Guru99 Provides FREE ONLINE TUTORIAL on Various courses like

Java | MIS | MongoDB | BigData | Cassandra | Web Services

SQLite | JSP | Informatica | Accounting | SAP Training | Python
----
Excel | ASP Net | HBase | Testing | Selenium | CCNA | NodeJS

TensorFlow | Data Warehouse | R Programming | Live Projects | DevOps
----

Top 10 Front End Developer Interview Questions & Answers

1 Who is Front End Developer? What he does?

In a website, front-end is the part that users accesses while interacting with the website including images, buttons, colours, animations, forms, typography etc.

While the frontend developer is a programmer that codes the front end of a website and ensures that a visibility of site remains same throughout different web browsers.

2) What are the skills required

The skills required for the Front End Developer includes

- HTML
- CSS
- JavaScript
- JQuery

Additional Skills
Some additional skills that might be helpful will be

- Knowledge of cross browser testing
- Knowledge of CMS like Wordpress, Joomla or Drupal
- Knowledge of PHP and OOP's (object oriented programming)
- Knowledge of SEO, and tools like Flash and Dreamweaver

3) Explain what is Three.js & its important features?

Three.js is an open source JavaScript 3D library that enables you to make and display animated, interactive 3D computer graphics on any compatible web browser without having a dependency on proprietary plug-ins.

Key features of Three.js include

- Renderers
- Scenes
- Cameras
- Lights
- Animations
- Materials
- Shaders
- Objects
- Geometry
- Loaders
- Export/Import
- Debugging
- Support

4) Explain what is the difference between WebGL and three.js?

WebGL

· WebGL allows you to control the GPU in more direct way

Three.js

· Three.js is built on top of WebGL and allows you to take care of lot of things like what objects to draw each frame

· It is more an "immediate mode"	· It is more a "retained mode"
· It does not have additional support for text, for shaders built, for picking, etc.	· It does have an additional support for text, for picking, for object hierarchy, etc.

5) Explain what is CoffeeScript?

CoffeeScript is a small programming language that compiles into JavaScript. It helps to write JavaScript code better by providing you with a more consistent syntax and avoiding the irregular nature of JavaScript language

The basic rule for Coffee Script

- Whitespace matters: There are no curly braces in CoffeeScript
- No parentheses: Functions that take arguments do not require parentheses

5) What are the benefits of Coffee Script over JavaScript?

- CoffeeScript allows you to express your program with a lot less code than JavaScript
- It has a lot of lightweight add-ons like Ruby string Interpolation and Python style list comprehension
- Makes everyday tasks easier to perform with CoffeScript rather than JavaScript

6) How variables differ in CoffeeScript than JavaScript?

For variables in JavaScript, you have to add semi-colon at the end of it to execute while in CoffeeScript there is no need to add Semi-colon at the end of the statement. Unlike, JavaScript, CoffeeScript adds up semi-colon with ease.

7) What are the basic rules to remember for Coffee Script?

The basic rule for Coffee Script

- Whitespace matters: There are no curly braces in CoffeeScript
- No parentheses: Functions that take arguments do not require parentheses

8) Explain functions in CoffeeScript?

Functions in CoffeeScript is an (Optional) list of parameters followed by an arrow and then the function body.

For example, log = (message) à console.log message

9) In CoffeeScript how clone-function is useful ?

Clone function is useful in creating a complete new object in Coffee Script by

- Copying all attributes from the source object to the new object
- Repeating the steps of copying attributes from the source object for all sub-objects by calling the clone-function
- Creating a new object as the source object

10) Explain what is a Thread-Local object in Python Flask?

Flask uses thread local objects internally so that user don't have to pass objects around from function to function within a request in order to stay threadsafe. This approach is useful, but it requires a valid request context for dependency injection or when attempting to reuse code which uses a value pegged to the request.