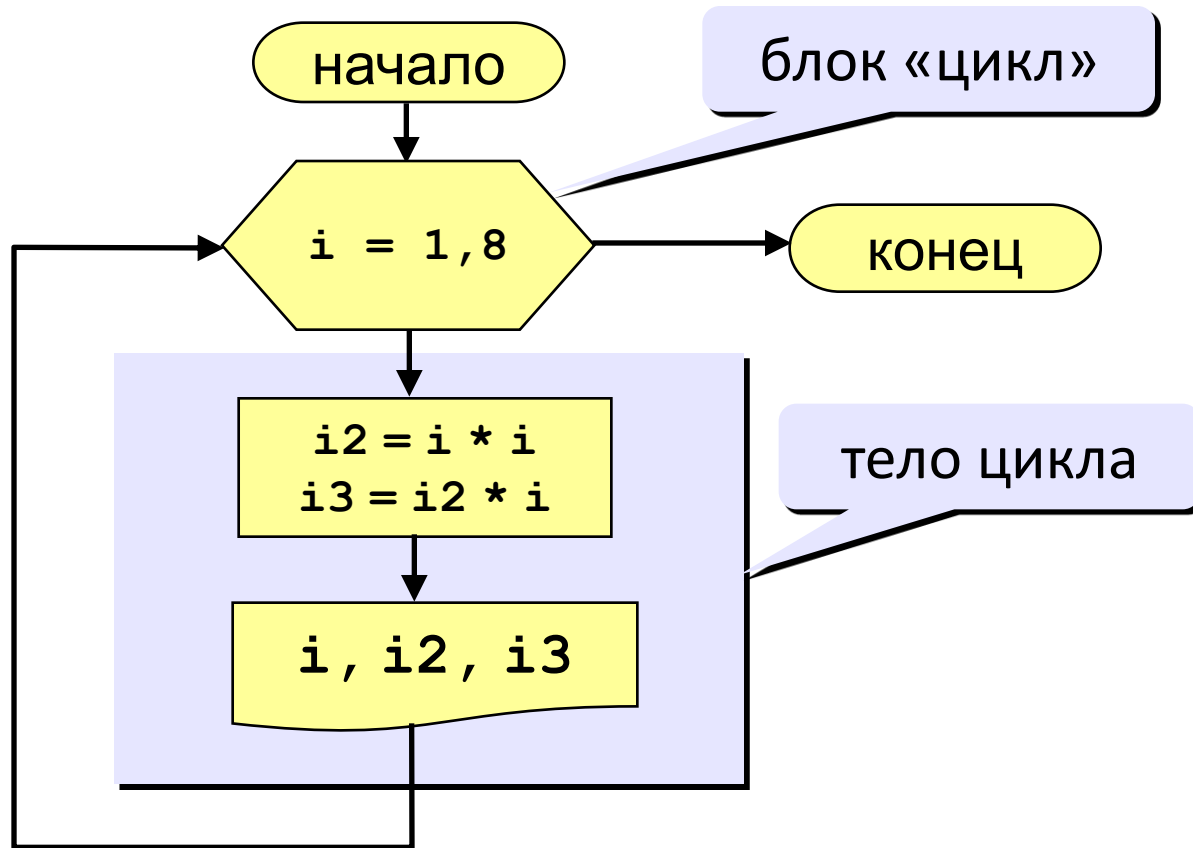


# Алгоритм (с блоком «цикл»)



for – «для»

in range – «в диапазоне» в переводе с английского

Общая форма записи:

```
for <переменная-счётчик> in range(<диапазон>) :  
    <действия>
```

Пример 1:

```
for i in range(5) :  
    print("Привет!")
```

Пример 2 (с  
начальным и  
конечным  
значением):

```
for i in range(3, 20) :  
    print(i)
```

Пример  
цикла FOR:

```
for i in range(5):  
    print("Привет!")
```

Переменная **i** принимает значения 0, 1, 2, 3, 4:

```
for i in range(5):  
    print(i)
```

for – «для»

in range – «в диапазоне» в переводе с английского

# Цикл с переменной

---

Увеличение переменной на 1:

```
for <переменная> in <конечное значение> :  
    {тело цикла}
```

Чему равно значение переменной  $a$ ?

```
a = 1  
for i in range(3): a = a + 1
```

$a = 4$

Можно указать начальное и конечное значение диапазона:

```
for i in range(3, 20):  
    print(i)
```

При этом переменная *i* будет принимать значения **от 3 до 19**

Можно указать шаг, с которым будет меняться значение переменной `i` :

```
for i in range(3, 20, 4):  
    print(i)
```



Начальное  
значение

Конечное  
значение

Шаг

Здесь значение переменной `i` будет меняться в диапазоне **от 3 до 19**, увеличиваясь каждый раз **на 4**

Вывод:

3

7

11

15

19

# Цикл `while` (“пока”)

- позволяет выполнить одну и ту же последовательность действий, пока проверяемое условие истинно. Условие записывается до тела цикла и проверяется до выполнения тела цикла.
- как правило, цикл `while` используется, когда невозможно определить точное значение количества проходов исполнения цикла.

# При выполнении цикла `while`

- Сначала проверяется условие, если оно ложно, то выполнение цикла прекращается и управление передается на следующую инструкцию после тела цикла `while`. Если условие истинно, то выполняется инструкция, после чего условие проверяется снова и снова выполняется инструкция. Так продолжается до тех пор, пока условие будет истинно. Как только условие станет ложно, работа цикла завершится и управление передастся следующей инструкции после цикла.



Например, следующий фрагмент программы напечатает на экран квадраты всех целых чисел от 1 до 10.

- `i = 1`
- `while i <= 10:`
- `print(i ** 2)`
- `i += 1`