

Программирование на языке Python

Циклические алгоритмы

Что такое цикл?

Цикл – это многократное выполнение одинаковых действий.

Два вида циклов:

- цикл с **известным** числом шагов (сделать 10 раз)
- цикл с **неизвестным** числом шагов (делать, пока не надоест)

Задача. Вывести на экран 10 раз слово «Привет».



Можно ли решить известными методами?

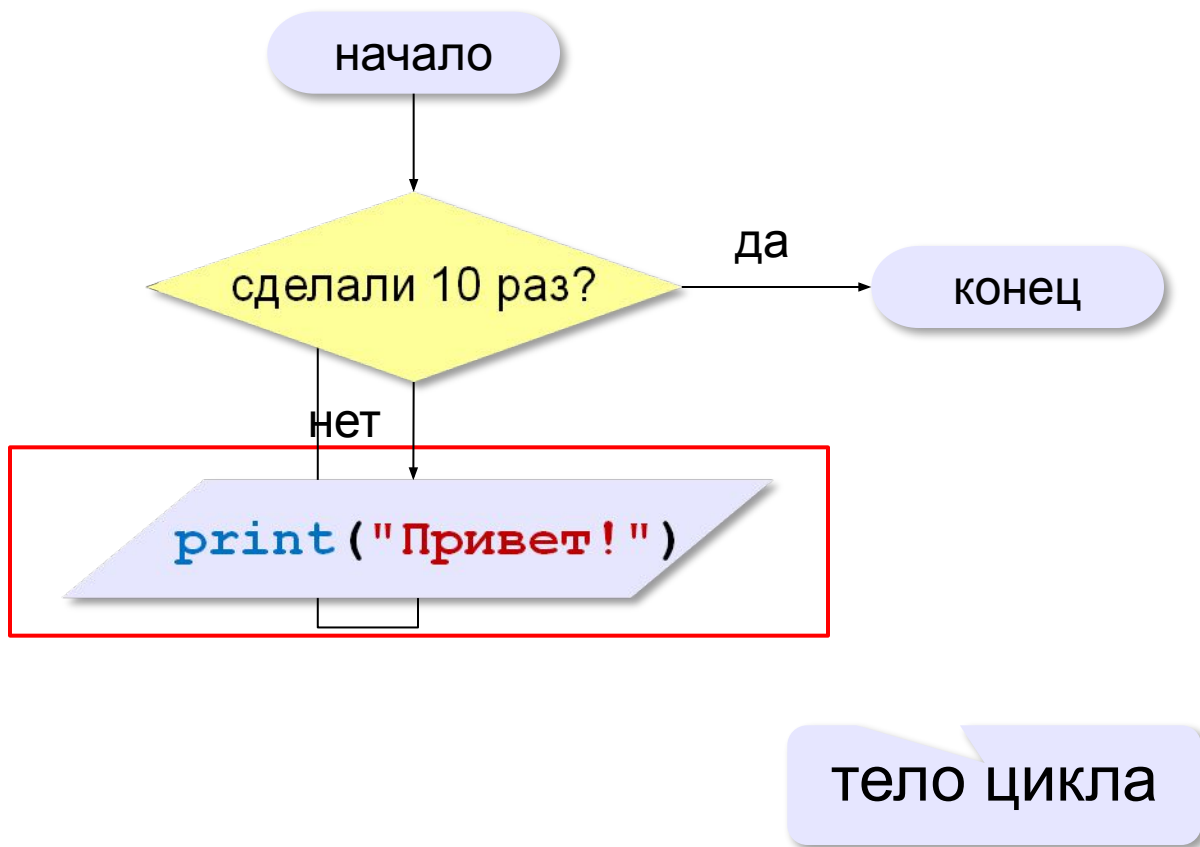
Повторения в программе

```
print ("Привет")  
print ("Привет")  
...  
print ("Привет")
```



Что плохо?

Блок-схема цикла



Как организовать цикл?

```
счётчик = 0
пока счётчик < 10:
    print("Привет")
    увеличить счётчик на 1
```

```
k = 0
while k < 10:
    print("Привет")
    k += 1
```



Как по-другому?

```
счётчик = 10
пока счётчик > 0:
    print("Привет")
    уменьшить счётчик на 1
```

```
k = 10
while k > 0:
    print("Привет")
    k -= 1
```

Сколько раз выполняется цикл?

```
a = 4; b = 6  
while a < b: a += 1
```

2 раза

a = 6

```
a = 4; b = 6  
while a < b: a += b
```

1 раз

a = 10

```
a = 4; b = 6  
while a > b: a += 1
```

0 раз

a = 4

```
a = 4; b = 6  
while a < b: b = a - b
```

1 раз

b = -2

```
a = 4; b = 6  
while a < b: a -= 1
```

защелкивание

Цикл с условием

Задача. Определить **количество цифр** в десятичной записи целого положительного числа, записанного в переменную n .

```
счётчик = 0
пока n > 0:
    отсечь последнюю цифру n
    увеличить счётчик на 1
```

n	счётчик
1234	0

? Как отсечь последнюю цифру?

```
n = n // 10
```

? Как увеличить счётчик на 1?

```
счётчик = счётчик + 1
```

```
счётчик += 1
```

Цикл с условием

начальное значение
счётчика

условие
продолжения

заголовок
цикла

```
count = 0
while n > 0 :
    n = n // 10
    count += 1
```

тело цикла



Цикл с предусловием – проверка на входе в цикл!

