

В цикле `for` указывается переменная и множество значений, по которому будет она будет пробегать. Множество значений может быть задано перечислением (забегая вперед, отметим, что такой объект будет являться списком/кортежем), диапазоном или строкой.

Например:

```
i = 1
for color in 'red', 'orange', 'yellow', 'green', 'cyan', 'blue', 'violet':

    print(i, '-th color of rainbow is ', color, sep='')
    i += 1
```

В этом примере переменная `color` последовательно принимает значения `'red'`, `'orange'` и т.д. В теле цикла выводится сообщение, которое содержит название цвета, то есть значение переменной `color`, а также номер итерации цикла число, которое сначала равно 1, а потом увеличивается на один (инструкцией `i += 1` с каждым проходом цикла).

В списке значений могут быть выражения различных типов, например:

```
for i in 1, 2, 'one', 'two', 'three':
    print(i)
```

При первых двух итерациях цикла переменная `i` будет принимать значение типа `int`, при последующих трёх — типа `str`.

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(n):
    Тело 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, иначе цикл выполнится `n` раз, а переменная `i` примет значения от 0 до `n-1`.

Если задать цикл таким образом:

```
for i in range(a, b):
    Тело цикла
```

то индексная переменная `i` будет принимать значения от `a` до `b - 1`, то есть первый параметр функции `range`, вызываемой с двумя параметрами, задаёт начальное значение индексной переменной, а второй параметр — значение, которая индексная переменная принимать не будет (но к которому должна приблизиться). 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
for i in range(1, n + 1):
    sum += i
```

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

Наконец, чтобы организовать цикл, в котором индексная переменная будет уменьшаться, необходимо использовать функцию `range` с тремя параметрами. Первый параметр задаёт 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 нечётным 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)`. В случае, если шаг отрицательный, пределы также должны идти от большего к меньшему.