

Язык программирования

Python

09 Цикл while



Понятие цикла

Цикл — это последовательность действий, выполняемая несколько раз подряд.

Цикл состоит из заголовка и тела цикла.

Тело цикла — последовательность операторов, которая будет выполняться несколько раз подряд.

Каждое отдельное выполнение тела цикла называют итерацией.

Говорят «цикл выполняется за такое-то количество итераций».

Количество итераций может быть небольшим (2-3 итерации), может быть огромным, а существуют и такие циклы, которые выполняются бесконечное число раз (они останавливаются операторами `break` или иным способом).

Понятие цикла

В Python циклы представлены операторами `while` и `for`.

Рассмотрим оператор `while`. В заголовке этого оператора, после ключевого слова `while`, указывается условие. Это условие, как и в случае с оператором `if`, представляет собой логическое выражение.

Это логическое выражение вычисляется перед выполнением каждой итерации цикла. Если результат выполнения такого выражения ложный, то выполнение цикла будет завершено.

Обычно, от итерации к итерации должно меняться какое-то значение, влияющее на условие.

Оператор while

Синтаксис оператора while выглядит следующим образом:

```
while condition:  
    statement1  
    ...  
    statementN
```

После ключевого слова while указывается условие (condition), затем двоеточие, а дальше блок инструкций, который будет выполняться, пока condition возвращает True.

Всегда говори «ДА»

```
answer = input()

while answer == 'y':
    print('Вы ответили Да')
    answer = input()
print('Конец программы')
```

В данном случае цикл `while` будет выполняться до тех пор, пока вы не введете, что-то что не `'y'` (от англ. yes).

Заметим, что `print('Конец программы')` здесь не входит в тело цикла, так как не отделен отступами.

Также заметим, что нам неизвестно количество итераций, которое выполнит программа. Здесь всё зависит от пользовательского ввода.

Оператор break

Оператор `break` позволяет нам досрочно выйти из цикла.

```
number = 0
while number < 10:
    if number == 7:
        break
    print(number, end = '')
    number += 1
```

Вывод:

0123456

Оператор continue

Оператор `continue` позволяет нам перейти к началу следующей итерации, минуя оставшееся тело цикла.

```
number = 0
while number < 10:
    if number == 7:
        continue
    print(number, end = '')
    number += 1
```

Вывод:

012345689

Ключевое слово else

Ключевое слово `else`, стоящее за телом цикла, позволяет нам выполнить блок инструкций после завершения цикла «естественным» путем. При завершении цикла оператором `break` блок операторов внутри `else` выполняться не будет.

```
number = 0
while number < 5:
    if number == 7:
        break
    print(number, end = '')
    number += 1
else:
    print('ok')
```

Вывод:

01234

ok

Оператор pass

Оператор `pass` не делает ничего. Он может использоваться, когда оператор требуется синтаксически, но программа не требует действий.

```
while True:
```

```
    pass # Ничего не делает, прервется выходом из программы
```

Перебор элементов кортежа

```
days = ('Monday', 'Tuesday', 'Wednesday', 'Thursday',  
'Friday', 'Saturday', 'Sunday', )
```

```
i = 0  
while i < len(days):  
    print(days[i], end = ' ')  
    i += 1
```

Вывод

Monday Tuesday Wednesday Thursday Friday Saturday Sunday

Итоги

Мы познакомились с такими понятиями, как цикл, заголовок цикла, тело цикла, итерация, оператор `while`, оператор `break`, оператор `continue`, ключевое слово `else`, оператор `pass`,

а также попрактиковались в работе с интерпретатором в файловом режиме.