

Цикл While

План урока

- Цикл в программировании
- Синтаксис цикла while в python
- Операторы else, break и continue
- Вложенные циклы
- Примеры

ЦИКЛ

Блок повторяющегося
при определенных условиях кода

Задача

Приготовить 5 бургеров

ооо

приготовить бургер
приготовить бургер
приготовить бургер
приготовить бургер
приготовить бургер



ооо

Пока приготовлено < 5
бургеров:
 приготовить бургер



Цикл while

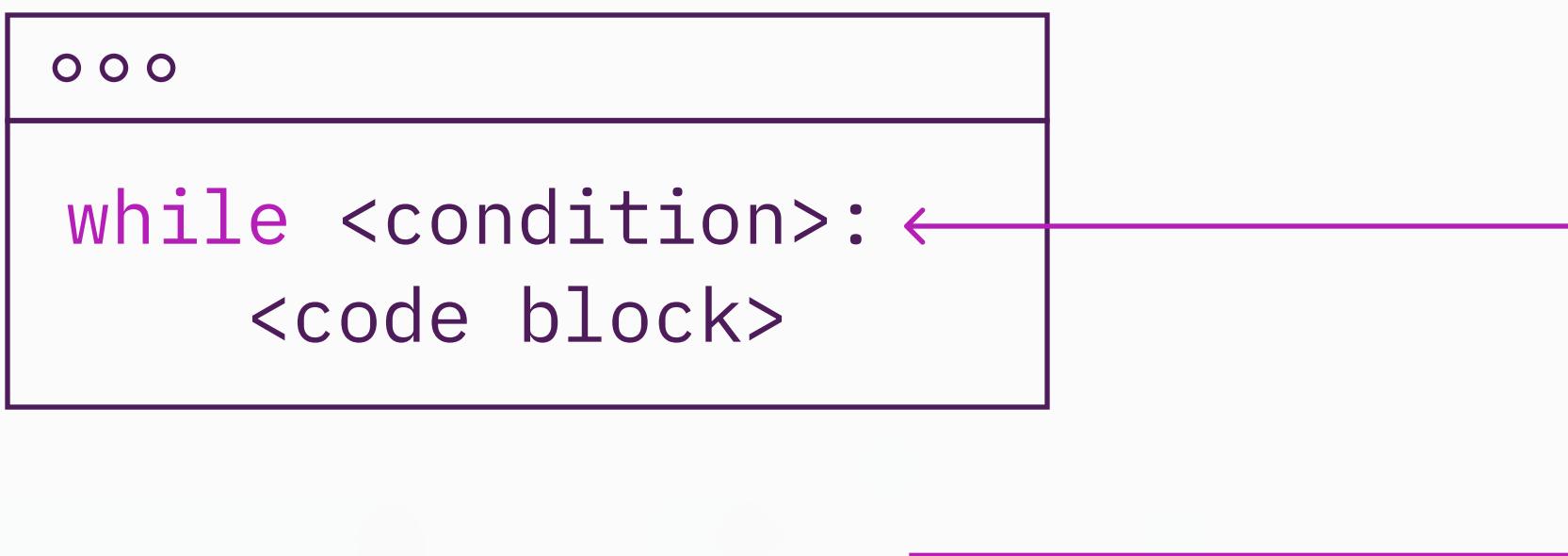
Конструкция цикла while

ооо

```
while <condition>:  
    <code block>
```

Цикл while

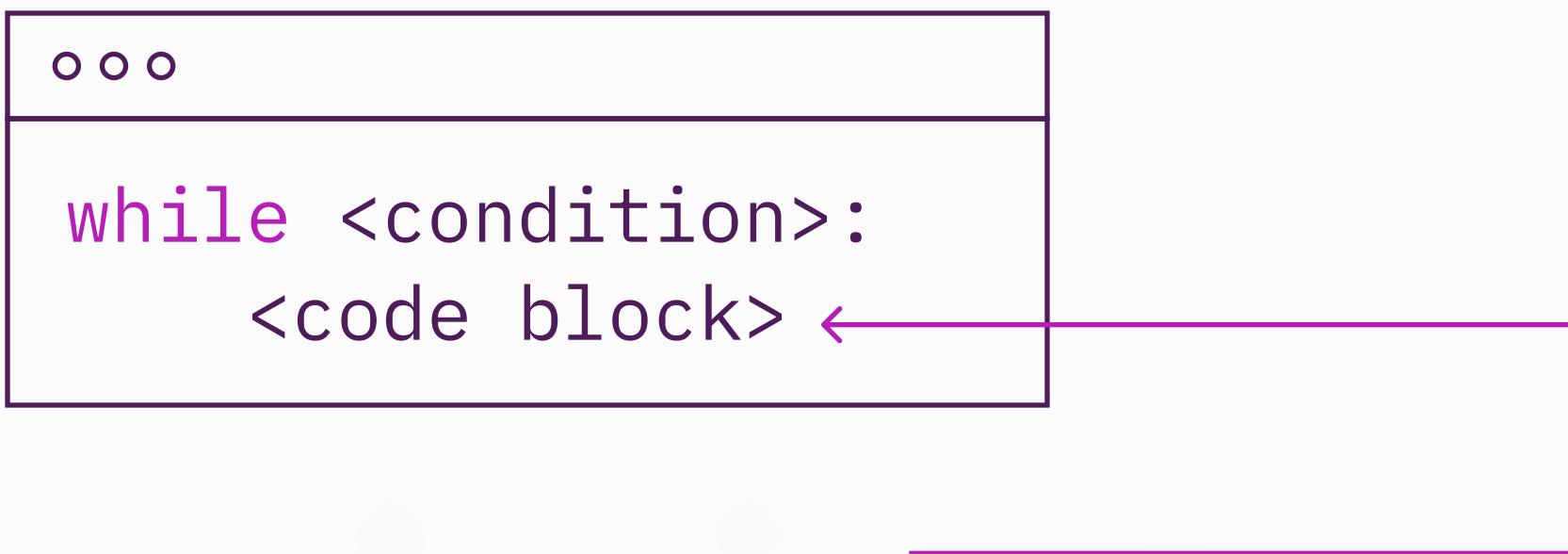
Конструкция цикла **while**



Условие выполнения блока
в цикле - логическое
выражение (тип bool)

Цикл while

Конструкция цикла `while`



Блок цикла - блок
повторяющегося кода.
Блок выполняется если
`condition == True`

Пример

Печатаем в консоль 4 символа #

ooo | код

```
sharps_to_print = 4
while sharps_to_print > 0:
    print("#")
    sharps_to_print -= 1
```

ooo | вывод

```
>>> #
>>> #
>>> #
>>> #
```

Пример

Печатаем в консоль 4 символа #

ooo | код

```
sharps_to_print = 4 ←  
while sharps_to_print > 0:  
    print("#")  
    sharps_to_print -= 1
```

ooo | вывод

```
>>> #  
>>> #  
>>> #  
>>> #
```

Заводим счетчик решеток, которые остается напечатать
Изначально он равен 4

Пример

Печатаем в консоль 4 символа #

ooo | код

```
sharps_to_print = 4
while sharps_to_print > 0: ←
    print("#")
    sharps_to_print -= 1
```

ooo | вывод

```
>>> #
>>> #
>>> #
>>> #
```

В цикле while мы проверяем условие, что счетчик решеток должен быть больше 0

Пример

Печатаем в консоль 4 символа #

ooo | код

```
sharps_to_print = 4
while sharps_to_print > 0:
    print("#") ←
    sharps_to_print -= 1
```

ooo | вывод

```
>>> #
>>> #
>>> #
>>> #
```

Печатаем решетку

Пример

Печатаем в консоль 4 символа #

ooo | код

```
sharps_to_print = 4
while sharps_to_print > 0:
    print("#")
