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 web development deals with the client-side of an application (the interface that the users see and interact with when using an application). The tech stack of frontend developers would most likely include languages such as HTML, CSS and JavaScript.
   Backend development deals with the server-side of an application, which is basically everything that goes on behind the scenes to make an application work. The tech stack would include backend language such as Java, Python, Ruby, Perl, and JavaScript (Noje.js).
   If I was hired to work on the backend programming for a web application I would be working on APIs, dealing and interacting with databases, file servers, or others.

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?**
   (Hint: refer to the Exercise section "The Benefits of Developing with Python")

   Similarities with JavaScript:
   - High-level scripting languages, in which commands are executed line by line via a standardized syntax.
   - Uses easily understandable keywords to make commands and perform tasks.
   - Dynamic typing, an approach that allows variables to assume any kind of value without producing errors.

   Differences:
   - Python is a scripting language used for developing both desktop and web applications; JavaScript is a client-side scripting language.
   - In Python, when a function is called with the wrong parameters, an exception is raised; JavaScript doesn't care about the functions being called with the wrong parameters.
   - Python has a comprehensive standard library; JavaScript has a limited set of utility objects.

Python is easy to learn and understand (one of its most striking features is readability which helps keep code error resistant), has a simple built-in package management (supports the installation of a number of open-source and proprietary packages). Python frameworks come with pre-installed common web operations such as URL routing, form handling and validation, template engines, database connections, web security, and session handling. This helps developers get essential operations up and running with relative ease. Python has strong community support.

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?**

    a. Gain more solid programming and coding concepts.
    b. Learn Python fundamentals.
    c. Get a comprehensive understanding of the benefits of developing with Python.