

Самое важное

Оператор while

Новый оператор позволяет выполнять выбранную часть кода повторно нужное количество раз. Это избавит вас от необходимости дублировать код и сделает его лаконичнее.

Синтаксис

while <условие>:

 блок кода

1. Пока выполняется условие, прописанное после while, блок кода внутри цикла будет повторяться.
2. Чтобы Python понял, какую часть кода повторять в цикле, а какую выполнять после цикла, используйте отступы:

```
while x < 10:
```

```
    x += 1 — этот код будет повторяться в цикле;
```

```
    x -= 1 — этот код будет выполнен после цикла.
```

3. При необходимости код можно прервать без выполнения условия:

```
while x < 10:
```

```
    if x == 5:
```

```
        break — эта команда прервёт выполнение цикла.
```

```
    x += 1
```

```
    x -= 1 — после прерывания будет выполнен код после цикла.
```

Если конкретное условие задать сложно, то можно использовать объект True.

Пример 1

while True: — это будет бесконечный цикл, который можно будет прервать оператором break.

Пример 2

```
x = True
```

```
while x:
```

```
    if y > 9 or user_input == 'exit':
```

```
        x = False — изменив значение переменной на False, вы остановите работу цикла.
```

Помимо break, вы также можете использовать **оператор continue**.

В отличие от break, continue не прерывает выполнение цикла, **но** прерывает выполнение текущей итерации и немедленно начинает следующую.

```
while True:
    user_input = input('Введите команду')
    if user_input == 'next':
        continue
    print('Выбрана команда - ', user_input)
```

Если пользователь введет слово 'next' в input, то сработает проверка и выполнится команда continue, принт после условного блока будет пропущен, выполнится проверка условия цикла. Если она равна True, начнётся следующая итерация.

Не допускай следующих ошибок!

Не забывайте, что условие цикла проверяется **после** выполнения **всей** итерации.

```
x = True
while x:
    if y > 9 or user_input == 'exit':
        x = False — когда вы заменили x на False, цикл не прервётся сразу.
    print('Hello, user!') — сперва он закончит остальной код в этой итерации.
```

И только после того, как текущая итерация будет завершена, сработает проверка и цикл будет остановлен.

Если же вам нужно немедленно прекратить выполнение цикла, используйте break