# Keywords

A keyword is a reserved word in a programming language that performs a specific purpose. In your first Python example, you briefly encountered the keywords **for** and **in**. Note that keywords will often appear in **bold** in this course.

In the next few weeks, you will also learn the following keywords:

Values: **True**, **False**, **None** Conditions: **if**, **elif**, **else** Logical operators: **and**, **or**, **not** Loops: **for**, **in**, **while**, **break**, **continue** Functions: **def**, **return**

You don't need to learn this whole list now. We'll dive into each keyword as we encounter them. There are additional reserved keywords in Python. If you would like to read about them, please visit the linked “Python Keywords” article in the Resources section at the end of this study guide.

# Arithmetic operators

Python can calculate numbers using common mathematical operators, along with some special operators, too:

**x + y**            Addition + operator returns the sum of x plus y **x - y**             Subtraction - operator returns the difference of x minus y **x \* y**            Multiplication \* operator returns the product of x times y **x / y**             Division / operator returns the quotient of x divided by y **x\*\*y**            Exponent \*\* operator returns the result of raising x to the power of y  **x\*\*2**            Square expression returns x squared **x\*\*3**            Cube expression returns x cubed **x\*\*(1/2)**   Square root (½) or (0.5) fractional exponent operator returns the square root of x **x // y**           Floor division operator returns the integer part of the integer division of x by y **x % y**          Modulo operator returns the remainder part of the integer division of x by y

## Order of operations

The order of operations are to be calculated from left to right in the following order:

1. **P**arentheses ( ), { }, [ ]
2. **E**xponents xy   (x\*\*y)
3. **M**ultiplication \* and **D**ivision /
4. **A**ddition + and **S**ubtraction -

You might find the **PEMDAS** mnemonic device to be helpful in remembering the order.

# Resources for more information

For more information about the concepts covered in this reading, please visit:

* [Built-in Functions](https://docs.python.org/3/library/functions.html) - Lists and summarizes Python’s built-in functions.
* [Python Keywords](https://www.w3schools.com/python/python_ref_keywords.asp) - Lists Python’s reserved keywords and a brief description of what each keyword does.
* [Different Arithmetic operators in Python](https://flexiple.com/python/arithmetic-operators-in-python/) - Provides more examples of the proper syntax for using arithmetic operators in Python.

For additional Python practice, the following links will take you to several popular online interpreters and codepads:

* [Welcome to Python](https://www.python.org/shell/)
* [Online Python Interpreter](https://www.onlinegdb.com/online_python_interpreter)
* [Create a new Repl](https://repl.it/languages/python3)
* [Online Python-3 Compiler (Interpreter)](https://www.tutorialspoint.com/execute_python3_online.php)
* [Compile Python 3 Online](https://rextester.com/l/python3_online_compiler)
* [Your Python Trinket](https://trinket.io/python3)