Practice 2- Math Operations in Python

Objective: learn basic mathematical operations Python

The Python language, due to the presence of a huge number of libraries for solving various kinds of computational problems, today is a competitor to such packages as Matlab and Octave. Launched in interactive mode, it actually turns into a powerful calculator. In this lesson, we will talk about the arithmetic operations available in this language. We will study arithmetic operations in relation to numbers.

If only integers are used as operands of some arithmetic expression, then the result will also be an integer. An exception is the division operation, which results in a real number. When using integer and real variables together, the result will be real.

This lesson will focus on the arithmetic operations available in this language.

If only integers are used as operands of some arithmetic expression, then the result will also be an integer. An exception is the division operation, which results in a real number. When using integer and real variables together, the result will be real.

Integers (int)

Numbers in Python 3 support a set of common math operations:

x+y	Addition
x-y	Subtraction
x*y	Multiplication
x / y	Division
x // y	Getting the integer part from division
x % y	Remainder of the division
-x	Change the sign of a number
abs(x)	The absolute value of a number
divmod(x, y)	Pair (x // y, x % y)
xy**	Exponentiation

pow(x, y[, z])	x : The number to be raised to a power. y : A number that is the power to raise the first argument to. If the number is negative or one of the numbers "x" or "y" is non-integer, then the argument "z" is not accepted. z : The number by which to modulo. If a number is specified, "x" and "y" are expected to be positive and of type int.

Application example the above operations on integers

```
x = 5
y = 2
z = 3
x+y = 7
x-y = 3
x*y = 10
x/y = 2.5
x//y = 2
x%y = 1
-x= -5
abs(-x) = 5
divmod(x,y) = (2, 1)
x**y = 25
pow(x,y,z) = 1
```

Real numbers (float)

Real numbers support the same operations as integers. However (because of the representation of numbers in a computer) real numbers are inexact and this can lead to errors.

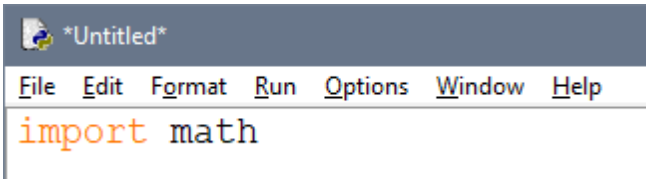
An example of applying the above operations on real numbers

```
x = 5.5
y = 2.3
x+y = 7.8
x-y = 3.2
x*y = 12.649999999999999
x/y = 2.3913043478260874
x//y = 2.0
x%y = 0.9000000000000004
-x= -5.5
abs(-x) = 5.5
divmod(x,y) = (2.0, 0.9000000000000004)
x**y = 50.44686540422945
```

Library (module) math

Python comes standard with the math library, which contains a large number of commonly used math functions.

To work with this module, you must first import it.



Consider the most commonly used functions of the math module

math.ceil(x)	Returns the nearest integer greater than x
mathfabs(x)	Returns the absolute value of the number x
math factorial(x)	Calculates the factorial of x
math floor(x)	Returns the nearest integer less than x
math.exp(x)	Calculates e**x
math.log2(x)	Base 2 logarithm
math.log10(x)	Log base 10
math.log(x[,base])	By default, calculates the logarithm to the base e, optionally you can specify the base of the logarithm
math pow(x, y)	Calculates the value of x to the power of y
math.sqrt(x)	The square root of x

An example of applying the above functions to numbers

The program defines 4variables - a, b, c, d, each ofwhich is either an integer, real, or negative.

The print() command prints the value of each variable to the screen when the program is executed.

The result of executing the function of the math module is placed in the variable z.

