# Flask Syllabus: Beginner to Advanced

<https://www.geeksforgeeks.org/flask-tutorial/>

* [Flask - Environment](https://www.tutorialspoint.com/flask/flask_environment.htm)
* [Flask - Application](https://www.tutorialspoint.com/flask/flask_application.htm)
* [Flask - Routing](https://www.tutorialspoint.com/flask/flask_routing.htm)
* [Flask - Variable Rules](https://www.tutorialspoint.com/flask/flask_variable_rules.htm)
* [Flask - URL Building](https://www.tutorialspoint.com/flask/flask_url_building.htm)
* [Flask - HTTP Methods](https://www.tutorialspoint.com/flask/flask_http_methods.htm)
* [Flask - Templates](https://www.tutorialspoint.com/flask/flask_templates.htm)
* [Flask - Static Files](https://www.tutorialspoint.com/flask/flask_static_files.htm)
* [Flask - Request Object](https://www.tutorialspoint.com/flask/flask_request_object.htm)
* [Sending Form Data to Template](https://www.tutorialspoint.com/flask/flask_sending_form_data_to_template.htm)
* [Flask - Cookies](https://www.tutorialspoint.com/flask/flask_cookies.htm)
* [Flask - Sessions](https://www.tutorialspoint.com/flask/flask_sessions.htm)
* [Flask - Redirect & Errors](https://www.tutorialspoint.com/flask/flask_redirect_and_errors.htm)
* [Flask - Message Flashing](https://www.tutorialspoint.com/flask/flask_message_flashing.htm)
* [Flask - File Uploading](https://www.tutorialspoint.com/flask/flask_file_uploading.htm)
* [Flask - Extensions](https://www.tutorialspoint.com/flask/flask_extensions.htm)
* [Flask - Mail](https://www.tutorialspoint.com/flask/flask_mail.htm)
* [Flask - WTF](https://www.tutorialspoint.com/flask/flask_wtf.htm)
* [Flask - SQLite](https://www.tutorialspoint.com/flask/flask_sqlite.htm)
* [Flask - SQLAlchemy](https://www.tutorialspoint.com/flask/flask_sqlalchemy.htm)
* [Flask - Sijax](https://www.tutorialspoint.com/flask/flask_sijax.htm)
* [Flask - Deployment](https://www.tutorialspoint.com/flask/flask_deployment.htm)
* [Flask - FastCGI](https://www.tutorialspoint.com/flask/flask_fastcgi.htm)

## ****Module 1: Introduction to Flask****

### ****Sub-Module 1.1: Understanding Web Frameworks****

* What is a Web Framework?
* Why use Flask over other frameworks?
* Difference between Flask and Django

### ****Sub-Module 1.2: Setting up Flask****

* Installing Flask
* Setting up a Virtual Environment
* First Flask Application (Hello World)
* Running Flask App with flask run
* Debugging Mode in Flask

## ****Module 2: Flask Basics****

### ****Sub-Module 2.1: Routing and URL Building****

* Understanding Routes
* Route Parameters and URL Variables
* Using url\_for() Function
* Static Files Handling

### ****Sub-Module 2.2: Templates and Jinja2****

* Introduction to Jinja2 Templating Engine
* Template Variables and Filters
* Template Inheritance
* Including and Extending Templates
* Handling Static Files (CSS, JS, Images)

### ****Sub-Module 2.3: Forms and User Input****

* Handling Form Data with request.form
* GET vs POST Methods
* WTForms for Form Handling
* Form Validation Techniques

## ****Module 3: Database Integration in Flask****

### ****Sub-Module 3.1: SQL with Flask****

* Introduction to Databases in Flask
* Connecting Flask with SQLite/MySQL/PostgreSQL
* Using SQLAlchemy ORM
* CRUD Operations in Flask with SQLAlchemy
* Using Flask-Migrate for Database Migrations

### ****Sub-Module 3.2: NoSQL with Flask****

* Introduction to NoSQL Databases
* Connecting Flask with MongoDB using Flask-PyMongo
* CRUD Operations with MongoDB

## ****Module 4: Flask Advanced Features****

### ****Sub-Module 4.1: Flask Authentication & Security****

* User Authentication with Flask-Login
* Password Hashing with Flask-Bcrypt
* Flask Sessions & Cookies
* JWT Authentication with Flask-JWT-Extended
* Protecting Routes with Authentication

### ****Sub-Module 4.2: RESTful API Development with Flask****

* Introduction to REST APIs
* Creating REST APIs with Flask & Flask-RESTful
* Handling Requests and Responses (JSON format)
* Error Handling in APIs
* API Versioning
* Securing APIs with JWT

### ****Sub-Module 4.3: Background Tasks and Celery****

* Introduction to Celery for Task Queues
* Using Celery with Flask
* Running Background Jobs with Celery

## ****Module 5: Deployment & Testing in Flask****

### ****Sub-Module 5.1: Testing in Flask****

* Writing Unit Tests with pytest
* Mocking API Responses
* Using Postman for API Testing
* Flask Testing Best Practices

### ****Sub-Module 5.2: Deploying Flask Applications****

* Deploying Flask on Heroku
* Deploying Flask on AWS EC2
* Deploying Flask with Docker and Kubernetes
* CI/CD Pipelines for Flask Applications

## ****Module 6: Real-World Projects****

**Flask Blog Application**

* + User Authentication & Authorization
  + CRUD Operations for Posts
  + Admin Panel for Managing Posts

**E-commerce API with Flask**

* + User Authentication & JWT
  + Product Listing & Filtering
  + Cart & Order Management

**Chat Application with Flask & WebSockets**

* + Real-time Chat using Flask-SocketIO
  + User Authentication & Group Chats

**Flask & Machine Learning Integration**

* + Deploying a Machine Learning Model with Flask
  + Creating an API for ML Model Predictions

## ****Module 7: Interview Questions & Preparation****

### ****Sub-Module 7.1: Flask Theory Questions****

* What is Flask and how does it work?
* Explain Flask request lifecycle.
* What is Jinja2 and how does Flask use it?
* How to handle static files in Flask?
* How does Flask handle sessions?
* What is Flask Blueprint and why is it used?

### ****Sub-Module 7.2: Flask Coding Questions****

* Implement a basic Flask API with CRUD operations.
* Write a Flask app with authentication using Flask-Login.
* Create a Flask API that returns data in JSON format.
* Write a Flask script to connect and query a MySQL database.

### ****Sub-Module 7.3: Real-World Scenario-Based Questions****

* How would you scale a Flask application?
* How would you handle API rate limiting in Flask?
* Explain how you would deploy a Flask application using Docker.
* How do you implement role-based access control in Flask?

### ****Conclusion****

This syllabus provides a step-by-step guide to learning Flask from beginner to advanced level. It includes practical exercises, real-world projects, and interview questions to make learners industry-ready.