

Циклы

Повторение

Цикл со счетчиком

Для элемента из последовательности

```
for x in [0, 1, 2, 3, 4]:  
    .... print('x =', x)
```

тело цикла

Генератор последовательности чисел range

Функция `range()` генерирует последовательность чисел в рамках заданного диапазона

Способы вызова:

- `range(стоп)` #сколько чисел начиная с нуля
`range(5)` -> [0, 1, 2, 3, 4]
`range(10)` -> [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
- `range(старт, стоп)` #[старт, стоп)
`range(0, 5)` -> [0, 1, 2, 3, 4]
`range(1, 6)` -> [1, 2, 3, 4, 5]
`range(3, 10)` -> [3, 4, 5, 6, 7, 8, 9]
- `range(старт, стоп, шаг)` #следующий = текущий + шаг
`range(0, 5, 1)` -> [0, 1, 2, 3, 4]
`range(1, 15, 3)` -> [1, 4, 7, 10, 13]
`range(10, 0, -1)` -> [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

Задачи для цикла со СЧЕТЧИКОМ

Назовём число **красивым**, если **сумма его цифр** делится на **количество цифр** в нем.

A: Ввести число x , вычислить и вывести сумму и количество его цифр.

Введите число: 213

Сумма цифр: 6

Количество цифр: 3

Введите число: 5128

Сумма цифр: 16

Количество цифр: 4

Введите число: 37285

Сумма цифр: 25

Количество цифр: 5

B: Ввести число x , вывести («да» или «нет») является ли это число красивым числом.

Введите число: 5128

Да

Введите число: 734

Нет

C: Ввести два числа a и b , вывести все красивые числа от a до b .

Введите два числа: 4950 5020

4952 4956 4961 4965 4969 4970 4974 4978 4983 4987 4992 4996 5003 5007 5012 5016

D: Ввести число m , вывести m -ое в порядке возрастания красивое число.

Введите число: 15

20

Введите число: 100

237

Цикл с условием while

Пока условие повторения

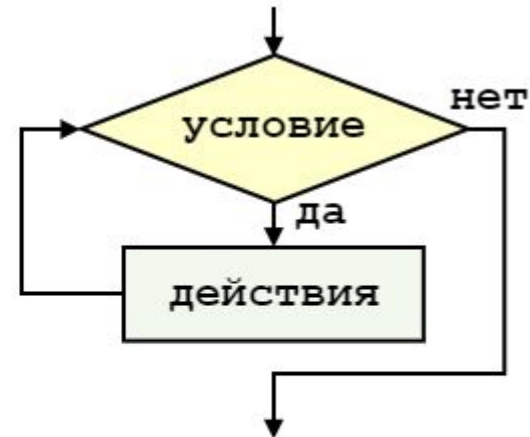
```
while условие:
```

```
..... действия
```

```
..... действия
```

Тело цикла – это команды, которые повторяются несколько раз.

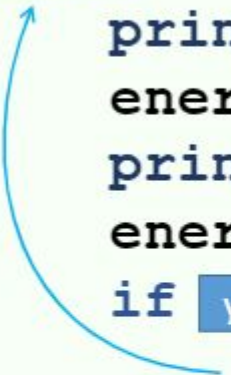
Не забывайте отступы!



Цикл while выполняется до тех пор, пока некоторое условие не станет ложным. Количество шагов (повторений) определяется исходными данными.

Управление циклом continue


```
energy = 23
print('Иду на стадион')
while energy > 10:
    print('Бегу полкруга')
    energy -= 2
    print('Приседаю')
    energy -= 1
    if учитель физ-ры отвернулся :
        continue
    print('Бегу полкруга')
    energy -= 2
print('Иду домой')
```



Оператор `continue` начинает следующий проход цикла, минуя оставшееся тело цикла.

Управление циклом break

```
energy = 23
print('Иду на стадион')
while energy > 10:
    print('Бегу полкруга')
    energy -= 2
    print('Приседаю')
    energy -= 1
    if звенит звонок с урока :
        break
    print('Бегу полкруга')
    energy -= 2
print('Иду домой')
```



Оператор `break` досрочно прерывает цикл.

Цикл с условием + действия на выходе из цикла.

