

### 3-4 Операторы организации циклической обработки. Циклы.

<https://pythonworld.ru/osnovy/cikly-for-i-while-operator-break-i-continue-volshebnoe-slovo-else.html>

**Цикл** — разновидность управляющей конструкции в высокоуровневых языках программирования, предназначенная для организации многократного исполнения набора инструкций.

Также циклом может считать любую многократно исполняемую последовательность инструкций, организованная каким-либо способом (например, с помощью условного перехода).

#### Виды циклов:

##### 1. Безусловные циклы

Выход из них не предусмотрен логикой программы. Их же можно назвать бесконечными. Специальных синтаксических средств для создания бесконечных циклов, ввиду их нетипичности, языки программирования не предусматривают, поэтому такие циклы создаются с помощью конструкций, предназначенных для создания обычных (или условных) циклов.

##### 2. Цикл с предусловием

Он выполняется, пока истинно некоторое условие, указанное перед его началом.

Пример:

```
i = 6
while i < 12: # пока i меньше 12
    print(i)
    i = i + 2 # i увеличить на 2
```

Результат:

```
6
8
10
```

*While* - один из самых универсальных циклов.. Выполняет тело цикла до тех пор, пока условие цикла истинно.

### 3. Цикл с постусловием

Условие проверяется после выполнения тела цикла. Отсюда следует, что тело всегда выполняется хотя бы один раз.

Пример:

```
i = 6
while True:
    i = i + 2 # i увеличить на 2
    print(i)
    if i > 12:
        break # прервать цикл
```

Результат:

```
8
10
12
14
```

### 3. Цикл for

Цикл *for* уже чуточку сложнее, чуть менее универсальный, но выполняется гораздо быстрее цикла *while*. Этот цикл проходится по любому итерируемому объекту (например строке или списку), и во время каждого прохода выполняет тело цикла.

Пример:

```
for i in 'Привет Мир':
    print(i * 2, end='_')
```

Результат:

```
ПП_рр_ии_вв_ее_тт_ _ММ_ии_рр_
```

### 4. Оператор Continue

Оператор *continue* начинает следующий проход цикла, минуя оставшееся тело цикла (*for* или *while*)

Пример:

```
for i in 'Привет Мир':
    if i == 'и':
        continue
    print(i * 2, end='_')
```

Результат:

```
ПП_рр_вв_ее_тт_ _ММ_рр_
```

## 5. Циклы с прерыванием. *Break*, *Else*

Оператор *break* досрочно прерывает цикл.

Пример:

```
for i in 'Привет Мир':  
    if i == 'и':  
        break  
    print(i * 2, end=' _')
```

Результат:

ПП\_пп\_

Слово *else*, примененное в цикле *for* или *while*, проверяет, был ли произведен выход из цикла инструкцией *break*, или же "естественным" образом. Блок инструкций внутри *else* выполнится только в том случае, если выход из цикла произошёл без помощи *break*.

Пример:

```
for i in 'Привет Мир':  
    if i == 'Б':  
        break  
else:  
    print('Буквы Б в строке нет')
```

Результат:

Буквы Б в строке нет