

Python

3

Оператор цикла

Цикл

Цикл в языке программирования Python предназначен для перебора элементов структур данных и некоторых других объектов.

Что значит перебор элементов? Например, у нас есть список, состоящий из ряда элементов. Сначала берем из него первый элемент, затем второй, потом третий и так далее. С каждым элементом мы выполняем одни и те же действия в теле

Цикл `while`

```
while условие_выражение:  
    инструкции
```

Пример

```
choice = "y"
```

```
while choice.lower() == "y":
```

```
    print("Привет")
```

```
    choice = input("Для продолжения нажмите Y, а для  
выхода любую другую клавишу: ")
```

```
print("Работа программы завешена")
```

Пример 2

```
number = int(input("Введите число: "))
i = 1
factorial = 1
while i <= number:
    factorial *= i
    i += 1
print("Факториал числа", number, "равен", factorial)
```

Цикл for

Другой тип циклов представляет конструкция **for**. Цикл for вызывается для каждого числа в некоторой коллекции чисел. Коллекция чисел создается с помощью функции **range()**. Формальное определение цикла for:

```
for element in list:  
    инструкции
```

Пример

```
for element in range(0,10):  
    print(element)
```


Цикл FOR

```
>>> spisok = [10, 40, 20, 30]
```

```
>>> for element in spisok:
```

```
...     print(element + 2)
```

```
...
```

```
12
```

```
42
```

```
22
```

```
32
```

Пример - увеличение на 2

```
>>> i = 0
```

```
>>> for element in spisok:
```

```
...     spisok[i] = element + 2
```

```
...     i += 1
```

```
>>> spisok
```

```
[12, 42, 22, 32]
```

break

Оператор `break` досрочно прерывает выполнение цикла

```
a = 5
while a > 0:
    if a % 3 == 0:
        break
    print(a)
    a -= 1
```

continue

Оператор `continue` начинает следующий проход цикла, минуя оставшееся тело цикла

```
a = 5
while a > 0:
    if a % 3 == 0:
        a -= 1
        continue
    print(a)
    a -= 1
```

Цикл с параметром

```
n = 15
for i in range(n):
    print(i**2)
```

```
n = 15
for i in range(start, stop, step):
    блок кода
```

Вложенные циклы

```
for i in [[1, 2, 3],  
          [4, 5, 6],  
          [7, 8, 9]]:  
    for j in i:  
        print(j)
```

Цикл перебора элементов последовательности

```
animal_list = ['cat', 'dog']  
for animal in animal_list:  
    print(animal)
```

```
animal_list = ['cat', 'dog']  
for x in коллекция:  
    блок кода
```

```
a_list = [23, 42]  
for elem in a_list:  
    print(elem)
```

enumerate

```
for i, elem in enumerate(['a', 'b', 'c', 'd']):  
    print(f'{i} - {elem}')
```


Цикл по словарю

```
my_dict = {'Ivanov': 45, 'Petrov': 37, 'Sidorov': 29}
for key, value in my_dict.items():
    print(key, value)
```

Оператор `else` в цикле `for`

В языке **Python** можно использовать оператор `else` вместе с циклами.

- если оператор `else` используется с циклом `for` — он будет выполнен, когда список для цикла будет завершен;
- при использовании `else` вместе с циклом `while` — он будет использован, когда условие примет значение ложь (`false`).

```
>>> for i in 'hello world':  
...     if i == 'a':  
...         break  
...     else:  
...         print('Буквы а в строке нет')  
...
```

Буквы а в строке нет