

# Программирование циклов.

## Оператор for

Болгова Н.А.

МБОУ СОШ с углубленным изучением  
отдельных предметов с.Тербуны  
Тербунского муниципального района Липецкой  
области

# Повторяем!

- Цикл – многократное повторение последовательности действий
- Повторяющаяся последовательность действий – тело цикла
- Типы циклов: с предусловием («Пока»), постусловием («До»), параметром («Для»)

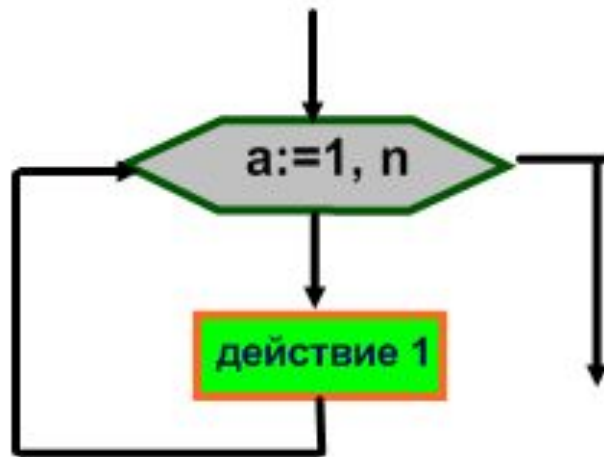
# Цикл с условием («Пока»)



нц пока УСЛОВИЕ  
действие 1  
кц

**while** <условие>:  
    **действие 1**

# Цикл с параметром («Для»)



НЦ для а от 1 до n  
действие 1

КЦ

а –

параметр/счетчик

# Отличие циклов:

- Цикл «Пока» выполняется неопределенное количество раз, пока условие верно (истинно)
- Цикл «Для» выполняется определенное количество раз (n-раз)

Тема урока

**Программирование  
циклов.  
Оператор for**

# Цикл с параметром

**for** <параметр> in range(n):

<4 пробела> **тело цикла**

**for** – для

# Цикл с параметром

**for** **a** in range(n):

**тело цикла**

(цикл выполняется **для a от 0 до n-1**,  
где **n** – количество раз выполнения,  
т.е. **a = 0,1,2,3,....., n-1**) ,т.е выполняется **с нуля**)



# Циклы с параметром (варианты)

**for  $i$  in range (1,  $n+1$ ):**

**<4 пробела> тело цикла**

(Для  $i$  от 1 до  $n$ , где  $n$  – кол-во раз  
выполнения,

т.е.  $i = 1, 2, 3, \dots, n-1, n$

Цикл выполняется на 1 раз меньше)

# Циклы с параметром (варианты)

**for *i* in range (a, b, d):**

**<4 пробела> тело цикла**

- (Для *i* от *a* до *b* с шагом *d*)

т.е.  $i = a, a+d, a+2d, \dots, b$ )

- Можно организовать цикл с конца, тогда шаг  $d = -1$

# Циклы с параметром (варианты)

**for** **i** in range (n):

<4 пробела> **if** <условие>:

<4 пробела> **действие 1**

**else:**

<4 пробела> **действие 2**

# Примеры задач

# 1. Дано натуральное число. Определите, является ли число простым, если нет, то вывести все его делители.

- Формат ввода: одно число
- Формат вывода:  
В первой строке — все делители введенного числа,  
Во второй — сообщение «ПРОСТОЕ»  
записанные через пробел (перед первым и  
после последнего делителя пробелов нет). »  
или «НЕТ».

Пример

Ввод	Вывод
12	1 2 3 4 6 12 НЕТ

```
a = int(input())
c = 0
for i in range(1, a + 2):
    if a % i == 0:
        print(i, end=' ')
        c += 1
print(' ')
if c == 2:
    print('ПРОСТОЕ')
else:
    print('НЕТ')
```

Пояснение:

- с – счетчик делителей
- выполняется от 1, т.к на «0» делить нельзя до a + 1
- если a кратно i, то выводим i (делитель)
- считаем кол-во делителей
- вывод пустой строки
- если c = 2,
- то вывод «Простое»
- иначе
- вывод «Нет»

# Проверка:

```
простое.py - C:/Users/17/Desktop/простое.py (3.6.2)
File Edit Format Run Options Window Help

a = int(input())
c = 0
for i in range(1, a + 2):
    if a % i == 0:
        print(i, end=' ')
        c += 1
print(' ')
if c == 2:
    print('ПРОСТОЕ')
else:
    print('HET')
```

```
Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
Python 3.6.2 (v3.6.2:5fd33b
on win32
Type "copyright", "credits"
>>>
===== RESTART:
12
1 2 3 4 6 12
HET
>>>
```

## 2. Найти произведение натуральных нечетных чисел последовательности

$$P = 1 * 3 * 5 * \dots * n$$

- Формат ввода: одно число
- Формат вывода:  
Произведение четных чисел.

Пример

Ввод

8

Вывод

105



Пояснение:

- $p$  – произведение

- Выполнение цикла от 1, т.к. на «0» делить нельзя до  $n + 1$

1

- если  $i$  некратно 2 ,

- то находим

произведение

```
n = int(input())
```

```
p = 1
```

```
for i in range(1, n + 1):
```

```
    if  $i \% 2 == 1$ :
```

```
        p *= i
```

```
print('Р четных = ', p)
```

2 вариант условия :  **$i \% 2 != 0$** :

### 3. Найти сумму всех чисел последовательности

$$s = 1 + 2 + 3 + \dots + n$$

- Формат ввода: одно число
- Формат вывода:  
Произведение четных чисел.

Пример

Ввод

8

Вывод

35

# Домашнее задание:

- Выучить оператор for (все варианты цикла)
- Составить программу к задаче 3

# Программа (проверка д.з)

```
n = int(input())
```

```
s = 0
```

```
for i in range(1, n + 1):
```

```
    s += i
```

```
print('S = ', s)
```

```
for i in range(n):
```

```
    s += i
```

```
print('S = ', s)
```

Цикл можно начать с нуля, т.к. он на сумму не влияет

# Литература:

- Поляков К.Ю., Еремин Е.А.  
«Информатика 10 класс (базовый и углубленный уровни)»- Москва, Бином, 2018)
- [informatics.mccme.ru](http://informatics.mccme.ru)
- ПИТОН тьютор
- Python 3.7.0