|  |
| --- |
|  |
| Basic :- **Question 1 : What is the purpose of the <DOCTYPE> declaration in HTML?** Ans: The <!DOCTYPE> declaration defines the document type and version of HTML being used. It helps browsers to interpret the document correctly by specifying the rules and specifications to follow.  **Question 2: Explain the difference between <div> and <span> elements.**  Ans: <div> is a block-level element used to group and structure content into larger sections, typically for layout purposes. <span> is an inline element used to apply styles or markup to a specific part of text or inline content.  **Question 3: What is the purpose of the <head> and <body> elements in HTML?**  Ans: The <head> element contains meta-information about the document, such as title, character set, and linked stylesheets. The <body> element holds the visible content of the page.  **Question 4: Explain the difference between em and rem units in CSS. When would you use one over the other?**  Ans: em units are relative to their parent element, while rem units are relative to the root element (<html>). rem units are generally preferred for global sizing (e.g., for font sizes) to avoid compounding effects, while em units can be useful for more local adjustments.  **Question 5: What is the CSS "cascading" in Cascading Style Sheets, and how does it work?**  Ans: Cascading refers to the process of combining multiple CSS rules to determine the final styles applied to an element. It follows a specific order of importance: user agent styles, author styles, and user styles, with more specific selectors taking precedence.  **Question 6: Explain the differences between flex-grow, flex-shrink, and flex-basis in Flexbox.**  Ans: flex-grow determines how much an item should grow relative to other items when there's extra space. flex-shrink specifies how much an item should shrink when there's insufficient space. flex-basis defines the initial size of an item before distributing extra or insufficient space.  **Question 7: Explain the purpose of the grid-template-areas property in CSS Grid.**  Ans: grid-template-areas allows you to define a grid layout by specifying the placement of grid items using named areas. It simplifies the grid layout definition and is especially useful for responsive designs. Medium:- **Question 1: What is the significance of the <meta> tag in HTML?**  Ans : The <meta> tag provides metadata about the HTML document, such as character set, author, and description. It's also used for SEO and specifying viewport settings for mobile devices.  **Question 2: What is the significance of the HTML defer and async attributes when including JavaScript files?**  The defer attribute ensures that the script is executed in order after the HTML is parsed, while the async attribute allows the script to be downloaded asynchronously and executed as soon as it's available, potentially out of order.  **Question 3: What is the significance of the HTML defer and async attributes when including JavaScript files?**  Ans : The defer attribute ensures that the script is executed in order after the HTML is parsed, while the async attribute allows the script to be downloaded asynchronously and executed as soon as it's available, potentially out of order.  **Question 4: What is the purpose of the HTML <details> and <summary> elements, and how are they used?**  Ans : <details> is used to create a disclosure widget, like a collapsible section. <summary> is used as the visible label for the disclosure widget. It provides a way to hide/show additional content.  **Question 5: Explain the concept of CSS specificity, and how is it calculated?**  Ans : CSS specificity determines which CSS rule takes precedence when multiple conflicting styles are applied. It is calculated based on the combination of selector types (ID, class, element) and the number of instances. For example, an ID selector has higher specificity than a class selector.  **Question :6 What is the "box-sizing" property in CSS, and why is it useful?**  Ans : The box-sizing property determines how an element's total width and height are calculated. Setting it to border-box includes padding and border within the specified width, making layout calculations more predictable.  **Question 7: Explain the difference between position: relative, position: absolute, and position: fixed in CSS, including their behavior within a positioned parent element.**  Ans : position: relative; positions an element relative to its normal position. It respects its parent's positioning. position: absolute; positions an element relative to its nearest positioned ancestor or the viewport if no ancestor is positioned.position: fixed; positions an element relative to the viewport.  **Question 8: What are CSS custom properties (variables), and how can they be defined and used in CSS?**  Ans : CSS custom properties (variables) are user-defined variables that store values for reuse in CSS. They are defined using the -- prefix and can be used within CSS rules.  **Question 9: How can you create a vertically centered flex container with unknown height using Flexbox?**  Ans : To center a flex container vertically with unknown height, you can use align-items: center; on the container and give it a height of 100% within a parent with display: flex; and flex-direction: column;. Hard : - **Question 1: What is semantic HTML?