

ОПЕРАТОР ЦИКЛА WHILE

ОТП, 6 КЛАСС, УРОК 2

ЧТО ТАКОЕ ЦИКЛ?

При написании кода программы может возникнуть необходимость выполнения одних и тех же действий несколько раз. Для этих целей используются циклы.

Циклы делятся изначально на 2 типа:

1. Определенный цикл – это цикл, в котором заранее известно количество повторений действий.
2. Неопределенный цикл – это цикл, в котором заранее не известно количество повторений действий, но известно условие окончания цикла.

Какой вид цикла вам хорошо известен?



ЦИКЛ WHILE

- Цикл **while** относится к неопределенным циклам, так как его можно использовать в тех случаях, когда заранее неизвестно количество повторений блока действий.
- Цикл **while** в Python выполняет блок кода программы, пока условие, указанное в цикле, имеет значение True(истина).

СИНТАКСИС ЦИКЛА WHILE

while <условие> :

<блок действий, которые повторяются при верном условии>

else:

<блок действий, если условие неверное>

Условия в цикле строятся по тому же принципу, что рассматриваются в алгоритме ветвления.

ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ ЦИКЛА WHILE

1. Переменным присваиваются начальные значения.
2. Проверяется условие, и, если оно истинно, выполняется блок кода внутри цикла, иначе выполнение цикла завершается.
3. Переменные изменяются по требованию задачи.
4. Переход к пункту 2.
5. Если внутри цикла не использовался оператор `break`, то после завершения выполнения цикла будет выполнен блок кода после инструкции `else`. Этот блок не является обязательным.

ПРИМЕР: ВЫВЕСТИ В СТОЛБИК ЧИСЛА ОТ 1 ДО 5

```
count = 1
```

```
while count <= 5:
```

```
    print(count)
```

```
    count += 1
```

Результат:

1

2

3

4

5

```
n=int(input('Введите число '))
s=0
while n!=0:
    s+=n
    n=int(input('Введите следующее число '))
print('Сумма введенных чисел = ', s)
```

IDLE Shell 3.8.10

File Edit Shell Debug Options Window Help

Python 3.8.10 (tags/v3.8.10:3d8993a, May 3 2021, 11:48:03) [MSC
AMD64] on win32

Type "help", "copyright", "credits" or "license()" for more info

>>>

===== RESTART: C:\Users\Даниил\Desktop\789.py =====

Введите число 4

Введите следующее число 12

Введите следующее число 32

Введите следующее число 4

Введите следующее число 14

Введите следующее число 34

Введите следующее число 0

Сумма введенных чисел = 100

>>> |

ПОСМОТРИТЕ НА КОД. СФОРМУЛИРУЙТЕ УСЛОВИЕ
ЗАДАЧИ

<code>n=int (input('Введите число '))</code>	Вводим первое число
<code>s = 0</code>	Заводим переменную для хранения результата
<code>while n != 0:</code>	Записываем цикл с условием
<code> s += n</code>	<i>В цикле:</i> к сумме прибавляем число
<code> n = int(input('Введите число '))</code>	<i>В цикле:</i> вводим в цикле следующее число
<code>print ('сумма =', s)</code>	После ввода 0 цикл завершает свою работу. Выводим на экран ответ

НАЙТИ КОЛИЧЕСТВО ИЛИ ПРОИЗВЕДЕНИЕ ЧИСЕЛ. ЧТО ИЗМЕНИТСЯ В КОДЕ?

НАЙТИ СУММУ, КОЛИЧЕСТВО, ПРОИЗВЕДЕНИЕ ЧИСЕЛ

```
n=int (input('Введите число '))  
s = 0  
while n != 0:  
    s += n  
    n = int(input('Введите число '))  
print ('сумма =', s)
```

```
n=int (input('Введите число'))  
k = 0  
while n != 0:  
    k += 1  
    n = int(input('Введите число '))  
print ('количество =', k)
```

```
n=int (input('Введите число '))  
p = 1  
while n != 0:  
    p *= n  
    n = int(input('Введите число '))  
print ('произведение =', p)
```

КАК НАЙТИ СУММУ ЧИСЕЛ, КРАТНЫХ 5? ЧТО ИЗМЕНИТСЯ В ПРОГРАММЕ?

```
n=int (input('Введите число '))
```

```
s = 0
```

```
while n != 0:
```

```
    s += n
```

```
    n = int(input('Введите число '))
```

```
print ('сумма =', s)
```

НАЙТИ СУММУ ЧИСЕЛ, КРАТНЫХ 5.

```
n=int (input('Введите число '))
```

```
s = 0
```

```
while n != 0:
```

```
    if n % 5 == 0:
```

```
        s += n
```

```
    n = int(input('Введите число '))
```

```
print ('сумма =', s)
```

РЕШЕНИЕ ЗАДАЧ

С КЛАВИАТУРЫ ВВОДИТСЯ ПОСЛЕДОВАТЕЛЬНОСТЬ НАТУРАЛЬНЫХ ЧИСЕЛ. ПРИЗНАКОМ ОКОНЧАНИЯ ВВОДА ЯВЛЯЕТСЯ ЧИСЛО 0. ПРОАНАЛИЗИРОВАТЬ ВВЕДЕННУЮ

ПОСЛЕДОВАТЕЛЬНОСТЬ.

- 1. Найти сумму четных чисел.
 - 2. Найти произведение чисел, кратных 7.
 - 3. Найти количество чисел, оканчивающихся на 4.
 - 4. Найти сумму чисел оканчивающихся на 0.
 - 5. Найти количество нечетных чисел.
 - 6. Найти произведение однозначных чисел.
 - 7. Найти сумму двузначных чисел.
 - 8. Найти произведение четных чисел, кратных 3.
 - 9. Найти количество чисел, кратных 6 и оканчивающихся на 4.
 - 10. Найти сумму двузначных чисел, состоящих из одинаковых цифр (11, 22 и т.п.).
- 