Lesson 3.01: Built in Functions/Imported Functions

# Learning Objectives

* Define and identify: Function, calling, importing, returning, abstraction
* Recognize and re-assess functions they have encountered so far
* Utilize built in functions and imported functions
* Describe the process of using a function and getting a value back

# Materials/Preparation

* Lab handout
* Read through the handout so that you are familiar with the requirements and can assist students

# Pacing Guide

|  |  |
| --- | --- |
| Duration | Description |
| 5 Minutes | Welcome, attendance, bell work, announcements |
| 10 Minutes | Lecture |
| 30 Minutes | Lab |
| 10 Minutes | Discussion |

# Instructor’s Notes

1. Lecture
   1. Functions
      1. Definition: *in the context of programming, a function is a named sequence of statements that performs a computation. When you define a function, you specify the name and the sequence of statements. Later, you can “call” the function by name. We have already seen some examples:* 
         1. print(“123”), type(32.0), len([“cats”, “dogs”, “rabbits”])
      2. Built in Functions.
         1. Arguments: values that a function takes in
            1. Len takes in a list
            2. Functions take in more than 1 value separate arguments by a comma
         2. Return: the value the function gives back
            1. Len returns a int representing the length of the list
         3. Abstraction
            1. We don’t need to know the stuff that happens to find the length.
      3. Imported functions
         1. Some functions are not included as built-in. For instance, how would you pick a random number from 1 to 5? Python language comes with a file where all these functions are written out for you. To access them all you have to do is *import* the module
            1. Module: *a file that contains a collection of related functions*
         2. Once you’ve imported the module you use the syntax module\_name.function\_name(arg1, arg2…)
            1. randint is defined in the random module and takes in two arguments: the first is the lower bound (inclusive) on the random integer and the second is the upper bound(exclusive)

>>> import random

>>> random.randint(0, 4)

3

1. Lab
   1. 8 ball
      1. have students make a list of 8-ball responses
      2. use randint to pick a random response
      3. print the 8-balls response to the user
   2. make sure students are using length, and properly 0-indexing
2. Discuss
   1. Go over solutions, talk about how could use this to randomize list of anything
   2. Could check validy of the random function if you wanted. Should distribute out evenly