

# Glossary

## argument

A value provided to a function when the function is called. This value is assigned to the corresponding parameter in the function. The argument can be the result of an expression which may involve operators, operands and calls to other fruitful functions.

## body

The second part of a compound statement. The body consists of a sequence of statements all indented the same amount from the beginning of the header. The standard amount of indentation used within the Python community is 4 spaces.

## compound statement

A statement that consists of two parts:

1. header - which begins with a keyword determining the statement type, and ends with a colon.
2. body - containing one or more statements indented the same amount from the header.

The syntax of a compound statement looks like this:

```
keyword expression:
    statement
    statement ...
```

## docstring

If the first thing in a function body is a string (or, we'll see later, in other situations too) that is attached to the function as its `__doc__` attribute.

## flow of execution

The order in which statements are executed during a program run.

## function

A named sequence of statements that performs some useful operation. Functions may or may not take parameters and may or may not produce a result.

## function call

A statement that executes a function. It consists of the name of the function followed by a list of arguments enclosed in parentheses.

## function composition

Using the output from one function call as the input to another.

## function definition

A statement that creates a new function, specifying its name, parameters, and the statements it executes.

**fruitful function**

A function that returns a value when it is called.

**header line**

The first part of a compound statement. A header line begins with a keyword and ends with a colon (:)

**import statement**

A statement which permits functions and variables defined in another Python module to be brought into the environment of another script. To use the features of the turtle, we need to first import the turtle module.

**lifetime**

Variables and objects have lifetimes — they are created at some point during program execution, and will be destroyed at some time.

**local variable**

A variable defined inside a function. A local variable can only be used inside its function. Parameters of a function are also a special kind of local variable.

**parameter**

A name used inside a function to refer to the value which was passed to it as an argument.

**refactor**

A fancy word to describe reorganizing your program code, usually to make it more understandable. Typically, we have a program that is already working, then we go back to “tidy it up”. It often involves choosing better variable names, or spotting repeated patterns and moving that code into a function.

---

© Copyright 2013 Brad Miller, David Ranum, Created using Runestone Interactive.