WSGI and Jinja2

In this lesson, we will learn about WSGI and Jinja, and what their purpose is in Flask.

WE'LL COVER THE FOLLOWING

Introduction

You might have heard comments such as *"Flask is 100% WSGI compliant"* or, *"Flask uses Jinja as a template language."* But *what exactly does this mean?* What are **WSGI** and **Jinja2**? Let's learn what these terms mean and their significance concerning Flask!

Web Server Gateway Interface (WSGI)

The **Web Server Gateway Interface**, or more commonly known as **WSGI**, is a standard that describes the specifications concerning the *communication between a web server and a client application*. The detail of these specifications is present in PEP333. Here are some benefits of **WSGI**:

- **Flexibility** with the components of the application.
- **Interoperability** within different Python frameworks.
- **Scalability** of the application with an increase in users.
- **Efficiency** in terms of speed of development.

We do not need to concern ourselves with the internal details of WSGI or how it is implemented because this aspect of the application is handled by the `Flask` module.

The Jinja template language

Jinja is a template language used in Python. But, you might ask, *what exactly is a template language?*

First, let's recall what we covered about templates in the previous lesson. We said that the templates are the front-end, which the user sees. In the case of a website, the templates are the `HTML` pages. A template language is one that we can use inside `HTML` so that the content on the `HTML` page becomes dynamic. This concept might be a little difficult to grasp right now, but no need to worry. We will be discussing Jinja in more detail later on.

In the next lesson, we will be solving a quiz to test the concepts we covered in this chapter.