122COM: Sync lesson

David Croft

February 5, 2016

Abstract

This week we are going to be working on some simple Python tasks to get back in the right frame of mind and sync up the lectures and labs for all the cohorts.

This week will be taught in Python only.

1 Introduction

The source code files for this lab and the lecture are available at:

`https://github.com/covcom/122COM_recap.git`

`git@github.com:covcom/122COM_recap.git`

2 Activities

In future weeks you will often find yourself modifying or expanding on existing code. This code will be hosted on a github repository for that week. Some of this code will come with tests that make sure your code is working correctly. This sort of automated testing will be use regularly throughout this module and this week's tasks are intended to get you used to it.

This week you can do the tasks in any order but try to do as many as possible.

Pre-lab work:

1. Download this weeks code off github.
   - I strongly recommend that you investigate how to clone git repositories.
2. Complete the `leap_year` function in the `lab_leap` program.
   - While my code is using an imported module to test if a year is leap or not, you've got to implement the logic from scratch.
   - You may want to research what the correct leap year logic is.

Lab work:

3. Complete the `main()` and `find_type()` functions in `lab_type` so that the program reads in input from the user and identifies if the input is an integer, string or float.
   - The program should print a message identifying the type.
   - Since the python `input()` function returns a string regardless of what the user enters, the function `find_type()` should identify if that string contains an int, float etc.
   - `find_type()` should return a string of either `'int'`, `'float'` or `'string'`.
   - E.g. `find_type('12345')` = `'int'`, `find_type('3.14')` = `'float'`
4. Extend your program to identify boolean inputs as well.
   - Should count as boolean if the user enters "true", "false", "True", "fAlSe" etc.

Extended work:

5. Complete `lab_morse`. The program should be able to translate text to and from morse code.
   - We're only going to worry about capital letters A→Z, numbers 0→9 and spaces.
   - You may need to research morse code.
   - Think carefully about the best types of data structures for this task.