**Journal**

Don't forget to fill up the Weekly Journal: <https://forms.gle/jNTr1VU9MV716Keb8>  **Done**

**Basic (same as last week, but now do the git lesson at least until part 5)**

1. Install git from <https://git-scm.com/downloads> **Done**
2. Start reading git material from <http://swcarpentry.github.io/git-novice/>. Feel free to complete all parts, but you must finish at least until part 5. You do NOT need to send me the exercises, just **send me a message telling me up to which lesson you completed. Done fully**

**Advanced**

1. Using your GitHub account that you sent me on the first class, follow the steps 1 to 6 on this [page](https://help.github.com/en/github/getting-started-with-github/create-a-repo) (you don’t need to do the “Commit your first change” part). Make sure to use a public repository on item 4. Send me the link of your repository! <https://github.com/CI1100/CEBD_1160-Homework_3>
2. Now that we know a few tricks with shell, why would we need to learn Python? Make a quick research comparing the two approaches (Shell vs Python), and which technology is best for what. **Send me a summary of your conclusions (no more than a paragraph)** and we’ll debate this in the next class

Exploring several resources in the internet I discovered that most of developers use two technologies and both have their own strengths and weaknesses. Below I state the main points came into my attention. First one is Shell Scripting is simple, but it’s not as powerful as Python (see point 3). Second one is that the start-up time of a Shell Script is 2.8 mili seconds while that of Python is 11.1 mili seconds. But when dealing with large programs, Shell will keep on getting complicated whereas Python does not. The third and the largest is that Shell Scripting works flawlessly and fast when dealing with files (by handling files: copying, cloning disks, writing backup apps for networking, storing file inputs and accessing them later and transferring those outputs later on to something else with the help of pipe) whereas Python on the other hand is more useful for dealing with chunks of data such as reading data from a file and processing data. One of the main strength of Shell programming is excellent in piping out the output of one command into another.

**Reach**

1. Send me what does this shell script do in a message:

   #!/bin/bash Comment line

   for DIR in 0 1 2 S Loop – repeat for variable DIR in the list of 0 1 2 S

   do do commands

     mkdir ${DIR}-files create directory Dirname-files

     touch ${DIR}-files/$DIR create empty file in created

above directory

   done

**Optional Homework Challenge [Ungraded]**

* 1. Reproduce the crontab exercise we did in class: save the current time **every minute** to a file in your Desktop. You can get the current time using the `date` command from shell.
  2. What would the crontab entry look like if we wanted to run it every 8 hours, everyday?