**Intro:**Python is:

* a general-purpose scripting language.
* a popular language used to code a variety of applications.
* a frequently used tool for automation.
* a cross-platform compatible language.
* a beginner-friendly language.

Python is not:

* a platform-specific / OS-specific scripting language.
* a client-side scripting language.
* a purely object-oriented programming language.

**Key Terms**

* **Programming code** - Programming code is a set of written computer instructions, guided by rules, using a computer programming language. It might help to think of the computer instructions as a detailed, step-by-step recipe for performing tasks. The instructions tell computers and machines how to perform an action. Programming code may also be referred to as source code or scripts.
* **Programming languages** - Programming languages are like human spoken languages in that they both use syntax and semantics. Programming languages are used to write computer programs.  Some common programming languages include Python, Java, C, C++, C#, and R.
* **Syntax** - Syntax is a set of rules for how statements are constructed in both human and computer languages. Programming syntax includes rules for the order of elements in programming instructions, as well as the use of special characters and their placements in statements. This concept is like the syntax rules for grammar and punctuation in human language.
* **Semantics** - Semantics refers to the intended meaning or effect of statements, or collections of words, in both human and computer languages. Semantic errors are also referred to as logical errors.
* **Computer program** - A computer program is a step-by-step list of instructions that a computer follows to reach an intended goal. It is important to be clear and precise about the actions a computer program is supposed to perform because computers will do exactly what they are instructed to do. Computer programs can be long, complex, and accomplish a variety of tasks. They are often developed by computer programmers and software engineers, but anyone can learn to create them. Computer programs may involve a structured development cycle. They can be written in a wide variety of programming languages, such as Python, Java, C++, R, and more. The completed format of a program is often a single executable file.
* **Script** - Scripts are usually shorter and less complex than computer programs. Scripts are often used to automate specific tasks. However, they can be used for complex tasks if needed. Scripts are often written by IT professionals, but anyone can learn to write scripts. Scripts have a shorter, less structured development cycle as compared to the development of complex computer programs and software. Scripts can be written in a variety of programming languages, like Python, JavaScript, Ruby, Bash, and more. Some scripting languages are interpreted languages and are only compatible with certain platforms.
* **Automation** - Automation is used to replace a repetitive manual step with one that happens automatically.
* **Output** - Output is the result of a task performed by a function or computer program. Output can include a single value, a report, entries into a database, and more.
* **Input** - Input is information that is provided to a program by the end user. Input can be text, voice, images, biometrics, and more.
* **Functions** - A function is a reusable block of code that performs a specific task.
* **Variables** - Variables are used to temporarily store changeable values in programming code.
* **Platform-specific / OS specific scripting language** - Platform-specific scripting languages, like PowerShell (for Windows) and Bash (for Linux), are used by system administrators on those platforms.
* **Client-side scripting language** - Client-side scripting languages, like JavaScript, are used mostly for web programming. The scripts are transferred from a web server to the end-user’s internet browser, then executed in the browser.
* **Machine language** - Machine language is the lowest-level computer language. It communicates directly with computing machines in binary code (ones and zeros). In binary code, one equals a pulse of electricity and zero equals no electrical pulse. Machine language instructions are made from translating languages like Python into complex patterns of ones and zeros.
* **Cross-platform** **language** - Programming language that is compatible with one or more platforms / operating systems (e.g., Windows, Linux, Mac, iOS, Android).
* **Object-oriented programming language** - In object-oriented programming languages, most coding elements are objects with configurable properties. For example, a form field is an object that can be configured to accept only dates as input in the mm/dd/yy format and can be configured to read from and write to a specific database.
* **Python interpreter -** An interpreter is the program that reads and executes Python code by translating Python code into computer instructions.

**Resources**

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

* [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)

# Functions

A function is a piece of code that performs a unit of work. In the examples you've seen so far, you have only encountered the **print()** function, which outputs a message to the screen. You will use this function frequently in this course to check the results of your code. The syntax of the print() function is modeled in the example below.

# Syntax for printing a string of text. Click Run to check the result.

print("Hello world!")

Hello world!

# 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\*\*e**            Exponent \*\* operator returns the result of raising x to the power of e
* **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 xe   (x\*\*e)
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.