

MATH

The Python Bakery

Computers Are Good at Math



In some ways, computers are just large machines for doing math.

Computers are very, very good at doing math.

In fact, modern computers can perform billions or even trillions of calculations per second.

However, the math they do is extremely simplistic, the kind that you learned as a child.

Fortunately, programs can use these simple operations together to build up more complicated operations.

Operators

- + Addition
- - Subtraction
- * Multiplication
- / Division
- // Integer Division
- ** Exponentiation
- % Modulo

There are seven basic mathematical operators in Python that you should be familiar with:

Addition, Subtraction, Multiplication, Float Division, Integer Division, Exponentiation, and Modulo.

Addition and Subtraction

```
# Addition
print(1 + 1)
print(-10 + 10)
print(5.0 + 5)

# Subtraction
print(5 - 8)
print(10 - 3.4)
```

The + operator is used for addition, and the - operator is used for subtraction. The value to the left and right of the operator are the "operands". This mostly works just like math you've always seen.

Multiplication and Exponentiation

```
# Multiplication
print(4 * 2)
print(-5 * 5)
print(.5 * 10)
print(.5 * 4)

# Exponents
print(2 ** 4)
print(10 ** .5)
```

One asterisk (*) is used for multiplication.

Two asterisks (**) are used for exponents (also known as powers).

Division

```
# Float division
print(3 / 12)
print(1 / 1)
print(2.5 / 5)

# Integer division
print(2.5 // .5)
print(2.5 // 5)
print(4 // 3)

# Oops, error!
print(1 / 0)
```

The forward slash is used for float division.

Two forward slashes are used for integer division.

When you divide two integers with float division, you will ALWAYS get a floating point number.

I'll say that again - whenever you do regular float division with a single slash, the result will always be a float!

If you want to remove the decimal value, you can use two forward slashes to truncate the result - then the result will always be equal to an integer.

Keep in mind, you cannot divide by 0!

Modulo

```
print(16 % 12)
print(4 % 2)
print(5 % 2)
print(6 % 2.1)
print(12.5 % .5)
```

You have probably never heard of the Modulo operator, but programmers use it occasionally: modulo calculates the remainder after division.

Modulo is sometimes called "Clock Arithmetic", because it makes it easy to figure out time.

If someone said "It is 16pm", you could do "16 modulo 12" and find out they meant "4pm".

However, modulo has many other uses, such as for figuring out if numbers are even or odd.

Any number modulo two will result in one if the number was odd, or zero if the number was even.

Using the Operators

```
print(2 + 3 + 4)
```

When you run a program with operators, Python will do the math and then replace the result.

You will not be able to see the computer do the math by looking at just the code!

The result will seem to appear in the console immediately.

However, the computer did not execute the code instantly, but instead had to first breakdown the operations into smaller steps.

First, the computer will add 2 and 3. Then the computer will add the resulting 5 to 4.

The final result of 9 is what is printed to the user, without any of the intermediate results being shown.

Order of Operations

- Please (Parentheses)
- Excuse (Exponents)
- My (Multiplication), Most (Modulo), Dear (Division)
- Aunt (Addition), Sally (Subtraction)

When you use operators, there is an order of operations.

Whenever there's a tie, the left-most operation happens first.

Notice that you can always override any ordering by using parentheses.

You might already be familiar with this order: if not, here is a useful mnemonic.

PEMMDAS: Please Excuse My Most Dear Aunt Sally.

Each word of the mnemonic references the order of the operation.

The words on the same line shown here are meant to have the same priority.

Floats and Integers

```
# Integers make integers
print(1 + 1)

# Any floats make floats
print(1 + 1.0)
print(1.0 - 1)

# Always integer
print(5.5 // .3)

# Always float
print(6 / 3)
```

You can recall that floats are decimal numbers and integers are whole numbers. When any operation involves any floats, the final result will always be a float, even if it could be expressed as a whole number. The only exception to this rule is when you do division. Remember, integer division will always produce an integer and float division will always produce a float.