ONE PAGE SUMMARY

IT & Software › Backend Development › Flask

Introduction to Flask

Flask: Your Gateway to Simple Web Applications

DEFINITION

Flask is a lightweight web framework for Python that allows developers to build web applications quickly and easily. It provides the tools needed to create dynamic websites with minimal setup.

KEY CONCEPTS

Routing

Routing defines how URLs are mapped to functions in your application, allowing users to access different pages.

Templates

Templates are HTML files that allow you to dynamically generate web pages by inserting data from your application.

Request Handling

Flask can handle incoming requests from users, allowing you to process data and return responses.

Flask Extensions

Extensions add additional functionality to Flask, such as database integration and authentication, making it more powerful.

Development Server

Flask includes a built-in development server that allows you to test your application locally before deploying it.

EXAMPLES

→ Creating a simple 'Hello, World!' app with Flask.

→ Using templates to display user data on a webpage.

→ Setting up a route to handle form submissions.

MEMORY TIPS

★ Remember 'Flask' as a 'flask of water'—it holds everything you need to build apps.

★ Think of 'Routing' as a 'map' guiding users to their destinations.

★ Associate 'Templates' with 'blueprints' for designing your web pages.

COMMON MISTAKES

✗ Forgetting to import Flask and necessary modules.

✗ Neglecting to define routes correctly, leading to 404 errors.

✗ Not using templates for dynamic content, resulting in static pages.

QUICK RECAP

Flask is a flexible web framework for Python that simplifies web application development. Key features include routing, templates, and request handling, making it easy to create dynamic websites.