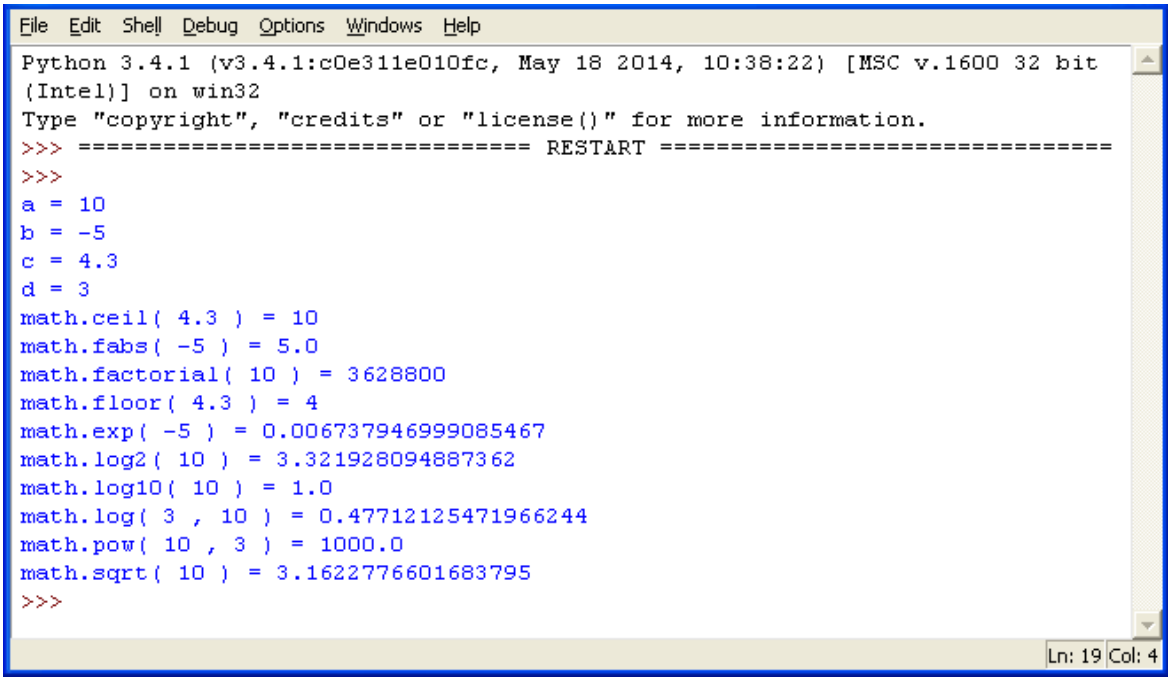
Then the print() command prints a message in the form of the function used and its argument, and the result of its execution.

File Edit Format Run Options Windows Help

```
import math
a=10
b=-5
c=4.3
d=3
print('a =',a)
print('b =',b)
print('c =',c)
print('d =',d)
z=math.ceil(a)
print('math.ceil(',c,') =',z)
z=math.fabs(b)
print('math.fabs(',b,') =',z)
z=math.factorial(a)
print('math.factorial(',a,') =',z)
z=math.floor(c)
print('math.floor(',c,') =',z)
z=math.exp(b)
print('math.exp(',b,') =',z)
z=math.log2(a)
print('math.log2(',a,') =',z)
z=math.log10(a)
print('math.log10(',a,') =',z)
z=math.log(d,a)
print('math.log(',d,',',a,') =',z)
z=math.pow(a,d)
print('math.pow(',a,',',d,') =',z)
z=math.sqrt(a)
print('math.sqrt(',a,') =',z)
```

Ln: 21 Col: 29

Python Program Example



The result of executing the program using the functions of the math module

Trigonometric functions of the math module

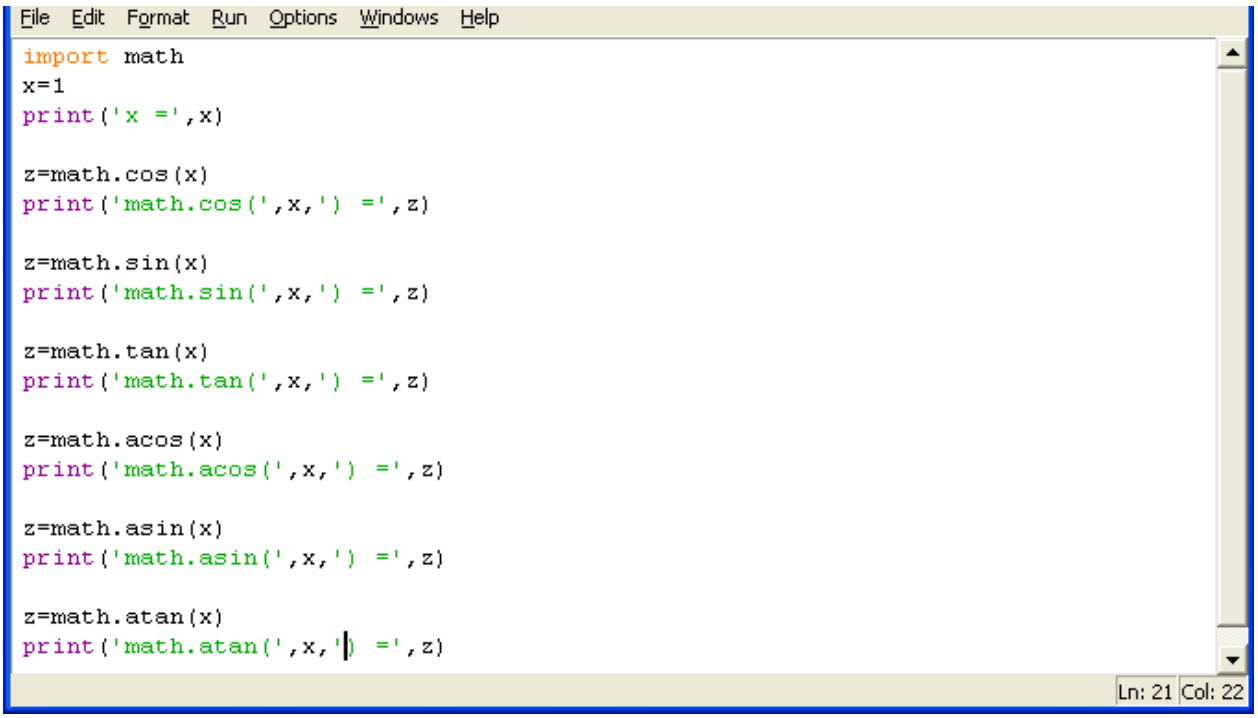
math.cos(x)	Returns the cos of the number X
math.sin(x)	returnssinnumber X
math.tan(x)	returnstannumber X
math.acos(x)	returnsacosnumber X
math.asin(x)	returnsasinnumber X
math.atan(x)	returnsatannumber X