**  Ans : Semantic HTML refers to using HTML elements that convey meaning about the structure of the web page's content. Examples include <header>, <nav>, <article>, and <footer>. Semantic HTML improves accessibility and helps search engines understand the content better.  **Question 2: Explain the purpose of the <iframe> element in HTML.**  Ans : <iframe> is used to embed another web page or external content within the current page.  **Question 3: Explain the difference between HTML5 Web Workers and Service Workers.**  Ans : Web Workers are used for running scripts in the background to improve performance, while Service Workers are a type of Web Worker used for caching and enabling features like offline browsing and push notifications.  **Question 4: What is the purpose of the <figure> and <figcaption> elements in HTML5, and how are they used?**  Ans : <figure> is used to group media content with a caption, and <figcaption> is used to provide a description or title for the content within the <figure>. They improve accessibility and semantics.  **Question 5: Explain the concept of HTML microdata. How is it used for search engine optimization (SEO)?**  Ans : HTML microdata is used to embed machine-readable data within HTML content, making it easier for search engines to understand the meaning of the content. It's typically used for SEO by providing structured data about items, events, and more.  **Question 6: What is the purpose of the HTML <template> element, and how does it differ from traditional HTML markup?**  Ans : The <template> element allows you to define client-side templates that can be cloned and inserted into the document using JavaScript. It differs from traditional markup by not rendering on the page until explicitly activated, improving performance and maintainability.  **Question 7: What is HTML5's contenteditable attribute, and how can it be used?**  Ans : The contenteditable attribute makes an element's content editable by the user. It's commonly used for creating rich text editors and interactive web applications.  **Question 8: Explain the purpose of the HTML <time> element and how it's used to represent dates and times.**  Ans : <time> is used to represent dates and times in a machine-readable format. It includes the datetime attribute for the machine-readable value and optional human-readable content. For example: <time datetime="2023-09-30T10:00:00Z">September 30, 2023</time>  **Question 9: What is the CSS "float" property, and how does it affect layout?**  Ans : The float property is used to make an element float to the left or right within its containing element. It's often used for creating multi-column layouts or positioning elements side by side. It can lead to layout issues if not managed correctly.  **Question 10: Explain the concept of CSS pseudo-elements ::before and ::after. How can they be used?**  Ans : ::before and ::after are pseudo-elements used to insert content before and after the content of an element, respectively. They are often used for decorative elements or to generate additional content without altering the HTML markup.  **Question 11: What is the CSS "z-index" property, and how is it used to control stacking order?**  Ans : The z-index property controls the stacking order of elements with position: relative, position: absolute, or position: fixed. Elements with a higher z-index value are stacked on top of elements with lower values. It's essential for managing layering in complex layouts.  **Question 12: Explain the concept of CSS specificity in relation to pseudo-classes and pseudo-elements.**  Ans : Pseudo-classes (e.g., :hover, :first-child) and pseudo-elements (e.g., ::before, ::after) have their specificity values. They are included in the specificity calculation when determining which styles to apply. Pseudo-elements generally have higher specificity than pseudo-classes.  **Question 13: Explain the concept of the "Holy Grail Layout" in the context of Flexbox.**  Ans : The Holy Grail Layout is a common design pattern that includes a header, footer, and three columns with a fluid center column. Flexbox can simplify the implementation of this layout by managing the space distribution and column order.  **Question 14: Explain the difference between implicit and explicit grids in CSS Grid.**  Ans : Explicit grids are defined explicitly using properties like grid-template-rows and grid-template-columns. Implicit grids are created automatically when grid items are placed outside the explicitly defined grid, and they can be controlled with properties like grid-auto-rows and grid-auto-columns.  **Question 15: How can you create a responsive, equal-height grid of items with CSS Grid when the content varies in height?**  Ans : You can use the align-content: stretch; property on the grid container to make grid items equal in height by stretching them to fill the container vertically.  **Question 16: How can you create a masonry-style layout with CSS Grid?**  Ans : To create a masonry layout where items are positioned based on available space rather than a fixed grid, you can use CSS Grid with auto-fill or auto-fit for the column tracks and grid-auto-rows: minmax(auto, 1fr); for variable row heights. |