Exercise 1.1: Getting Started with Python

Learning Goals

- Summarize the uses and benefits of Python for web development
- Prepare your developer environment for programming with Python

Reflection Questions

- In your own words, what is the difference between frontend and backend web development? If you were hired to work on backend programming for a web application, what kinds of operations would you be working on?
  - Frontend web development involves the front-facing *visual* aspects of a web app. This includes all of the things the user sees and interacts with, such as buttons, menus, images, objects, and all the stylings and design that go along with it.
  - Back end web development involves all of the logic that determines what information the front end will be accessing to provide the user experience with data and storage, as well as all other programming logic that supplies the app with tools and data, rather than the rendering of the app on the user's screen. Examples include data bases for the reading and storing of information that users provide, share, and read; as well as security and authorization programs for registration and logins of users.
  - If hired to be a backend web developer, I would most likely be working with databases that store information that users access with searches that are relevant to the what information and services the app provides. I would also probably be developing and maintaining API's that structure how the data will be accessed by the front end development team. Anther common backend duty is the constant improvement and maintenance of the security of the app: User registrations, logins, profile management, secure file sharing, etc.

- Imagine you're working as a full-stack developer in the near future. Your

team is asking for your advice on whether to use JavaScript or Python for a project, and you think Python would be the better choice. How would you explain the similarities and differences between the two languages to your team? Drawing from what you learned in this Exercise, what reasons would you give to convince your team that Python is the better option? *(Hint: refer to the Exercise section "The Benefits of Developing with Python")*

- I would argue that because Python is just as well maintained and community-supported as any major framework, that it is a safe language to commit to. With that comes a wealth of both open-source and proprietary packages that are as easy to employ as they are to use. It also comes with a comprehensive set of essential operations for web development. Finally, and possibly *most* importantly is the simple yet effective format and readability of the code. This helps with work flow, integrating new employees, and reducing both the volume of errors as well as the time needed to find and fix errors.

- Now that you've had an introduction to Python, write down 3 goals you have for yourself and your learning during this Achievement. You can reflect on the following questions if it helps you. What do you want to learn about Python? What do you want to get out of this Achievement? Where or what do you see yourself working on after you complete this Achievement?
  - I want to get a basic grasp of the aspects of Python that are unique (notably those differences from JS, React, and Angular)
  - I want to dive deeply into more complex scripting logic. I think the understandings that come from learning scripting will make stronger with grasping programmatic logic as a whole, and will be applicable to my skills in all/most aspects of coding.
  - I want to learn and compare Python to other coding I have done with a focus on layering causal action trees, such that may be common in the coding logic in video games. I think i have not got enough practice in this kind of programming logic, and I believe it will make me a stronger developer overall.