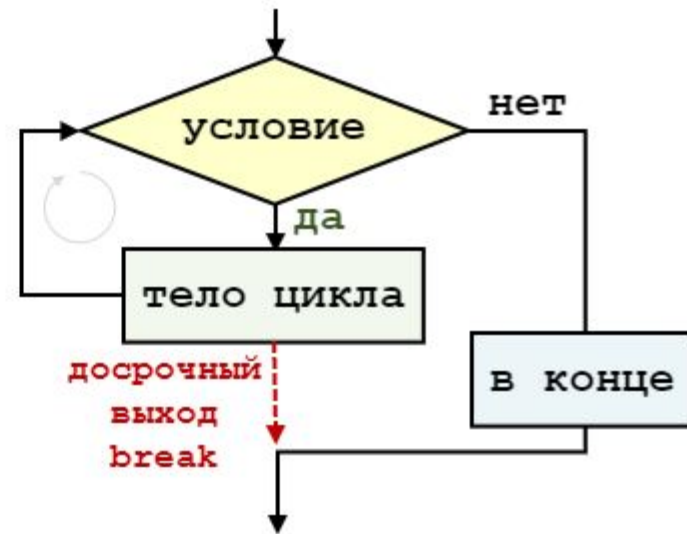
while **условие:**

.... **тело цикла**

else:

.... **действия**

Выполняются только в том случае, если не было досрочного выхода из цикла



Определите результат
выполнения цикла:

```
x = 1
while x <= 10:
    print(x, end=' ')
    x += 1
```

Задача 1

Используя цикл `while` выведите последовательность чисел:

1) 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

2) 10 11 12 13 14 15 16 17 18 19 20

3) 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29

4) 0 5 10 15 20 25 30 35 40 45 50 55 65 70 75 80 85 90 95 100

5) 20 18 16 14 12 10 8 6 4 2

6) 1 2 4 8 16 32 64 128 256 512 1024

7) 1 5 17 53 161 485 1457 4373

Задача «Утренняя пробежка». В первый день спортсмен пробежал x километров, а затем он каждый день увеличивал дистанцию пробег на 10% от предыдущего значения. По данному числу y определите номер дня, на который пробег спортсмена составит не менее y километров.

```
x = float(input('В первый день: '))
y = float(input('Нужно: '))
n = 1
while x < y:
    x = x + 0.1*x
    n += 1
print(n)
```

```
В первый день: 2
Нужно: 3.5
7
```

1-й день: x

2-й день: $x = x + 0.1 \cdot x$

3-й день: $x = x + 0.1 \cdot x$

4-й день: $x = x + 0.1 \cdot x$

...