An example of applying the above functions to numbers

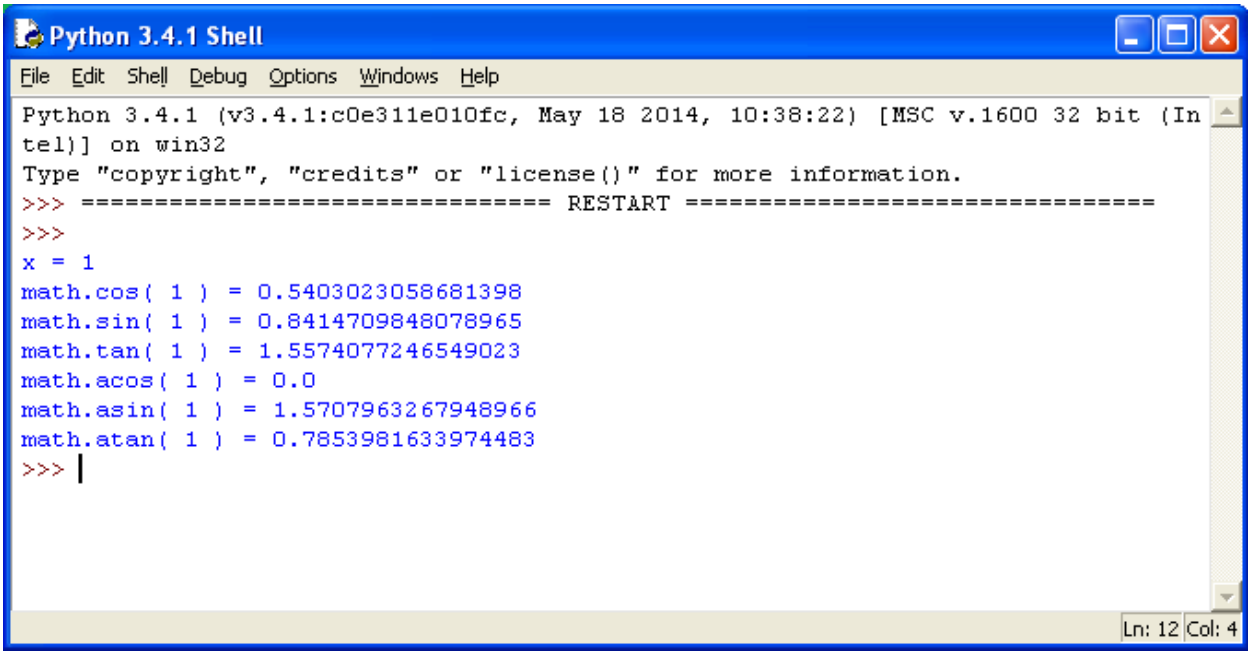
The program defines a variable x containing an integer. The value of the variable is printed by the print() command to the screen.

The result of executing the trigonometric function of the math module is placed in the variable z.

Then the print() command outputs a message in the form of the function used and its argument and the result of its execution.



