# Assignment #2

## Objective

Learn commenting, numbers, math, variables, strings, and printing.

## Due

Tuesday Jan 26th at 9am. Remember to turn in your weekly participation log as well.

## Background

In class, you setup a GitHub repository and prepared it for submission of all of your future assignments. Form this point onward all assignments will be turned in via GitHub. Please commit to your GitHub repository all the completed exercises from this assignment in a directory named ‘assigment2’ and send the URL for your GitHub repository to Dr. Ficklin in a private slack message before the assignment due date. Full credit will be earned upon completion of the assignment regardless of correctness.

This assignment will take a good amount of time. Please try to get started as soon as possible to ensure you have plenty of time to work through it***. Avoid trying to do the entire thing at one time***. Doing a few exercises and taking a break will help with memory retention. If you have questions, use Slack! Remember to post questions to the #assignments channel to give anyone the chance to respond.

If you find you are not able to work through everything because it is taking too long, please let the instructor know. If most of the students are struggling to finish the work in a reasonable amount of time, then expectations can be adjusted. Not everyone will pick up programming at the same pace. You may find that you devote more or less time than other students.

**Tasks**

1. From Shaw’s *Learning Python 3 the Hard Way*, read and complete exercises #1-9 (pages 34-62).
   1. Create the programs the author instructs you to.
   2. Make sure that your output is like that shown in the “***What You Should See***” section that appears after each exercise. If you do not have the same results, then check your code, fix any mistakes and re-run until your output looks like those shown.
   3. At the end of each exercise is a “***Study Drills***” section.
      1. Some items in the Study Drills are meant to help you think through what you are doing. For these, take some time to think about the questions, but you do not need to write down answers or turn them in.
      2. If the Study Drill instructs you to write code, then please do so as instructed and turn that in.
      3. For some of the questions in the Study Drills section you may have to do a bit of web searching to find answers. For example, Exercise #3 Study Drill question #4 asks you to convert your script to use floating point numbers. A floating-point number is not described by the author, so you may have to do a bit of searching to find out. Do not spend too much time on these if you cannot figure them out. Write them down to ask in class or ask for help on Slack.
      4. For Exercise #10 you do not need to make flash cards to help you memorize.
   4. For the sake of time you can skip the ‘***Break It***’ sections when they appear.
   5. Finally, avoid the urge to skip over the “***Common Student Questions***” section. The answers to some of those questions may help you out!
   6. Notes:
      1. We discussed the topics of Python setup and text editor usage in class. The beginning of Exercise #1 indicates you should review Exercise #0 before proceeding, however we covered this material in class.
      2. **Be sure to save each exercise into a separate file**. Each exercise indicates the name you should use (e.g. ex1.py for the first exercise).
      3. With the Shaw text it is easy to cut-and-paste the code and execute it. But as the author points out this does little to teach. **Please take the time to type the code**.