# Flask Quiz

### Multiple Choice (2)

1. Flask is often described as a “microframework” because:  
 a) It is smaller in file size than Django  
 b) It comes with only essential components by default  
 c) It cannot handle databases  
 d) It is designed only for mobile applications

2. Which of the following is a \*disadvantage\* of using Flask?  
 a) Lightweight and flexible  
 b) Easy to learn for beginners  
 c) Lacks built-in tools for large, complex applications  
 d) Strong community support

### Short Answer (5)

3. What is Flask primarily used for in web development?

4. Which programming language is Flask built on?

5. What command is used to install Flask using pip?

6. In Flask, what does the `@app.route("/")` decorator do?

7. Why is Flask often chosen for building APIs over full-stack web apps?

### Extended Response (3)

8. Explain the typical workflow of a Flask application, from receiving a request to sending a response.

9. Compare Flask with Django in terms of design philosophy, use cases, and scalability.

10. Outline the pros and cons of using Flask for a new project. Under what circumstances would you recommend using Flask, and when might you recommend against it?

## Answer Key

### Multiple Choice

1. b) It comes with only essential components by default

2. c) Lacks built-in tools for large, complex applications

### Short Answer

3. Flask is primarily used to build web applications and APIs.

4. Python.

5. pip install flask

6. It maps the URL `/` to the associated Python function, defining the route.

7. Flask is lightweight, flexible, and doesn’t enforce structure, making it ideal for quick, custom API development.

### Extended Response

8. Workflow: A client sends a request → Flask routes the request based on URL rules → The associated view function runs → Logic is executed (e.g., querying a DB) → Flask builds a response (HTML, JSON, etc.) → Response is sent back to the client.

9. Flask is minimalistic, offering flexibility and simplicity; Django is a full-featured “batteries-included” framework. Flask is best for small apps, APIs, or projects needing custom structure. Django suits large, complex, enterprise-level apps with built-in tools.

10. Pros: Lightweight, flexible, easy to learn, good for APIs and prototypes.  
 Cons: Lacks built-in features for large-scale apps (ORM, auth, admin).  
 Recommend Flask for small to medium apps, APIs, microservices, and learning.  
 Recommend against Flask when building large, feature-rich applications needing many built-in components (Django or other frameworks may be better).