**Code Editors and Virtual Environments**

1. **There are many code editors, they all have their all quirks and features** 
   1. Some are built for a specific language
   2. Vscode, atom, notepad++
2. **Terminal/command line/shell** 
   1. Command line is used for opening programs
   2. Computers used to only have terminals and command lines to get someplace
   3. Git – program to download for windows; has its own command line – git-bash?
3. **Compiler/interpreter** 
   1. When you write programs; they are text files; but the computer doesn’t understand these codes innately; what humans write is called source code – and it needs to be translated into machine code for the computer
   2. That is what the compiler does
4. **Debugger** 
   1. An add on to….
5. **Test runner** 
   1. Advanced topic
6. **Version control system** 
   1. Git via github
7. **Integrated development environment (IDE)** 
   1. Bundles everything above except the operating system
   2. E-clipse, xcode
   3. Big huge bloated mess (too many features)
   4. No sharp line between what’s a code editor and what’s a IDE; it’s a spectrum
8. **Code editor** 
   1. Code is just text
   2. You could write code in any program that can produce text
   3. **Most code editors include a whole bunch of additional features like** 
      1. Syntax highlighting
      2. Integrated file explorer
      3. Integrated terminal/command line
      4. Automatic code completion
      5. Linters “automatically fixes typos/formatting”
9. **Terminal. Shell/ command** 
   1. You do not need to learn how to use the terminal/ command line
   2. It is just more efficient
10. **Version control (git)** 
    1. Most important skill for someone wanting to be a professional coder
    2. Not the same as github
    3. Free-open sourced
    4. Owned by Microsoft

**Python Virtual Environments**

1. *“which” command shows us the version of python that will be used if typing python into the command line*
2. *“python --version” command shows us what version of python is active*
3. **Virtual environment** 
   1. Created to deal with having multiple pythons on your computer
   2. Isolates current program
   3. Future proofs program
   4. Should be set up on a project by project basis