**Full Stack Development Questions**

1. **What does "Full Stack" mean in the context of web development?**
   * **Answer**: "Full Stack" refers to a comprehensive approach in web development where a developer works on both the front end (client side) and back end (server side) of a web application, handling all aspects of development.
2. **How does Full Stack development differ from traditional web development?**
   * **Answer**: Full Stack development involves working across all layers of an application, including both client and server-side technologies, whereas traditional web development may focus on either the front end or the back end separately.
3. **What does the MEAN Stack consist of?**
   * **Answer**: The MEAN Stack includes MongoDB (database), Express.js (server framework), Angular (front-end framework), and Node.js (runtime environment).
4. **Explain the Client-Server architecture model in web development.**
   * **Answer**: The Client-Server architecture involves a client making requests to a server, which processes the request and returns the appropriate response. The client typically handles the user interface, while the server manages data and application logic.
5. **What is the LAMP stack, and what does it consist of?**
   * **Answer**: The LAMP stack is a set of open-source software used for web development. It consists of Linux (operating system), Apache (web server), MySQL (database), and PHP (server-side scripting language).
6. **Explain some common front-end frameworks and libraries used in web development.**
   * **Answer**: Common front-end frameworks and libraries include React (for building user interfaces), Angular (for single-page applications), and Vue.js (for reactive interfaces).
7. **Construct a feature comparison chart between the MEAN and MERN stacks.**
   * **Answer**: The MEAN stack uses Angular for front-end development, while the MERN stack uses React. Both stacks include MongoDB, Express.js, and Node.js. The choice between MEAN and MERN often depends on the developer's preference for Angular or React.
8. **Identify the key responsibilities of a Full Stack developer and list the essential technologies required.**
   * **Answer**: Full Stack developers handle both front-end and back-end development, including database management, server configuration, and user interface design. Essential technologies include HTML, CSS, JavaScript (React or Angular), Node.js, and databases (MongoDB or MySQL).
9. **Model the Client-Server architecture and label the roles of clients and servers.**
   * **Answer**: In the Client-Server architecture, the client requests resources or services from the server. The server processes these requests and returns the appropriate response, such as web pages or data.
10. **Experiment with different Client-Server architecture configurations and solve common challenges.**
    * **Answer**: Experiment with configurations like RESTful APIs and WebSocket connections. Challenges include managing request load, ensuring security, and handling server downtime.
11. **Analyze the role of a Full Stack developer in modern web development.**
    * **Answer**: Full Stack developers provide end-to-end solutions, working across the entire technology stack to build and maintain web applications.
12. **Categorize the various technologies used in Full Stack development into front-end and back-end categories.**
    * **Answer**: Front-end technologies include HTML, CSS, JavaScript, and frameworks like React or Angular. Back-end technologies include Node.js, Express.js, databases like MongoDB or MySQL, and server-side languages like PHP or Python.
13. **Discover the key advantages of Full Stack development over specialization in either front-end or back-end development.**
    * **Answer**: Full Stack development offers a holistic view of the application, leading to better integration and coordination between front-end and back-end. It can improve project efficiency and reduce the need for communication between specialized developers.
14. **Distinguish between the components of the MEAN Stack and classify them according to their roles in web development.**
    * **Answer**: MongoDB is the database, Express.js is the server framework, Angular is the front-end framework, and Node.js is the runtime environment.
15. **Examine the themes underlying the choice of MongoDB as the database component in the MEAN Stack.**
    * **Answer**: MongoDB is chosen for its flexibility with JSON-like documents, scalability, and ease of integration with Node.js.
16. **Examine the functions of Full Stack development in enabling end-to-end control over the development process.**
    * **Answer**: Full Stack development allows developers to manage both the user interface and server-side logic, offering complete control over the entire development process from conception to deployment.
17. **Contrast the motives behind choosing Full Stack development versus traditional web development for a given project.**
    * **Answer**: Full Stack development is often chosen for projects requiring seamless integration across various layers of the application, while traditional web development may be used when focusing on specific parts of the stack.
18. **Analyze the relationship between Full Stack development and overall project efficiency and agility.**
    * **Answer**: Full Stack development can enhance project efficiency by reducing handoffs between different developers and providing a unified approach to development, which can increase agility in responding to changes.

