

ОПЕРАТОРЫ ЦИКЛА

С помощью циклов в программе можно выполнять повторяющиеся действия.

Различают два вида циклов:

- 1.циклы с условием
- 2.циклы по переменной

Цикл с условием выполняется до тех пор, пока некоторое условие (условие продолжения работы цикла) не станет ложным. Если это условие никогда не станет ложным, программа заиклиивается.

Пример оператора цикла с условием:

Напечатать все значения переменной X, меньшие значения Y

повторять, пока условие
начало цикла
последовательность инструкций
конец цикла

```
x = 20
y = 30
while x < y:
    print(x, end=' ')
    x = x + 3
```

20 23 26 29

Цикл по переменной применяют тогда, когда количество шагов цикла заранее известно или может быть вычислено до начала цикла. В заголовке цикла по переменной указывают начальное значение, конечное значение и шаг изменения переменной цикла.

ЦИКЛЫ ПО ПЕРЕМЕННОЙ

Необходимо организовать цикл, в котором блок операторов выполнится заданное число. Для этого можно применить вид цикла – **цикл по переменной** (или цикл с параметром). На языке Python записывается :

```
for i in range(10):  
print( "привет" )
```

for означает «для», переменная *i* - *переменная цикла*, изменяется в диапазоне (**in range**) от 0 до 10, не включая 10 (то есть от 0 до 9 включительно).

Таким образом, цикл выполняется для $i = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9$ – ровно 10 раз.

Переменная *i* – это счётчик выполненных итераций цикла. Можно было записать этот цикл иначе:

```
for i in [0,1,2,3,4,5,6,7,8,9]:  
print( "привет" )
```

В квадратных скобках через запятую перечислены все значения переменной, при которых выполняется цикл. Если их много, такой способ неудобен, лучше использовать встроенную функцию **range**. цикл с переменной всегда заканчивается, программа не может зациклиться.

Шаг изменения переменной цикла

По умолчанию функция **range** строит последовательность, в которой каждое следующее число на 1 больше предыдущего. Но это правило можно изменить, если при вызове функции **range** указать третий аргумент – шаг изменения переменной цикла.

ПРИМЕР

Программа печатает квадраты натуральных чисел от 10 до 1 в порядке убывания:

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

В этом примере шаг равен -1 , то есть каждое следующее число на 1 меньше предыдущего. Конечное значение 0 не входит в последовательность.

Пример,

Перебрать в цикле все значения переменной i от 0 до 100, кратные пяти: 0, 5, 10, ..., 100. Для этого нужно взять шаг изменения переменной 5:

```
for i in range(0, 101, 5):  
... # что-то делать с i
```

Второй аргумент функции range равен 101 для того, чтобы последнее значение переменной i было равно 100. Значение-ограничитель должно быть больше, чем 100 (чтобы число 100 появилось в последовательности), но меньше, чем 106 (чтобы следующее число, 105, не появилось).

ДОМАШНЕЕ ЗАДАНИЕ!!!!

Задача. Определить сумму положительных и отрицательных значений функции $Z=a+b^x$, где X -это параметр цикла, который принимает значения на числовом интервале от x_n до x_k с шагом h

Ниже представлена программа вычисления на языке Питон (два варианта), записать в тетрадь и составить блок-схему алгоритма вычисления

ЦИКЛЫ:

<pre># все параметры цикла вводит пользователь from math import* a = int(input()) b = int(input()) xn = int(input()) xk = int(input()) h = int(input()) s1 = 0 s2 = 0 for x in range(xn, xk, h): z = a+b**x print (z) if z>=0: s1 = s1+z else: s2 = s2+z print ("s1=",s1,"s2=",s2)</pre>	<pre>2 3 4 10 1 19685 s1= 19685 s2= 0</pre>
<pre>a = int(input()) b = int(input()) s1 = 0 s2 = 0 for x in range(-4, 11, 1): z = a+b**x print ("z=",z,"x=",x) if z>=0: s1 = s1+z else: s2 = s2+z print ("s1=",s1,"s2=",s2) к-во результатов=10-(-4)/1+1=15</pre>	<pre>1 3 z= 1.0123456790123457 x= -4 z= 1.037037037037037 x= -3 z= 1.1111111111111112 x= -2 z= 1.3333333333333333 x= -1 z= 2 x= 0 z= 4 x= 1 z= 10 x= 2 z= 28 x= 3 z= 82 x= 4 z= 244 x= 5 z= 730 x= 6 z= 2188 x= 7 z= 6562 x= 8 z= 19684 x= 9 z= 59050 x= 10 s1= 88588.49382716049 s2= 0</pre>