

ПЕРЕМЕННЫЕ И ПРОСТЕЙШИЕ ОПЕРАЦИИ С НИМИ

Переменная – это место в оперативной памяти для временного хранения данных. Переменная имеет имя, тип и значение.

Основные типы переменных

str	строка символов
int	целое число
float	вещественное число
bool	логическая переменная

Арифметические действия

a ** b	возведение в степень
a * b	произведение
a / b	частное от деления
a + b	сумма
a - b	разность

Не запуская код, ответьте на вопрос: что выведет на экран данная программа?

```
1 # Пример № 1
2 a=5
3 b=a*10
4 a=a+b+5
5 print(a+b)
```

```
1 # Пример № 2
2 a=3
3 a=a*5+a**2
4 print(a)
```

Ввод значений переменных

a = int (input('a= '))	ввод <u>целого</u> числа a с пояснениями
z=float (input('z= '))	ввод вещественной переменной z с пояснениями
x=15	прямое присвоение

Вывод значений переменных

print(y) print((a+b)*c)	Можно выводить значение переменной или значение выражения.
print('Привет!')	Текст для вывода может записываться и в двойных, и в одинарных кавычках.
print('Сумма ', s)	При выводе значение переменной можно дополнять пояснениями.

3. Напишите программу, которая выводит на экран значение выражения 2^{1234} .
4. Напишите программу, которая будет считывать два целых числа (каждое в отдельной строке) и выводить их сумму, разность, произведение и частное.
5. Заданы целые числа **a** и **b**. Создайте проект, позволяющий вычислять число **c** по формуле:
$$c = \frac{a^2 + b}{b^2 - a}$$
6. Два велосипедиста едут на встречу друг другу со скоростью V_1 и V_2 . Расстояние между ними S , через сколько часов они встретятся.

ОПЕРАЦИИ ЦЕЛОЧИСЛЕННОГО ДЕЛЕНИЯ И ВЗЯТИЯ ОСТАТКА ОТ ДЕЛЕНИЯ

Операция целочисленного деления (`//`) делит ту часть числа, которая делится нацело, остальное просто отбрасывает.

Операция взятия остатка от деления (`%`).

```
code.py
1 print(19/3)
2 print(19//3)
3 print(19%3)
4 print((-19)//3)
5 print((-19)%3)
```

```
Оболочка
>>> %Run code.py
6.333333333333333
6
1
-7
2
Обратите
внимание на
отрицательные
числа.
>>>
```

$19 // 3 = 6$

$18 + 1$

$18 : 3 = 6$

$19 \% 3 = 1$

$18 + 1$

$-19 // 3 = -7$

-21 (округление «вниз»!)

$-21 : 3 = -7$

$-19 \% 3 = 2$

$-21 + 2$

остаток ≥ 0

Рассмотрим пример. Дано трёхзначное число. Найдите сумму его цифр.

```
1 n = int(input('n='))
2 d3 = n % 10
3 d2 = n // 10 % 10
4 d1 = n // 100
5 print(d1 + d2 + d3)
```

```
Оболочка
n=292
13
```

- просит пользователя ввести число
- последняя цифра числа
- вторая цифра трёхзначного числа
- первая цифра трёхзначного числа
- вывод результата

ОПЕРАЦИИ С ВЕЩЕСТВЕННЫМИ ЧИСЛАМИ

Число, в котором выделяются целая и дробная части, в информатике называется вещественным.

Дробная часть от целой отделяются точкой (".").

Модуль **math** – предоставляет обширный функционал для работы с числами. Для его подключения в начало программы пишут строку **from math import***. И тогда можно использовать следующие функции:

fabs(X) - модуль X.

sqrt(X) - квадратный корень из X.

cos(X) - косинус X (X указывается в радианах).

sin(X) - синус X (X указывается в радианах).

tan(X) - тангенс X (X указывается в радианах).

degrees(X) - конвертирует радианы в градусы.

radians(X) - конвертирует градусы в радианы.

Pi (pi = 3,1415926...)

e (e = 2,718281...)