    sharps_to_print -= 1 ←
```

ooo | вывод

```
>>> #
>>> #
>>> #
>>> #
```

Уменьшаем
счетчик
решеток на 1

ИТЕРАЦИЯ

Процесс исполнения блока
кода называется **итерацией**

Оператор `else`

Конструкция цикла `while` с `else`

```
...
while <condition>:  
    <code block>  
else:  
    <code block>
```

Данный блок выполняется
при невыполнении условия
`<condition>`, и цикл завершается

Блок выполняется только один
раз при невыполнении условия цикла

Оператор break

Конструкция цикла **while** с **break**

```
ooo  
while <condition>:  
    <code block>  
    if <inner condition>:  
        break ←  
    else:  
        <code block>
```

При вызове оператора
break цикл, внутри
которого он вызван
прекращается
мгновенно

При вызове оператора break
– блок else не выполняется

Оператор continue

Конструкция цикла **while** с **continue**

```
ooo  
while <condition>:  
    <code block>  
    if <inner condition>:  
        continue ←  
        <code block>  
    else:  
        <code block>
```

При вызове оператора `continue` итерация цикла мгновенно прекращается, после чего мы вновь возвращаемся к проверке условия `<condition>`

Цикл while

Вложенные циклы

ооо

```
while <condition>:  
    <code block>  
    while <inner condition>:  
        <inner code block>
```

Оператор **break**, вызванный во внутреннем цикле, прервет только **внешний** цикл

Оператор **continue**, вызванный во внутреннем цикле, прерывает итерацию только **внутреннего** цикла

Пример

Считывать целые числа из консоли
и выводить квадраты этих чисел
с новой строки до тех пор, пока
не будет введено число 0

ооо	ввод
>>> 4	
>>> -2	
>>> 3	
>>> 0	

ооо	вывод
>>> 16	
>>> 4	
>>> 9	

Пример

Считывать целые числа из консоли
и выводить квадраты этих чисел
с новой строки до тех пор, пока
не будет введено число 0

ooo | код

```
num = int(input())

while num != 0:
    print(num ** 2)
    num = int(input())
```

