Exercise 1.1

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?

Front end development works on the side of applications that users see (layouts, graphic designs, etc.). Front end takes the information that back end development provides and formats it in a useful and visually appealing way. On the other hand, back end development works on the side of applications that users don't see. Back end handles all of the necessary things for an application to function correctly, most of the time the user being unware of it. Things like database structure and management, security, and following HTTPS protocols are all examples of things would fall under the responsibility of back end development. As important as it is, back end development tends to only get noticed when something goes wrong.

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?

I would list the similarities and differences between the two languages, shown below:

Similarities:

1. Versatility: Both Python and JavaScript are versatile languages that can be used for a wide range of applications, including web development, data science, machine learning, and more.

2. Ease of Learning: Both languages are considered relatively easy for beginners to pick up. Python, with its clear syntax and readability, is often praised for its simplicity. JavaScript, being the language of the web, is also approachable.

3. Community Support: Both languages have large and active communities, providing extensive resources, libraries, and frameworks. This ensures good community support and a wealth of third-party packages.

Differences:

1. Use Cases:

- Python: Often used for backend development, data analysis, artificial intelligence, and scientific computing. It's known for its readability and clean syntax.

- JavaScript: Primarily used for front-end web development but also employed on the server side (Node.js). It's essential for building interactive and dynamic web applications.

2. Execution Environment:

- Python: Generally executed on the server side using frameworks like Django or Flask.

- JavaScript: Executed in web browsers on the client side and on the server side using Node.js.

3. Learning Curve:

- Python: Often seen as more beginner-friendly due to its clean and readable syntax.

- JavaScript: Can be perceived as having a steeper learning curve, especially for beginners dealing with asynchronous programming concepts.

By considering these factors, my team could make an informed decision based on the specific requirements and goals of the project. If the project heavily involves backend development, data science, or machine learning, Python would be the more suitable choice.

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?

Goal 1: Become more familiar with the Python framework Django

Goal 2: Learn how to do all of the things with Python that I've learned previously in Javascript

Goal 3: Use the features unique to Python to create a web app that stands apart from the Javascript apps I've created in the past