Язык программирования Python

7. Дано двузначное число. Найти число десятков и число единиц в нем, сумму и произведение его цифр.
8. Дано двузначное число. Получить число, образованное при перестановке цифр заданного числа.
9. Напишите программу, которая моделирует работу следующего автомата. Автомат получает на вход четырёхзначное число. Затем вычисляются три суммы: сумма первых двух чисел, сумма средних цифр и сумма последних двух цифр. Результат работы автомата – произведение этих сумм.
10. n школьников делят между собой k яблок поровну, не делящийся остаток остаётся в корзинке. Сколько яблок достанется каждому школьнику? Сколько яблок останется в корзинке?
11. Напишите программу, которая вычисляет квадратный корень введённого числа.
12. Даны целочисленные ненулевые координаты точки T на плоскости. Из начала отсчёта в точку T провели отрезок, который является диагональю прямоугольника. Вычислить площадь и периметр этого прямоугольника.
13. Периметр прямоугольника P , одна из его сторон a . Какова площадь данного прямоугольника?
14. Площадь прямоугольника S , a одна из его сторон a . Чему равен периметр прямоугольника?

СЛУЧАЙНЫЕ ЧИСЛА

Случайные числа – это такая последовательность чисел, в которой невозможно назвать следующее число, зная сколько угодно предыдущих.

from random import * x1 = randint(a,b)	<i>получение <u>целого</u> числа x1 в интервале [a, b] с помощью генератора случайных чисел</i>	<i>a и b в интервал входят, но их нужно предварительно ввести одним из вышеперечисленных способов или просто написать в скобках числа</i>
from random import * x2 = uniform(a, b)	<i>получение <u>вещественного</u> числа x2 в интервале [a, b] с помощью генератора случайных чисел</i>	

15. Напишите программу, которая вводит с клавиатуры два целых числа, a и b, и выводит на экран 2 случайных целых чисел на отрезке [a; b].
16. Игральный кубик бросается четыре раза (выпадает четыре случайных значения). Сколько очков в среднем выпало?
17. Игральный кубик бросается три раза (выпадает три случайных значения). Из этих чисел составляется целое число, программа должна найти его квадрат.
18. Получите случайное трёхзначное число и выведите через запятую его отдельные цифры.

ВЕТВЛЕНИЕ

Разветвляющийся алгоритм – это алгоритм, содержащий хотя бы одно условие, в результате которого обеспечивается переход на один из двух возможных шагов.

При записи условий используются следующие операторы сравнения:

Обозначение	Операция	Запись на Python
==	равно	x==1
!=	не равно	x!=y
>	больше	x+a> y
<	меньше	x <y-b

Обозначение	Операция	Запись на Python
>=	больше или равно	x>=0
<=	меньше или равно	x <=1

Сложные условия

x > 0 and y < 0	x > 0 or y < 0
все условия должны выполняться одновременно	должно выполняться хотя бы одно из условий

Кроме того, операции сравнения в Python можно объединять в цепочки (в отличие от большинства других языков программирования, где для этого нужно использовать логические связки), например, **x == 0 == y, 10 <= n <= 100**.

Язык программирования Python

Условная инструкция в Python имеет следующий синтаксис:

if условие:

 блок инструкций 1

else:

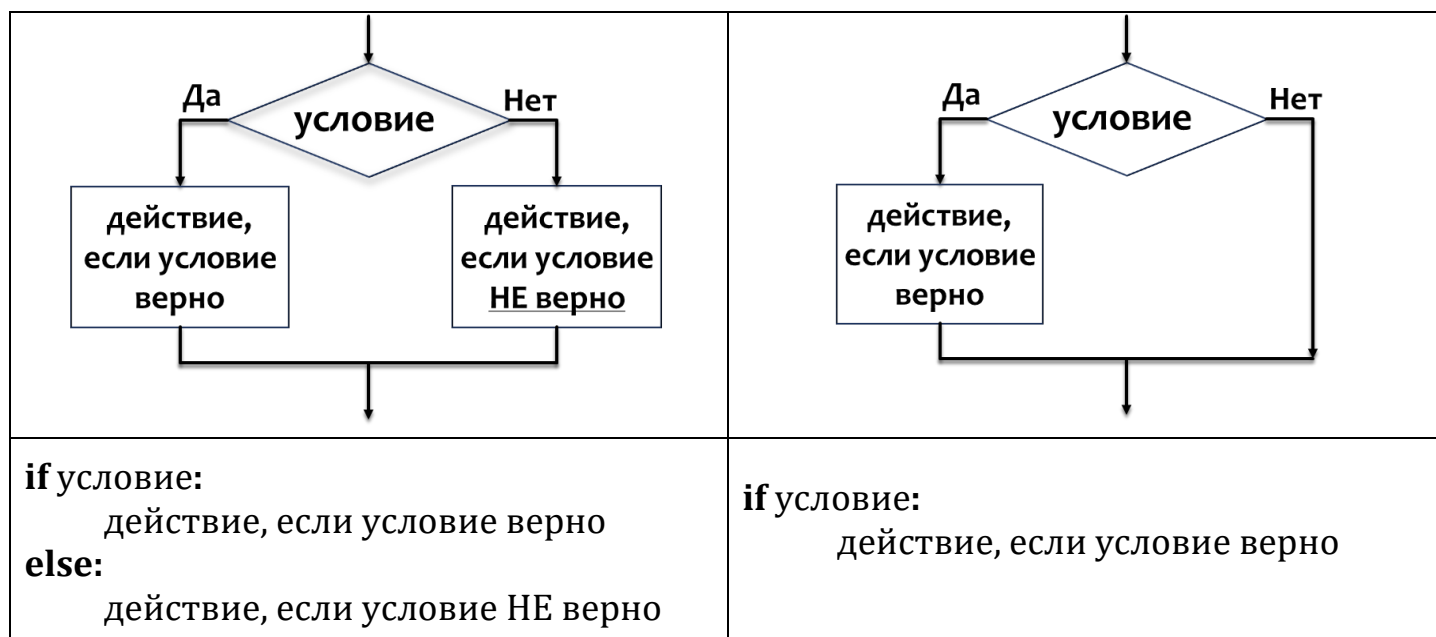
 блок инструкций 2

Для выделения блока инструкций, относящихся к инструкции **if** или **else**, в языке Python используются отступы.

Все инструкции, которые относятся к одному блоку, должны иметь равную величину отступа, то есть одинаковое число пробелов в начале строки.

Можно использовать в качестве отступа символ табуляции.

ПОЛНЫЙ И НЕПОЛНЫЙ УСЛОВНЫЙ ПЕРЕХОД



19. Напишите программу, которая вводит целое и выводит ответ «да», если оно чётное и «нет», если оно нечётное.
20. Напишите программу, которая вводит трёхзначное число и выводит ответ «да», если в его записи есть цифра 0, и «нет», если ноль отсутствует. *Попробуйте обойтись одним условным оператором.*
21. Занятия в начальных классах отменяются в тех случаях, когда температура воздуха ниже -25 градусов, а также при ветре не менее 7 м/с и температуре ниже -20 градусов. Составьте программу, которая по утренней сводке погоды определяет, пойдут ли дети в школу.
22. Паша очень любит кататься на общественном транспорте, а получая билет, сразу проверяет, счастливый ли ему попался. Билет считается счастливым, если сумма первых трех цифр совпадает с суммой последних трех цифр номера билета. Однако Паша очень плохо считает в уме, поэтому попросил вас написать программу, которая проверит равенство сумм и выведет "Счастливый", если суммы совпадают, и "Обычный", если суммы различны. На вход программе подаётся строка из шести цифр.

КАСКАДНОЕ ВЕТВЛЕНИЕ И ВЛОЖЕННЫЕ УСЛОВНЫЕ ОПЕРАТОРЫ

Рассмотрим на примере: составить программу, которая по знаку арифметической операции выведет на экран вычисления в виде равенства.

elif = else if

23. Напишите программу, которая получает на вход три целых числа, по одному числу в строке, и выводит на консоль в три строки сначала максимальное, потом минимальное, после чего оставшееся число.

