JD Learning Journal

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

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?***

   Frontend, also known as client-side, is what the user sees. When developing the frontend of an application you'd be focusing on any functions that a user will interact with. This includes the basic webstack of HTML, CSS, and JS. Backend, on the other hand, can also be referred to as server-side development. In this role developers are focused on data handling and integrating the many pieces of an application.

   In a backend role one would be handling operations pertaining to database interactions and API creation/management.

2. ***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?***

   Both JavaScript and Python are very popular languages. As such they both offer a wide range of online communities and are constantly evolving to provide better features and modules.

   For this project, I'd propose Python over JavaScript for a couple of reasons. Python is user friendly and comes with a minimal learning curve. While JavaScript has many modules that can be installed, Python comes with a wide range of operations that are offered out of the box. Additionally, Python was built to be readable, which helps to make the language simple to pick up. This is opposed to JavaScript, which has a far steeper learning curve and requires a bit more skill when picking up on existing code bases.

3. ***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?***

    1. I want to learn how to take an idea and put it into practice with Python. There are often ideas for automation that I have and am currently unable to cleanly and concisely turn those ideas into a script.
    2. The main thing I want to get out of this Achievement is how I can use Python to automate tasks that would otherwise be manual and time consuming. This can be applied both to pet projects and in my professional projects.
    3. I see myself implementing my new Python skills in my current position as a Security Engineer. With these skills I'd be able to cut down on the time to complete tasks significantly, and improve current processes as well.