения от a до b - 1, то есть первый параметр функции range, вызываемой с двумя параметрами, задает начальное значение индексной переменной, а второй параметр — первое значение, которое индексная переменная принимать не будет. Если же a≥b, то цикл не будет выполнен ни разу. Например, для того, чтобы просуммировать значения чисел от 1 до n можно воспользоваться следующей программой:

```
sum = 0
n = 5
for i in range(1, n + 1):
    sum += i
print(sum)
```

В этом примере переменная i принимает значения 1, 2, ..., n, и значение переменной sum последовательно увеличивается на указанные значения.

Функция range

Наконец, чтобы организовать цикл, в котором индексная переменная будет уменьшаться, необходимо использовать функцию range с тремя параметрами. Первый параметр задает начальное значение индексной переменной, второй параметр — значение, до которого будет изменяться индексная переменная (не включая его!), а третий параметр — величину изменения индексной переменной. Например, сделать цикл по всем нечетным числам от 1 до 99 можно при помощи функции range(1, 100, 2), а сделать цикл по всем числам от 100 до 1 можно при помощи range(100, 0, -1).

Более формально, цикл for i in range(a, b, d) при d > 0 задает значения индексной переменной i = a, i = a + d, i = a + 2 * d и так для всех значений, для которых i < b. Если же d < 0, то переменная цикла принимает все значения i > b.

range(start, stop, step):

Настройка функции print()

По умолчанию функция print() принимает несколько аргументов, выводит их через пробел, после чего ставит перевод строки. Это поведение можно изменить, используя именованные параметры sep (разделитель) и end (окончание).

```
print(1, 2, 3)
print(4, 5, 6)
print(1, 2, 3, sep=', ', end='. ')
print(4, 5, 6, sep=', ', end='. ')
print()
print(1, 2, 3, sep='', end=' -- ')
print(4, 5, 6, sep=' * ', end='.')
```

Результат:

```
1 2 3
4 5 6
1, 2, 3. 4, 5, 6.
123 -- 4 * 5 * 6.
```