



Top 20 Python Full Stack Interview Questions and Answers

Introduction

Python full-stack development typically entails utilizing Python as your main programming language and having experience with a variety of frameworks, including Django, Flask, Pyramid, Tornado, and others. With just Python, you can become a full-stack developer. Python applies to both front-end and back-end web development. Here are the **Python full-stack interview questions and answers** that are carefully prepared at SLA. Explore our [Python full-stack course syllabus](#) to get started in this promising career.

List of Python Full Stack Interview Questions for Freshers

1. How does front-end development relate to the Document Object Model (DOM)?

2. Explain the difference between inline, internal, and external CSS styles.
3. What is a responsive web design, and how can a web application implement it?
4. In web development, what is cross-origin resource sharing (CORS), and how may it be handled?
5. Explain middleware concerning web frameworks such as Flask and Django.
6. How do web applications' sessions get managed?
7. What is Object-Relational Mapping, or ORM, and how does it help with Python database interactions?
8. What advantages does SQL have over NoSQL?
9. In what ways do databases employ optimization techniques?
10. What is a microservice's life cycle?

Python Full Stack Interview Questions and Answers for Freshers

1.How does front-end development relate to the Document Object Model (DOM)?

The Document Object Model (DOM) is the programming interface for web documents. Programs can alter the document's structure, design, and content by using it as a representation of the page. To allow computer languages to interact with the page, the document is represented as nodes and objects in the Document Object Model (DOM).

2.Explain the difference between inline, internal, and external CSS styles.

External CSS is used for a more structured and scalable approach to styling, allowing for reusability and maintainability, whereas internal CSS is used for different elements within the same **HTML** document. Inline CSS is used for quick and particular styling.

3.What is a responsive web design, and how can a web application implement it?

Responsive web design, or RWD, is the process of creating websites that adapt to the device of the user. Any device that views a website should be able to see it with optimal clarity and usability.

4.In web development, what is cross-origin resource sharing (CORS) and how may it be handled?

The same-origin policy is expanded upon by cross-origin resource sharing or CORS. It is required for approved resource exchanges with outside parties. For instance, CORS is required if you need to retrieve data from authorized or public external APIs.

5.Explain middleware concerning web frameworks such as Flask and Django.

A component that resides between the web server and the view function in Django allows you to handle requests and responses. This component is referred to as middleware. In Django, middleware is essential for managing security and authentication issues.

6.How do web applications' sessions get managed?

The technique of securely managing several requests made by a single user or organization to a web-based application or service is known as session management. A set of HTTP requests and transactions that are initiated by the same user is called a session. HTTP is used by browsers and websites to communicate.

**Take your Knowledge
Test Report**

Check your Score



7.What is Object-Relational Mapping, or ORM, and how does it help with Python database interactions?

Data may be smoothly translated between an object-oriented programming language (OOP) like Python or Java and a relational database management system (RDBMS) thanks to a programming method called object-relational mapping, or ORM. ORM systems provide the necessary connection between the two dissimilar ideas.

8.What advantages does SQL have over NoSQL?

As **SQL databases** scale vertically, higher-end hardware is needed to support growing demands. NoSQL databases are more appropriate for web-scale applications because of their horizontal scalability, which enables them to manage higher traffic or data quantities by distributing data over several servers.

9.In what ways do databases employ optimization techniques?

Organizations can improve system performance, lower operating costs, and provide a better user experience by deliberately applying techniques, including caching mechanisms, indexing, query optimization, and efficient database design.

10.What is a microservice's life cycle?

Microservice development follows a life cycle procedure that is similar to the well-known Software Development Life Cycle (SDLC): design, create, deploy, maintain, and manage.

Reshape your career with our [Python full-stack tutorial for beginners](#).

List of Python Full Stack Interview Questions for Experienced

1. Write down the usages of HTML5 elements, such as <header>, <nav>, <section>, and <article>.

2. How does the Model-View-Controller (MVC) design pattern fit into frameworks like Flask and Django?
3. How can the data transfer security of a web application be guaranteed? Describe the functions of SSL/TLS, HTTPS, and encryption.
4. In collaborative development and deployment, discuss the value of version control systems like Git.
5. In web development, what are microservices, and how are they different from monolithic architectures?
6. How are authorization and authentication handled?
7. Explain the idea of DevOps and how it fits into the continuous integration and deployment (CI/CD) phase of the full-stack development process.
8. What distinguishes service-based architecture from microservices?
9. Describe the distinction between an application server and a web server. How do they collaborate when developing websites?
10. What does a RESTful API perform, and how is it different from GraphQL or SOAP?

