Glossary

**activecode**

A unique interpreter environment that allows Python to be executed from within a web browser.

**algorithm**

A general step by step process for solving a problem.

**bug**

An error in a program.

**byte code**

An intermediate language between source code and object code. Many modern languages first compile source code into byte code and then interpret the byte code with a program called a *virtual machine*.

**codelens**

An interactive environment that allows the user to control the step by step execution of a Python program

**comment**

Information in a program that is meant for other programmers (or anyone reading the source code) and has no effect on the execution of the program.

**compile**

To translate a program written in a high-level language into a low-level language all at once, in preparation for later execution.

**debugging**

The process of finding and removing any of the three kinds of programming errors.

**exception**

Another name for a runtime error.

**executable**

Another name for object code that is ready to be executed.

**formal language**

Any one of the languages that people have designed for specific purposes, such as representing mathematical ideas or computer programs; all programming languages are formal languages.

**high-level language**

A programming language like Python that is designed to be easy for humans to read and write.

**interpret**

To execute a program in a high-level language by translating it one line at a time.

**low-level language**

A programming language that is designed to be easy for a computer to execute; also called machine language or assembly language.

**natural language**

Any one of the languages that people speak that evolved naturally.

**object code**

The output of the compiler after it translates the program.

**parse**

To examine a program and analyze the syntactic structure.

**portability**

A property of a program that can run on more than one kind of computer.

**print function**

A function used in a program or script that causes the Python interpreter to display a value on its output device.

**problem solving**

The process of formulating a problem, finding a solution, and expressing the solution.

**program**

A sequence of instructions that specifies to a computer actions and computations to be performed.

**programming language**

A formal notation for representing solutions.

**Python shell**

An interactive user interface to the Python interpreter. The user of a Python shell types commands at the prompt (>>>), and presses the return key to send these commands immediately to the interpreter for processing.

**runtime error**

An error that does not occur until the program has started to execute but that prevents the program from continuing.

**semantic error**

An error in a program that makes it do something other than what the programmer intended.

**semantics**

The meaning of a program.

**shell mode**

A style of using Python where we type expressions at the command prompt, and the results are shown immediately. Contrast with **source code**, and see the entry under **Python shell**.

**source code**

A program, stored in a file, in a high-level language before being compiled or interpreted.

**syntax**

The structure of a program.

**syntax error**

An error in a program that makes it impossible to parse — and therefore impossible to interpret.

**token**

One of the basic elements of the syntactic structure of a program, analogous to a word in a natural language.



 [**Next Section - CHAPTER 2**](https://learn.launchcode.org/runestone/static/thinkcspy/ChapterHeadings/Heading2.html)

Mark as completed