Competency 200 – Python Getting Started - Logic

Plan

Leader: L Shumlich

Date: 2019/02

Overview & Purpose

Set up the Python environment and perform basic logic.

Prerequisite

- Review of the JavaScript Competencies and Practise Exercises. Larry will do a technical interview to review those skills and will ask for supplemental exercises.
- This exercise must be done in groups of two or three. Talk to Larry about getting in a group.
- Udemy Python and Flask Bootcamp - Skim Section 1, 2, 3, 4 - Complete Section 5 Python Level One. Complete the install in Section 1, Chapter 4 for Anaconda Python.

References

- Udemy - Python and Flask Bootcamp
- https://www.python.org/

- https://www.w3schools.com/python/default.asp
- https://www.python.org/dev/peps/

Exercise - Install Python - did it work?

This should have been done as part of the Udemy course Python and Flask Bootcamp, Section 1, Chapter 4 - Installing and Setup. This installation takes 5 Gig of disk space.

https://www.anaconda.com/ Follow the instructions.

Now verify the install has worked successfully.

- print “Hello World” on the console

Exercise - Basic Syntax

This is the second programming language that we have encountered in this program. The syntax is different than JavaScript but the concept is the same. Both languages are OO ish and both languages are interpretive.

Review the new coding style as described: https://www.python.org/dev/peps/

Go to “Comp 100” - “Exercise - Basic Syntax” and code the same exercise in Python. Create a file called evolveu\python\syntax.py. When complete, compare the JavaScript code to the Python code. Notice the differences. Notice the Similarities.

Exercise - TDD

Python has a sophisticated testing framework. Research the framework and use it as much as possible. Setup a few examples.

Exercise - Simple Functions

Write a function called “email”. This function will receive two parameters: first name and last name. It will return a well formatted email. For example:

Larry Shumlich ⇒ larry.shumlich@evolveu.ca

Heiko Peters ⇒ heiko.peters@evolveu.ca

Write the code in a function and call the function for each person you want to format the email for. Write an automated test that will check that the results are what you expect.

Email the test to larry before you write the code.

Write the tax function from the JavaScript exercise in Comp 100 - “Exercise - Canadian Taxes”. Hard code some input. Create the tests first. Copy and reuse as much code as possible: https://www.canada.ca/en/revenue-agency/services/tax/individuals/frequently-asked-questions-individuals/canadian-income-tax-rates-individuals-current-previous-years.html#federal

Review each others code for quality of code. Same rules from JavaScript apply for Python.