Lab 0: Environment, Python, and Testing!

Lab 0 is intended to make sure you have a working development environment, can download starter code from GitHub, test your code, and **commit and push** files to GitHub.

Part 0

Determine and set up your code development and test environment. If you choose to develop code on your own machine using an IDE, remember that the tests we run will be in the server environment.

Part 1

Download (clone) the planets.py and planets_tests.py files from GitHub to your development environment.

In the plantets.py file, add code to implement the functionality shown by the following sample run:

Sample Run:

```
What do you weigh on earth? 136

On Mars you would weigh 51.68 pounds.
On Jupiter you would weigh 318.24 pounds.
```

Important Information and Requirements:

- To calculate a person's weight on Mars, multiply their weight on earth by 0.38.
- To calculate a person's weight on Jupiter, multiply their weight on earth by 2.34.
- Initially, execute the planets.py and confirm that your output looks like the sample run after you have added the necessary code to the weight_on_planets function.
- After confirming the sample run, then run the unit tests in planets_tests.py to check your solution for Mars. Note there is also a test to check that the exception is raised when the input string does not match a planet in the function.
- Add another test to planets_tests.py for Jupiter
- Add another planet, Venus (multiply weight by 0.91), to the weight_on_planets function, and include a test in planet_tests.py to check the new code.
- For this lab, no need to check for valid input of weight, can assume it will be a positive number
- **Commit and push** your modified planets.py and planets_tests.py files to GitHub.

For folks new to Python or who need review

The basic Python Tutorial
https://docs.python.org/3/tutorial/

This site gives the essentials for those familiar with Java.
Python for Java Programmers: http://python4java.necaiseweb.org/Fundamentals/Fundamentals

If you want to use an IDE, you are welcome to. If you do not already have an IDE that you are familiar with, PyCharm is a good choice for Python development.
PyCharm IDE: https://www.jetbrains.com/pycharm/

Videos on specific topics.

Installing Python on Mac/Windows: https://www.youtube.com/watch?v=YYXdXT2l-Gg&list=PL- osiE80TeTt2d9bfVyTiXJA-UTHn6WwU

Strings: https://www.youtube.com/watch?v=k9TUPpGqYTo&list=PL-osiE80TeTt2d9bfVyTiXJA-UTHn6WwU&index=2

Conditionals: https://www.youtube.com/watch?v=DZwmZ8Usvnk&list=PL-osiE80TeTt2d9bfVyTiXJA- UTHn6WwU&index=6

Loops: https://www.youtube.com/watch?v=6iF8Xb7Z3wQ&list=PL-osiE80TeTt2d9bfVyTiXJA- UTHn6WwU&index=7

Functions: https://www.youtube.com/watch?v=9Os0o3wzS_I&list=PL-osiE80TeTt2d9bfVyTiXJA-UTHn6WwU&index=8

Modules: https://www.youtube.com/watch?v=CqvZ3vGoGs0&list=PL-osiE80TeTt2d9bfVyTiXJA- UTHn6WwU&index=9

Interactive Interpreter

The Python interpreter can be used in an interactive mode. In this mode, you will be able to type a statement and immediately see the result of its execution. Interactive mode is very useful for experimenting with the language and for testing small pieces of code, but your general development process with be editing and executing a file as discussed previously.

Start the interpreter in interactive mode by typing **python** at the command prompt. You should now see something like the following.

```
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 07:18:10) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

The >>> is the interpreter's prompt. You can type an expression at the prompt to see what it evaluates to. Type each of the following (hit enter after each one) to see the result. When you are finished, you can exit the interpreter by typing ctrl-D (i.e., hold the control key and hit d).

- 0 + 1
- 2 * 2
- 19 // 3
- 19 / 3
- 19 / 3.0
- 19.0 // 3.0
- 4 * 2 + 27 // 3 + 4
- 4 * (2 + 27) // 3 + 4