24. Найти значение функции у заданной формулой:

25. Напишите программу, которая получает номер месяца и выводит соответствующее ему время года или сообщение об ошибке.

26. Напишите программу, которая вводит с клавиатуры номер месяца и день и определяет, сколько дней осталось до Нового года. При вводе неверных данных должно быть выведено сообщение об ошибке.

27. Автомат получает на вход **четырёхзначное** десятичное число. Новое десятичное число строится по следующим правилам.

1) Вычисляются два числа — сумма «крайних» цифр четырёхзначного числа и сумма «средних» цифр четырёхзначного числа.

2) Полученные два числа записываются друг за другом в порядке **невозрастания** (без разделителей).

Пример. Исходное число: 7345. Сумма «крайних» цифр: 12, сумма «средних» цифр числа: 7. Результат: 127.

Напишите программу, которая принимает введенное с клавиатуры число и определяет число, которое будет получено в результате работы автомата.

```
1 x = int (input('x= '))
2 y = int (input('y= '))
3 z = str (input('znak '))
4 if z=='+':
5     print(x+y)
6 elif z=='-':
7     print(x-y)
8 elif z=='*':
9     print(x*y)
10 elif z=='/':
11     if x==0:
12         print('Делить на ноль нельзя!!!')
13     else:
14         print(x/y)
15 else:
16     print('только +, -, * или /')
```

Оболочка <

>>> %Run 'каскадное ветвление.py'

```
x= 5
x= 6
znak *
30
```

$$y = \begin{cases} 30 - \frac{x}{x-5}, & \text{при } x = 100; \\ 13x^2 + x, & \text{при } x < 100 \\ x^3 - 100, & \text{при } x > 100 \end{cases}$$

Цикл с условием (итерационный)

Цикл – это повторяющаяся последовательность действий.

while условие:

повторяющиеся действия

позволяет выполнять одну и ту же последовательность действий, пока проверяемое условие истинно (число повторений заранее не известно)

break – оператор досрочного завершения цикла (используется редко).

1. Какое будет значение *i* после выполнения инструкций?

```
i, n = 0, 625
while n > 0:
    n = n // 5
    i = i + 1
print(i)
```

2. Найдите и исправьте ошибку в программе:

```
k = 0
while k < 10:
    print('Привет!')
```

3. Какое значение будет у переменной *i* после выполнения фрагмента программы? Сколько итераций цикла будет выполнено в этом фрагменте программы?

```
i = 0
while i <= 10:
    i = i + 1
    if i > 7:
        i = i + 2
print(i)
```

4. Сколько всего знаков * будет выведено после выполнения фрагмента программы:

```
i = 0
while i < 5:
    print('*')
    if i % 2 == 0:
        print('**')
    if i > 2:
        print('***')
    i = i + 1
```

5. Какое значение будет выведено после выполнения фрагмента программы?

```
i=0
while i<5:
    a=int(input())
    b=int(input())
    if (a==0) and (b==0):
        break #досрочно завершаем цикл
    print (a*b)
    i+=1
```

6. Рассмотрим программу, которая определяет сумму последовательности целых чисел. Программа получает на вход целые числа, количество введенных чисел не известно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность). Количество чисел не превышает 1000. Введенные числа по модулю не превышают 30 000.

```
s=0
x=int(input('x= '))
while x!=0:
    s=s+x
    x=int(input('x= '))
print('s= ', s)
```

7. Дополните программу № 6:

- а) выводится не только сумма, но и все введенные числа;
- б) среднее арифметическое всех чисел;
- в) выводится сумма всех четных чисел;
- г) выводятся нечетные числа;
- д) выводится количество четных и нечетных чисел.