Python Full Stack Technical Interview Questions and Answers for Experienced

1. Write down the usages of HTML5 elements, such as `<header>`, `<nav>`, `<section>`, and `<article>`.

HTML tags like `<header>`, `<nav>`, `<section>`, `<article>`, etc. that give meaning or structure to a page's content are known as semantic elements in HTML5. They facilitate accurate styling and page interpretation by assisting **web developers** and browsers in comprehending the kind of material contained within these elements.

When semantic elements are used properly, a web page's accessibility and **search engine optimization (SEO)** are enhanced in addition to its structure and meaning. It improves the usability and discoverability of the page by assisting screen readers and search engines in comprehending the context and content of the page.

`<header>` vs. `<div id="header">`: Use `<header>` For a document or section's main header. It denotes the beginning of a section or the entire page.

`<nav>` vs. `<div id="nav">`: `<nav>` is used for navigation menus. It facilitates search engine and assistive technology recognition of your page's navigation features.

From a conceptual standpoint, the `<section>` and `<article>` components are interchangeable. Consider the following while determining which of them to select:

- An article should be reusable or independently distributable.
- A section is a content grouping based on a theme.

2.How does the Model-View-Controller (MVC) design pattern fit into frameworks like Flask and Django?

The MVC (Model-View-Controller) design pattern, which divides an application into three interdependent elements (Models, Views, and Controllers) for a better web experience and more organized code development projects, is the foundation of Django's architecture.

3.How can the data transfer security of a web application be guaranteed? Describe the functions of SSL/TLS, HTTPS, and encryption.

The client connects to the server over a secure URL (HTTPS). The client receives its public key and certificate from the server. To make sure the certificate is authentic, the client confirms this with a Trusted Root Certification Authority. The strongest encryption that each server and client can support is agreed upon.

4.In collaborative development and deployment, discuss the value of version control systems like Git.

By streamlining development procedures, version control solutions enable quicker feature delivery and iteration. The time from development to deployment can be greatly shortened by using effective branching and merging features, which let

developers work on different project components simultaneously and without interruption.

5. In web development, what are microservices, and how are they different from monolithic architectures?

A microservices architecture consists of many smaller, independently deployable services, whereas a monolithic program is constructed as a single, cohesive unit. Enroll in our [Python full-stack developer course](#) and begin your desired IT career.

6. How are authorization and authentication handled?

There are various phases involved in implementing authentication in microservices. Select an authentication method first, such as JSON Web Tokens (JWT), OpenID Connect, or OAuth 2.0. Next, use an identity provider like Active Directory, LDAP, or a custom user store to centralize user identity management.

**Take your Knowledge
Test Report**

Check your Score



7. Explain the idea of DevOps and how it fits into the continuous integration and deployment (CI/CD) phase of the full-stack development process.

Automation is given top priority in the **DevOps** culture to expedite the development process. Pipelines for continuous deployment (CD) and continuous integration (CI) are essential to this strategy. Testing, building, and deployment are just a few of the software development life cycle stages that these pipelines automate.

8. What distinguishes service-based architecture from microservices?

Each microservice is a considerably smaller software component that focuses on a specific task, whereas each SOA service is a whole business capability. By

addressing SOA's drawbacks, microservices improve software's compatibility with contemporary cloud-based enterprise contexts.

9. Describe the distinction between an application server and a web server. How do they collaborate when developing websites?

Static stuff such as files, HTML pages, pictures, and videos is sent by web servers. Application servers provide dynamic content, such as tailored information, real-time updates, and customer service. doesn't often employ multithreading.

10. What does a RESTful API perform, and how is it different from GraphQL or SOAP?

An architectural idea for application communication is a REST API. Conversely, GraphQL is an API query language, a specification, and a collection of tools. GraphQL uses HTTP to communicate with a single endpoint. Furthermore, the development of REST has shifted its focus towards creating new APIs.

Conclusion

To ace a full-stack interview in Python, one needs to have a well-rounded skill set in the basics of the Python language, experience with frameworks such as Django or FastAPI, and a strong foundation in frontend development.

Preparation is the key that unlocks natural talent and leads to a job offer. If you are ready to take your coding skills to the next level and create a portfolio that will make you stand out, then join our [Software Training Institute in Chennai](#). The journey to your tech career begins now. Enroll today!