



# Цикл **while**

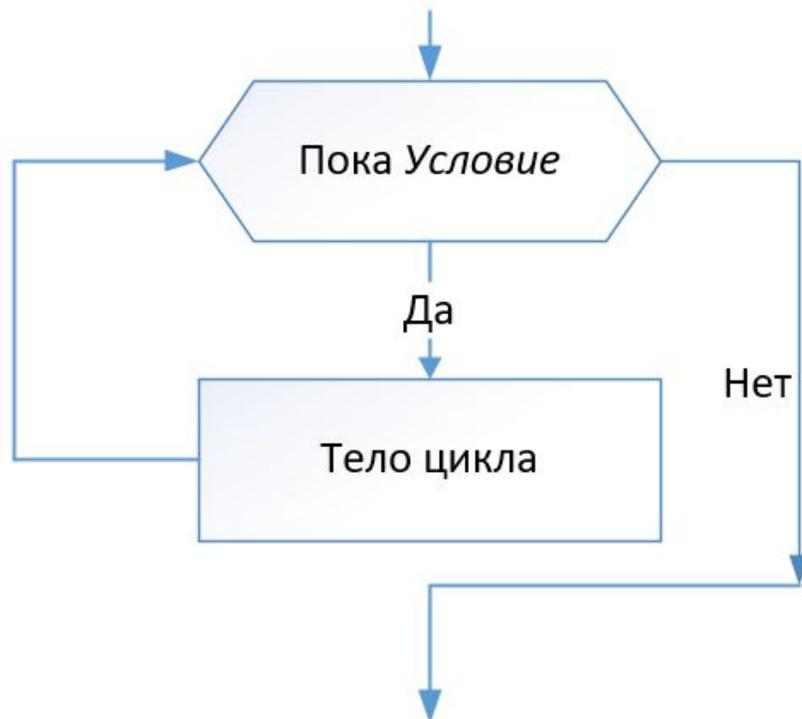
---

PYTHON



# Если надо повторять действия

Цикл – многократное повторение одних и тех же действий.



**НЦ Пока Условие**

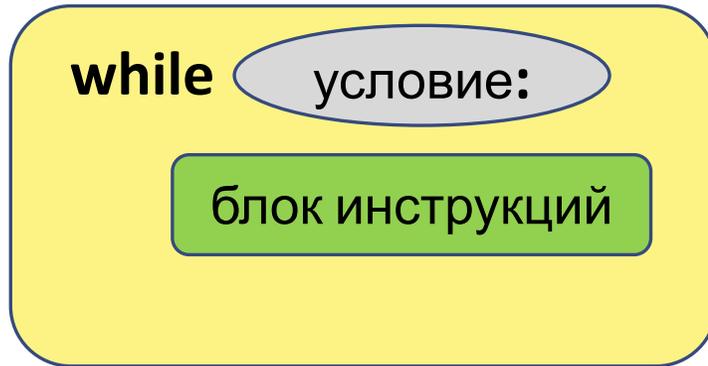
ТелоЦикла

**КЦ**



# while - цикл с предусловием

---



Пока **условие истинно**, выполняем блок инструкций.

**Сколько раз?**

**Неизвестно**

Пример:

```
i = 2
```

```
while i <= 10:
```

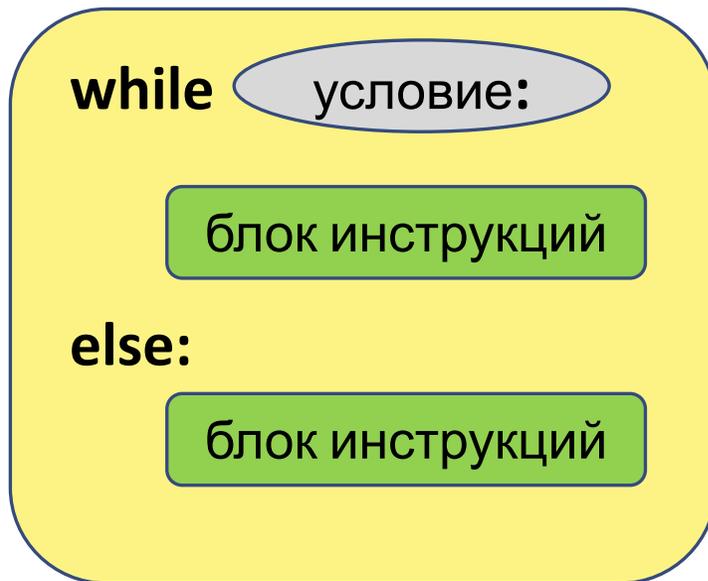
```
    print(i ** 2)
```

```
    i = i + 2
```



# else

После тела цикла можно написать команду **else:** и после него блок инструкций, который будет выполнен *один раз* после окончания цикла.



Пример:

```
i = 1
while i <= 10:
    print(i)
    i += 1
else:
    print('Цикл окончен, i =', i)
```

PS. Кажется, что в этом нет никакого смысла, ведь эту инструкцию можно было бы написать просто после цикла. Но смысл появляется вместе с командой **break**.



# break

---

**break** – досрочно прерывает цикл

Если во время исполнения цикла встречается команда **break**, то сразу прекращается выполнение цикла и происходит выход из цикла. При этом команда **else** не выполняется!

Пример:

```
a = int(input())
while a != 0:
    if a < 0:
        print('Встретилось отрицательное число', a)
        break
    a = int(input())
else:
    print('Ни одного отрицательного числа не
встретилось')
```



Слово **else**, применённое в цикле, проверяет, был ли произведён выход из цикла инструкцией **break**, или же "естественным" образом.



# continue

---

`continue` – продолжение цикла.

Если команда `continue` встречается в теле цикла, то все инструкции, следующие за ней, игнорируются и начинается новый проход цикла.

Пример:

```
j = 0
```

```
while j < 5:
```

```
    j += 1
```

```
    if j == 3:
```

```
        print('Пропускаем j == 3')
```

```
        continue
```

```
    print(j)
```





# Полезные замечания

---

Если инструкции **break** и `continue` содержатся внутри нескольких вложенных циклов, то они влияют лишь на исполнение самого внутреннего цикла.

Увлечение инструкциями `break` и `continue` не поощряется, если можно обойтись без их использования.

Если вы случайно попали в бесконечный цикл, нажмите сочетание клавиш `Ctrl + C`.



# Однолинейный цикл **while**

---

Цикл `while` может быть указан в одной строке.

```
n = 5
```

```
while n > 0: n -= 1
```

Если в блоке, составляющем тело цикла, есть несколько операторов, они могут быть разделены точкой с запятой (;)

```
n = 5
```

```
while n > 0: n -= 1; print(n)
```



# Цикл **while** с постусловием

В Python **цикл с постусловием отсутствует!**  
Но, если возникает необходимость исполнить тело цикла хотя бы один раз, то реализуется вот такая конструкция:

```
while True:  
    <тело цикла>  
if условие:  
    break
```

Задача. Обеспечить ввод **положительного** числа в переменную **n**.

```
while True:  
    print ( "Введите положительное число:" )  
    n = int ( input ( ) )  
    if n > 0: break
```

бесконечный  
цикл

тело цикла

условие  
выхода

прервать  
цикл

- при входе в цикл условие *не* проверяется
- цикл всегда выполняется хотя бы один раз