Задачи

«3»: Ввести с клавиатуры количество повторений и вывести столько же раз какое-нибудь сообщение.

Пример:

Сколько раз :

5

Привет!

Привет!

Привет!

Привет!

Привет!

Задачи

«4»: Ввести с клавиатуры натуральное число и определить, сколько раз в его записи встречается цифра 1.

Пример:

Введите число:

51211

3

«5»: Ввести с клавиатуры натуральное число и найти сумму значений его цифр.

Пример:

Введите число:

1234

Сумма цифр 10

Задачи

«6»: Ввести натуральное число и определить, верно ли, что в его записи есть две одинаковые цифры, стоящие рядом.

Пример:

Введите натуральное число:

12342

Нет.

Пример:

Введите натуральное число:

12245

Да.

Алгоритм Евклида

Алгоритм Евклида. Чтобы найти НОД двух натуральных чисел, нужно вычитать из большего числа меньшее до тех пор, пока они не станут равны. Это число и есть НОД исходных чисел.

$$\text{НОД}(14,21) = \text{НОД}(14,7) = \text{НОД}(7, 7) = 7$$

```
пока a != b:  
    если a > b:  
        a -= b # a = a - b  
    иначе:  
        b -= a # b = b - a
```

```
while a != b:  
    if a > b:  
        a -= b  
    else:  
        b -= a
```

$$\text{НОД}(1998,2) = \text{НОД}(1996,2) = \dots = \text{НОД}(2, 2) = 2$$

Алгоритм Евклида

Модифицированный алгоритм Евклида. Заменять большее число на остаток от деления большего на меньшее до тех пор, пока меньшее не станет равно нулю. Другое (ненулевое) число и есть НОД чисел.

$$\text{НОД}(1998, 2) = \text{НОД}(0, 2) = 2$$

```
пока a != 0 and b != 0 :
```

```
    если a > b :
```

```
        a = a % b
```

```
    иначе :
```

```
        b = b % a
```

```
если a != 0 :
```

```
    вывести a
```

```
иначе :
```

```
    вывести b
```



Какое условие?



Как вывести результат?

Задачи

«3»: Ввести с клавиатуры два натуральных числа и найти их НОД с помощью алгоритма Евклида.

Пример:

Введите два числа:

21 14

НОД (21 , 14) =7

«4»: Ввести с клавиатуры два натуральных числа и найти их НОД с помощью **модифицированного** алгоритма Евклида. Заполните таблицу:

a	64168	358853	6365133	17905514	549868978
b	82678	691042	11494962	23108855	298294835
НОД (a , b)					

Задачи

«5»: Ввести с клавиатуры два натуральных числа и сравнить количество шагов цикла для вычисления их НОД с помощью обычного и модифицированного алгоритмов Евклида.

Пример:

Введите два числа:

1998 2


НОД(1998, 2) = 2

Обычный алгоритм: 998

Модифицированный: 1

Цикл с переменной

Задача. Вывести 10 раз слово «Привет!».

 Можно ли сделать с циклом «пока»?

```
i = 0
while i < 10 :
    print ("Привет!")
    i += 1
```

Цикл с переменной:

```
for i in range(10) :
    print ("Привет!")
```

в диапазоне
[0, 10)

 Не включая 10!

`range(10)` → 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Цикл с переменной

Задача. Вывести все степени двойки от 2^1 до 2^{10} .

 Как сделать с циклом «пока»?

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

возведение
в степень

Цикл с переменной:

```
for k in range (1, 11) :
    print ( 2**k )
```

в диапазоне
[1, 11)

 Не включая 11!

`range (1, 11)` → 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Цикл с переменной: другой шаг

10, 9, 8, 7, 6, 5, 4, 3, 2, 1

шаг

```
for k in range(10, 0, -1) :  
    print ( k**2 )
```



Что получится?

1, 3, 5, 7, 9

```
for k in range(1, 11, 2) :  
    print ( k**2 )
```

100

81

64

49

36

25

16

9

4

1

1

9

25

49

81

Сколько раз выполняется цикл?

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

a = 4

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

a = 1

```
a = 1  
for k in range(1, 3, -1): a += 1
```

a = 1

```
a = 1  
for k in range(3, 0, -1): a += 1
```

a = 4

Задачи

- «3»: Ипполит задумал трёхзначное число, которое при делении на 15 даёт в остатке 11, а при делении на 11 даёт в остатке 9. Найдите все такие числа.
- «4»: Вводится натуральное число N . Программа должна найти **факториал** (обозначается как $N!$) – произведение всех натуральных чисел от 1 до N . Например, $5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$.

Пример:

Введите число :

5

$5! = 120$.