**HTML and CSS Questions**

1. **What is the purpose of HTML in web development, and why is it considered the foundation of web pages?**
   * **Answer**: HTML (Hypertext Markup Language) is used to structure content on the web. It provides the basic building blocks for web pages, defining elements like headings, paragraphs, links, and images.
2. **List the different heading levels available in HTML and explain when to use each.**
   * **Answer**: HTML provides six heading levels: <h1> through <h6>. <h1> is used for the main title, <h2> for subsections, and so on. Headings help organize content hierarchically.
3. **Show how to apply inline styles and external stylesheets to HTML elements.**
   * **Answer**:
     + **Inline Styles**: <p style="color: blue;">Hello World</p>
     + **External Stylesheet**: <link rel="stylesheet" href="styles.css">
4. **What are the advantages and disadvantages of inline styles versus external stylesheets?**
   * **Answer**: Inline styles are quick for small changes but can clutter HTML and are harder to maintain. External stylesheets are cleaner, promote reusability, and are easier to manage across multiple pages.
5. **Classify the <p> tag as a block-level element and explain its role in defining paragraphs in HTML documents.**
   * **Answer**: The <p> tag is a block-level element used to define paragraphs, creating vertical spacing before and after the text.
6. **Contrast the usage of line breaks (<br>) with paragraph tags for structuring text content.**
   * **Answer**: <br> inserts a line break without additional spacing, suitable for single-line breaks. <p> creates a new paragraph with spacing, suitable for separating blocks of text.
7. **Compare the rendering of paragraphs in HTML documents with and without CSS styles applied.**
   * **Answer**: Without CSS, paragraphs render with default browser styling. With CSS, you can control margins, padding, font sizes, and other styles for more precise control over presentation.
8. **Explain how to create hyperlinks using the <a> tag and the href attribute.**
   * **Answer**: Hyperlinks are created with the <a> tag and href attribute. For example: <a href="https://www.example.com">Visit Example</a>.
9. **Relate the role of anchor tags in navigation and content linking within web pages.**
   * **Answer**: Anchor tags are used for linking to other pages, sections within a page, or external resources, facilitating navigation and content linking.
10. **Demonstrate the usage of target attribute values (\_blank, \_self, \_parent, \_top) for controlling link behavior.**
    * **Answer**:
      + \_blank: Opens the link in a new tab.
      + \_self: Opens the link in the same frame (default).
      + \_parent: Opens the link in the parent frame.
      + \_top: Opens the link in the full body of the window.
11. **Classify the <ul>, <ol>, and <dl> tags as unordered lists, ordered lists, and description lists, respectively.**
    * **Answer**: <ul> is used for unordered lists (bulleted), <ol> for ordered lists (numbered), and <dl> for description lists (term-definition pairs).
12. **Build a college timetable using the HTML table tag.**
    * **Answer**:

<table>

<tr>

<th>Time</th>

<th>Monday</th>

<th>Tuesday</th>

</tr>

<tr>

<td>9:00 AM</td>

<td>Math</td>

<td>Physics</td>

</tr>

<tr>

<td>10:00 AM</td>

<td>Chemistry</td>

<td>Biology</td>

</tr>

</table>

1. **Build an account statement for a customer using HTML table tag.**
   * **Answer**:

<table>

<tr>

<th>Transaction Date</th>

<th>Customer Acc No</th>

<th>Amount</th>

<th>Credit/Debit</th>

</tr>

<tr>

<td>2024-09-01</td>

<td>123456</td>

<td>$500</td>

<td>Credit</td>

</tr>

<tr>

<td>2024-09-05</td>

<td>123456</td>

<td>$200</td>

<td>Debit</td>

</tr>

</table>

1. **Build a student result table using HTML table tag.**
   * **Answer**:

<table>

<tr>

<th>Subjects</th>

<th>English</th>

<th>Physics</th>

<th>Chemistry</th>

<th>Maths</th>

<th>Python</th>

<th>Total</th>

</tr>

<tr>

<td>Student 1</td>

<td>85</td>

<td>90</td>

<td>78</td>

<td>88</td>

<td>92</td>

<td>433</td>

</tr>

</table>

1. **Discover various methods of specifying colors in HTML, including color names, hexadecimal codes, and RGB values.**
   * **Answer**:
     + **Color Names**: <p style="color: red;">Text</p>
     + **Hexadecimal Codes**: <p style="color: #ff0000;">Text</p>
     + **RGB Values**: <p style="color: rgb(255, 0, 0);">Text</p>
2. **Analyze different text formatting tags available in HTML with use cases.**
   * **Answer**:
     + <sub>: Displays text as subscript (e.g., H<sub>2</sub>O).
     + <sup>: Displays text as superscript (e.g., x<sup>2</sup>).
     + <pre>: Displays preformatted text, preserving whitespace.
     + <code>: Used for code snippets.
3. **Functionally examine the role of the class attribute in applying CSS styles to multiple HTML elements.**
   * **Answer**: The class attribute allows multiple elements to share the same CSS styles. Example: <p class="highlight">Text</p> with CSS .highlight { color: yellow; }.
4. **Compare block-level, inline, and inline-block tags. Name a few tags of each type and provide scenarios for their use.**
   * **Answer**:
     + **Block-level**: <div>, <p>. Use for larger sections or elements.
     + **Inline**: <span>, <a>. Use for small segments or inline elements.
     + **Inline-block**: <img>, <button>. Use for elements that need to maintain block characteristics but align inline.
5. **Distinguish between the 3 ways to use CSS: inline, internal, and external. Compare each way and explain their use.**
   * **Answer**:
     + **Inline CSS**: Applied directly to HTML elements using the style attribute. Use for quick, specific styling.
     + **Internal CSS**: Defined within the <style> tag in the HTML <head>. Use for styles specific to a single document.
     + **External CSS**: Defined in a separate CSS file linked with the <link> tag. Use for site-wide styling and better organization.
6. **Contrast different pseudo-classes available in CSS. Apply pseudo-classes to an HTML unordered list.**
   * **Answer**:
     + **:hover**: Applies styles when an element is hovered over.
     + **:focus**: Applies styles when an element is focused.
     + **:nth-child(n)**: Applies styles to specific children based on their position.

<ul>

<li class="item">Item 1</li>

<li class="item">Item 2</li>

<li class="item">Item 3</li>

</ul>

<style>

.item:hover { color: red; }

.item:nth-child(2) { font-weight: bold; }

</style>