An example program using the trigonometric functions of the math module



The result of the program execution using the trigonometric functions of the math module

Constants:

- **math.pi**- Pi number.
- **math.e**- number e (exponent).

Example

Write a program that would evaluate the given arithmetic expression given the given variables. Variables are entered from the keyboard. Output the result with 2 decimal places.

Exercise

$$Z = \frac{9\pi t + 10\cos(x)}{\sqrt{t} - |\sin(t)|} * e^{x}$$

x=1
0; t=1

Solution

First, we import the math module. To do this, we use the import math command.

Then you should enter the values of two variables of integer type x and t.

To enter data, the input command is used, but since integers are given in the condition, you must first determine the type of variables: x=int(), t=int().

Having determined the type of variables, you should enter them, for this, in the brackets of the int () command, you need to write the input () command.

For variable x, it looks like this: x=int(input("message when entering a value")).

For the variable t, the same is true: t=int(input("message when entering a value")). The next step is to compose an arithmetic expression, the result of which we will place in the variable z.

Let's set up the numerator first. It will look like this: 9*math.pi*t+10*math.cos(x).

Then you need to compose the denominator, while paying attention to the fact that the numerator is divisible by the denominator, therefore, both the numerator and the denominator must be placed in brackets (), and a division sign / should be written between them.

It will look like this: (9*math.pi*t+10*math.cos(x))/(math.sqrt(t)- math.fabs(math.sin(t))).

The last step is to multiply the fraction by the exponent to the x power.

Since the whole fraction is multiplied, the composed expression should be placed in brackets (), and only then write the function math.pow(math.e,x).

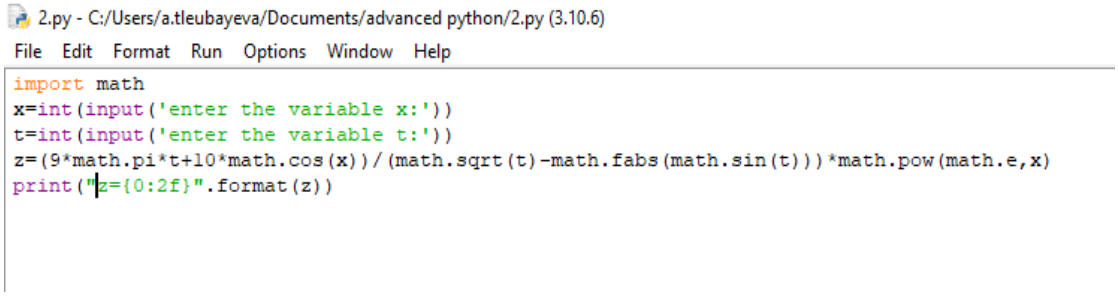
As a result, the expression will look like:
z=((9*math.pi*t+10*math.cos(x))/(math.sqrt(t)-
math.fabs(math.sin(t)))) *math.pow(math.e,x).

When compiling this expression, you should pay attention to the number of opening and closing brackets.

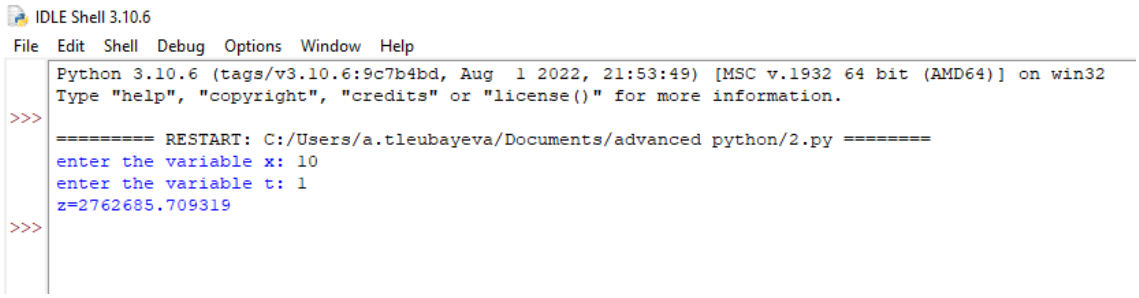
Using the print() command, print the value of the variable, formatting it with the format command.

The format itself is written in apostrophes in curly braces {}.

In the task, you need to output a number with two decimal places, so the format will look like this: {0:.2f}, where 2 is the number of decimal places, and f indicates that a real number is being formatted. In this case, before 2, you need to put a dot, thereby indicating that we are formatting the fractional part of the number.



Result



Play the task from the example. Take screenshots of code and results.