ЦИКЛ ПО ПЕРЕМЕННОЙ

Цикл по переменной используют, когда количество повторений цикла известно или может быть вычислено заранее

for i in range(n): <i>повторяющиеся действия</i>	i – переменная цикла (имя может быть любое); n – верхняя граница диапазона; i меняется от 0 до n-1 с шагом +1 (т.е. на первом круге i будет равна 0, на втором i=1, на третьем i=2 и т.д.)
for i in range(a, b): <i>повторяющиеся действия</i>	a – начальное значение переменной i; b – верхняя граница диапазона; a ≤ b!!! шаг изменения переменной i равен +1 .
for i in range(a, b, t): <i>повторяющиеся действия</i>	a – начальное значение переменной i; b – верхняя граница диапазона; t – шаг (на сколько изменяется переменная i на каждом круге цикла) если a ≤ b , то t > 0 ; если a > b , то t < 0 .

Вывод на экран чисел
от 0 до 5.

```
1 n=6
2 for i in range(n):
3     print(i)
```

Оболочка ×

```
0
1
2
3
4
5
```

Вывод на экран степеней
двойки в интервале
от a до (b-1).

```
1 a=5
2 b=10
3 for i in range(a,b):
4     print(2**i)
```

Оболочка ×

```
32
64
128
256
512
```

Вывод на экран
обратного отсчёта.

```
1 a=5
2 b=0
3 d=-1
4 for i in range(a, b, d):
5     print(i)
6 print('Поехали!')
```

Оболочка ×

```
5
4
3
2
1
Поехали!
```

8. Что будет выведено на экран в результате работы следующих циклов?

```
k=1
for i in range(5):
    print(i, end="")
```

```
k=1
for i in range(5):
    print(k*k, end="")
    k+=2
```

```
k=8
for i in range(5,0,-1):
    print(2*i-k, end="")
    k-=2
```

9. Даны два целых числа A и B. Выведите все числа от A до B включительно в порядке возрастания, если A < B, или в порядке убывания в противном случае.

10. На вход программы подаётся натуральное 10-значное число. Выведите все цифры этого числа в обратном порядке по одной.

Язык программирования Python

11. Дано натуральное число **n**. Выведите слово **YES**, если число **n** является точной степенью двойки, или слово **NO** в противном случае. Операцией возведения в степень пользоваться нельзя!
12. По данному целому числу **n** распечатайте все квадраты натуральных чисел, не превосходящие **n**, в порядке возрастания.
13. Напишите программу, которая вводит с клавиатуры натуральное число и находит наибольшую цифру в его десятичной записи.
14. Напишите программу, которая предлагает ввести пароль и не переходит к выполнению основной части, пока не введён правильный пароль. Основная часть – вывод на экран «секретных сведений» (придумайте их сами).
15. Напишите программу, которая вводит с клавиатуры натуральное число и определяет, простое оно или нет. Для этого нужно делить число на все натуральные числа, начиная с 2, пока не получится деление без остатка.
16. Напишите программу, которая в последовательности натуральных чисел определяет минимальное число. Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность). Количество чисел не превышает 1000. Введённые числа по модулю не превышают 30000.
17. Измените программу № 16:
 - а) выводится минимальное число, кратное заданному;
 - б) вычисляется максимальное чётное число последовательности.
18. По данному натуральному $n \leq 100$ вычислите сумму квадратов $1^2 + 2^2 + 3^2 + \dots + n^2$.
19. По данному целому неотрицательному n ($0 \leq n \leq 15$) вычислите значение факториала $n!$.

Напишите программу, которая выводит все нечетные числа от 5 до 55.
20. Напишите программу, которая находит сумму положительных чисел и произведение отрицательных чисел в ряду: -5, -4, -3, ..., 9.
21. Напишите программу, которая в промежутке $[-8, 13]$ находит количество четных отрицательных чисел и сумму нечетных положительных чисел.
22. Задумано трёхзначное число, которое при делении на 15 даёт в остатке 11, а при делении на 11 даёт в остатке 9. Найдите все такие числа.
23. Найдите все пятизначные числа, которые при делении на 133 дают в остатке 125, а при делении на 134 дают в остатке 111.