Пример

Считывать целые числа из консоли
и выводить квадраты этих чисел
с новой строки до тех пор, пока
не будет введено число 0

ооо	код
num = int(input()) while num != 0: print(num ** 2) num = int(input())	

Считываем число
с клавиатуры

Пример

Считывать целые числа из консоли
и выводить квадраты этих чисел
с новой строки до тех пор, пока
не будет введено число 0

ooo	код
<pre>num = int(input()) while num != 0: print(num ** 2) num = int(input())</pre>	

Запускаем цикл
while и, пока **num**
не равно нулю,
выводим квадрат
этого числа

Пример

Считывать целые числа из консоли
и выводить квадраты этих чисел
с новой строки до тех пор, пока
не будет введено число 0

ooo	код
<pre>num = int(input()) while num != 0: print(num ** 2) num = int(input())</pre>	

В конце блока
цикла мы
считываем
num снова

Пример

Считывать целые числа из консоли
и выводить квадраты этих чисел
с новой строки до тех пор, пока
не будет введено число 0

ooo

код

```
num = int(input())
while num != 0:
    print(num ** 2)
    num = int(input())
```

При выполнении цикла
важно обновлять условие

Иначе цикл будет **бесконечным**

Пример

Вывести в консоль таблицу
умножения размером $n \times n$

ооо	ввод
>>> 3	

ооо	вывод
>>> 1x1=1 1x2=2 1x3=3 >>> 2x1=2 2x2=4 2x3=6 >>> 3x1=3 3x2=6 3x3=9	

Пример

Вывести в консоль таблицу
умножения размером $n \times n$

ооо	ввод
<pre>n = int(input()) i = 1 while i <= n: j = 1 while j <= n: print(f"{i}x{j}={i*j}", end=" ") j += 1 print("") i += 1</pre>	

Пример

Вывести в консоль таблицу
умножения размером $n \times n$

ооо	ввод
<pre>n = int(input()) ← i = 1 while i <= n: j = 1 while j <= n: print(f"{i}x{j}={i*j}", end=" ") j += 1 print("") i += 1</pre>	

Вводим n с клавиатуры

Пример

Вывести в консоль таблицу
умножения размером $n \times n$

ооо	ввод
<pre>n = int(input()) i = 1 ← while i <= n: j = 1 ← while j <= n: print(f"{i}x{j}={i*j}", end=" ") j += 1 print("") i += 1</pre>	

Вводим счетчики i и j
 i - это номер строки,
 j - номер столбца

Пример

Вывести в консоль таблицу
умножения размером $n \times n$

ооо	ввод
<pre>n = int(input()) i = 1 while i <= n: j = 1 while j <= n: print(f"{i}{x}{j}={i*j}", end=" ") j += 1 print("") i += 1</pre>	



На пересечении i -ой строки и j -того столбца у нас будет выражение $i*j$, равное их произведению

Пример

Вывести в консоль таблицу
умножения размером $n \times n$

ооо	ввод
<pre>n = int(input()) i = 1 while i <= n: j = 1 while j <= n: print(f"{i}x{j}={i*j}", end=" ") j += 1 print("") i += 1</pre>	

Запускаем внешний цикл while
с условием $i \leq n$

Пример

Вывести в консоль таблицу
умножения размером $n \times n$

ооо	ввод
<pre>n = int(input()) i = 1 while i <= n: j = 1 while j <= n: print(f"{i}x{j}={i*j}", end=" ") j += 1 print("") i += 1</pre>	

Внутри этого цикла
введена переменная $j = 1$

Пример

Вывести в консоль таблицу
умножения размером $n \times n$

ооо	ввод
<pre>n = int(input()) i = 1 while i <= n: j = 1 while j <= n: ← print(f"{i}x{j}={i*j}", end=" ") j += 1 print("") i += 1</pre>	

Запускаем внутренний
цикл **while**, в котором
проверяем условие: $j \leq n$

Пример

Вывести в консоль таблицу
умножения размером $n \times n$

ооо	ввод
<pre>n = int(input()) i = 1 while i <= n: j = 1 while j <= n: print(f"{i}x{j}={i*j}", end=" ") j += 1 print("") i += 1</pre>	

Выводим результат

Пример

Вывести в консоль таблицу
умножения размером $n \times n$

ооо	ввод
<pre>n = int(input()) i = 1 while i <= n: j = 1 while j <= n: print(f"{i}x{j}={i*j}", end=" ") j += 1 print("") i += 1</pre>	



Печатаем все
произведения
через пробел

Пример

Вывести в консоль таблицу
умножения размером $n \times n$

ооо	ввод
<pre>n = int(input()) i = 1 while i <= n: j = 1 while j <= n: print(f"{i}x{j}={i*j}", end=" ") j += 1 print("") i += 1 ←</pre>	

По завершении внутреннего цикла
печатаем символ перехода на другую
строку и прибавляем единицу к i