Пока $x < y$

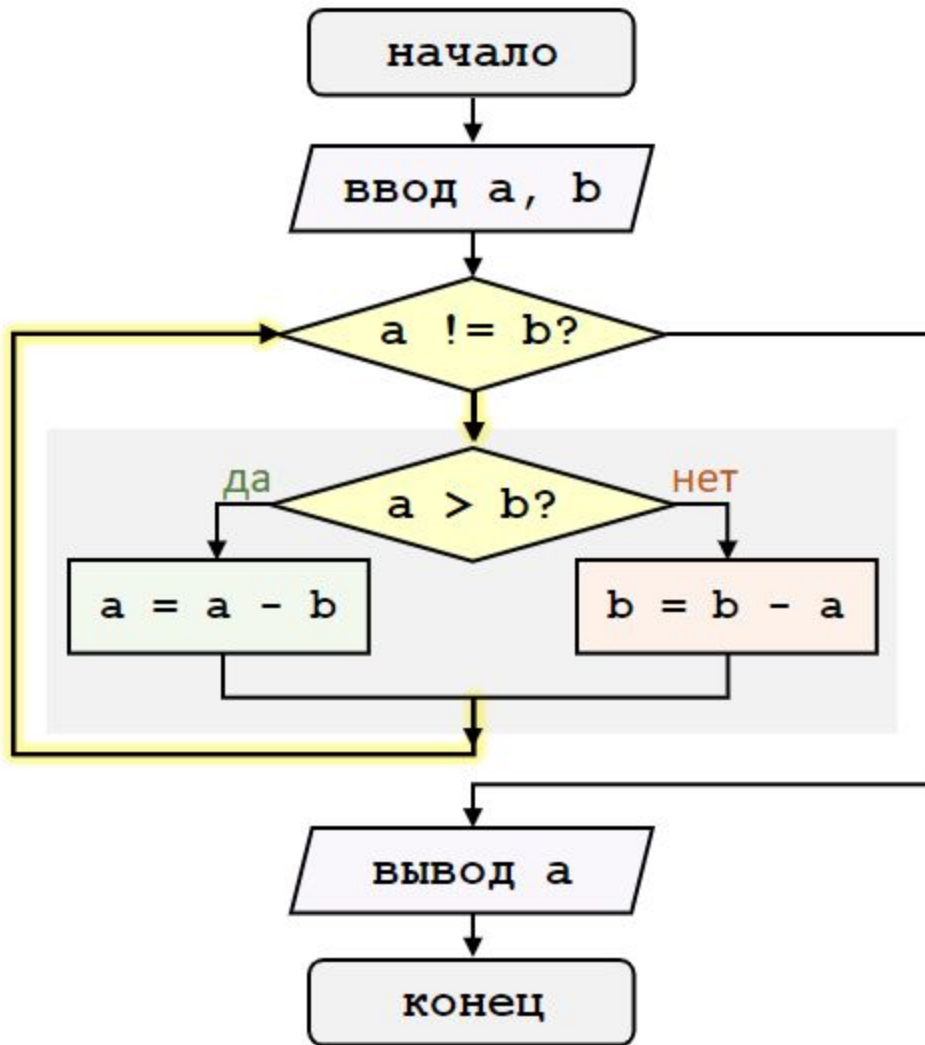
Определить результат выполнения циклов

```
s = 0
n = 1
while s < 51:
    s = s + 11
    n = n * 2
print(n)
```

```
s = 260
n = 0
while s > 0:
    s = s - 15
    n = n + 2
print(n)
```

```
s = 0
n = 75
while s + n < 150:
    s = s + 15
    n = n - 5
print(n)
```

Алгоритм Евклида



Алгоритм Евклида (нахождение НОД)

До тех пор, пока числа не станут равны, заменяем большее из двух чисел разностью большего и меньшего. Это и есть НОД.



Евклид
(365-300 до. н. э.)

НОД (147 и 84) = ?

a	b
147	84
63	84
63	21
42	21
21	21

НОД (147 и 84) = 21

* НОД - наибольший общий делитель

Цикл while – stepik 7.4_1

- На вход программе подается последовательность слов, каждое слово на отдельной строке. Концом последовательности является слово «**КОНЕЦ**» (без кавычек).
Напишите программу, которая выводит члены данной последовательности.
- **Формат входных данных**
На вход программе подается последовательность слов, каждое слово на отдельной строке.
- **Формат выходных данных**
Программа должна вывести члены данной последовательности.

Цикл while – stepik 7.4_3

- На вход программе подается последовательность слов, каждое слово на отдельной строке. Концом последовательности является слово «**КОНЕЦ**» (без кавычек). Напишите программу, которая выводит члены данной последовательности.
- **Формат входных данных**
На вход программе подается последовательность слов, каждое слово на отдельной строке.
- **Формат выходных данных**
Программа должна вывести члены данной последовательности.

Задача о цифрах числа:

Вывод цифр числа (в обратном порядке):

```
a = int(input('Введите число: '))  
while a > 0:  
    d = a % 10  
    print(d)  
    a = a // 10
```

1234
4
3
2
1

Вычислить сумму цифр произвольного целого числа:

```
a = int(input('Введите число: '))  
s = 0  
while a > 0:  
    d = a % 10  
    s = s + d  
    a = a // 10  
print('Сумма цифр =', s)
```


Задача о цифрах числа № 2

Получив на вход число x , этот алгоритм печатает два числа: a и b . Укажите наименьшее из таких чисел x , при вводе которых алгоритм печатает сначала 13, а потом 5.

```
x = int(input())
a = 0
b = 10
while x > 0:
    d = x % 10
    a = a + d
    if d < b:
        b = d
    x = x // 10
print(a, b)
```