What is Flask?

In this lesson, a brief introduction of the Flask framework and its history is mentioned.

Introduction

Flask is a **web development framework** developed in Python. It is *easy to learn* and *use*. Flask is **"beginner-friendly"** because it does not have *boilerplate code* or *dependencies,* which can distract from the primary function of an application.

Origins of Flask

Flask originated in 2004 when a developer named **Armin Ronacher** created it as an *April Fool's joke*. However, it quickly gained popularity in the **open-**

source community anyway. Consequently, it developed into a popular open-source project and gained a *massive* following, which it maintains today.

Features of Flask

Some features which make Flask an ideal framework for web application development are:

1. Flask provides a **development server** and a **debugger**.
2. It uses **Jinja2** templates.
3. It is compliant with **WSGI 1.0**.
4. It provides integrated support for **unit testing**.
5. Many extensions are available for Flask, which can be used to enhance its functionalities.

In this course, we will be using a few of these extensions to create a working product.

What is a micro-framework?

Micro-frameworks are the opposite of full-stack frameworks, which also offer additional modules for features such as authentication, database ORM, input validation and sanitization, etc.

Why is Flask called a micro-framework?

Flask is known as a *micro-framework* because it is *lightweight* and *only* provides components that are *essential*. It only provides the necessary components for web development, such as **routing, request handling, sessions,** and so on. For the other functionalities such as *data handling*, the developer can write a custom module or use an *extension*. This approach avoids unnecessary boilerplate code, which is not even being used.

In the next lesson, we will find out how some big-name companies use `Flask` in their products.