1. **Difference between <section> and <article>**
    - **<section>**: Defines a generic section within a document, often grouping related content under a common theme, but not necessarily standalone.

    - **<article>**: Represents a self-contained composition, like a blog post, that could be distributed independently from the rest of the content.

2. **Difference between <header> and <nav>**:
    - **<header>**: A container for introductory content or navigation links related to the current section or page, often containing a logo, heading, or other initial information.
    - **<nav>**: Specifically for a set of navigational links, guiding users to other sections or pages.

3. **Element to display extra information on click**:
    - **<details>** and <summary>: <details> creates a collapsible element, and <summary> provides a visible heading that, when clicked, shows or hides additional content within <details>.

4. **Difference between <meter> and <progress>**:
    - **<meter>**: Displays a scalar measurement within a known range, like disk usage or temperature.
    - **<progress>**: Shows the progress of a task, such as a download or upload percentage.

5. **Div color determination**:
    - The color would be **purple** due to the inline style (style="color: purple;"), which has the highest specificity, overriding both #myDiv (even with !important) and .myClass.

6. **Flex display for a desktop menu bar (desktop-first approach)**:
    - Use display: flex; flex-direction: row; for a horizontal layout.

7. **Responsive layout adjustments**:
    - **Mobile (up to 600px)**: Convert to a single column with centered items.
    - **Tablet (601px - 900px)**: Arrange in two rows with items wrapped using flex-wrap and calculated widths.

8. **Creating a gradient background and centered white content section**:
    - Use CSS with a linear gradient (background: linear-gradient(...)) for section Linear Gradient Background, set the white Content Section for white color and centered positioning, and direction Row for a flexible layout that defaults to column.

9. **CSS selectors**:
    - **Attribute Selector**: Targets elements with specific attributes, e.g., [type="text"].
    - **Negation Pseudo-Class**: Selects elements not matching a selector, e.g., :not(.className).
    - **Nth Pseudo-Class**: Targets elements based on their position, e.g., :nth-child(2n) selects every second child.

HTML5 Questions

1. **What is the purpose of the <!DOCTYPE html> declaration in HTML5?**
   - **Answer**: It tells the browser that the document is HTML5, helping it display the page correctly.
2. **Explain the difference between <div> and <span> elements in HTML.**
   - **Answer**: <div> is a block-level element used for grouping larger sections, while <span> is an inline element used for small parts within text.
3. **What are semantic elements, and why are they important?**
   - **Answer**: Semantic elements, like <header>, <section>, and <article>, describe their purpose clearly. This makes code easier to read for both developers and browsers, improving accessibility and SEO.
4. **How do <header>, <footer>, and <nav> elements differ?**
   - **Answer**: <header> is for the top section, often containing a logo or title; <footer> is for the bottom, usually with copyright info; and <nav> is for navigation links.
5. **When would you use the <section> element?**
   - **Answer**: When you need to group related content under a common theme, like a chapter or section in a book.
6. **What is the purpose of the <article> element?**
   - **Answer**: It's for standalone content that could be reused or shared independently, like a blog post.
7. **What does the <mark> element do?**
   - **Answer**: <mark> highlights text, like with a yellow marker, making it stand out.
8. **How does the <figure> and <figcaption> pair work?**
   - **Answer**: <figure> is for content like images or code blocks, and <figcaption> is for adding a caption below the figure.
9. **Explain the use of the <details> and <summary> elements.**
   - **Answer**: <details> creates a collapsible section that can be shown or hidden, and <summary> provides the clickable label to open or close it.
10. **How do <progress> and <meter> differ?**
    - **Answer**: <progress> shows task progress (like download completion), while <meter> represents a measurement within a range (like a temperature gauge).

Explain the difference between <section> and <article> in HTML5.

- **Answer**: <section> is for grouping related content under a common theme, while <article> is for standalone content that could be reused, like a blog post.

What element should you use to show extra information on a click?

- **Answer**: Use <details> with <summary> to create a collapsible section.

CSS Basics and Core CSS Questions

11. **What is CSS used for in web development?**
    - **Answer**: CSS styles HTML elements, controlling colors, fonts, layout, and spacing.
12. **How do you link a CSS file to an HTML document?**
    - **Answer**: Using <link rel="stylesheet" href="styles.css"> inside the <head> section.
13. **What are attribute selectors in CSS?**
    - **Answer**: Selectors that style elements with certain attributes, like [type="text"] for input fields of type "text".
14. **Explain how pseudo-classes work in CSS.**
    - **Answer**: Pseudo-classes style elements based on states, like :hover for when the user hovers over an element.
15. **What is a pseudo-element, and can you give an example?**
    - **Answer**: It styles parts of an element, like ::before to add content before the element.
16. **What is CSS specificity, and why is it important?**
    - **Answer**: It determines which CSS rules apply if there are multiple conflicting styles. Specificity is calculated based on the selector type.
17. **How does the !important declaration affect CSS rules?**
    - **Answer**: It overrides all other rules, making the specific style apply even if other rules have higher specificity.

Advanced CSS Concepts

18. **What is a CSS transition, and when would you use it?**
    - **Answer**: A transition makes CSS changes (like color) happen gradually over time, adding smooth effects.
19. **How is an animation different from a transition?**
    - **Answer**: Animations can have multiple steps (keyframes) and run automatically, while transitions need an event (like hover) to start.
20. **What is a transform in CSS, and give an example?**

- **Answer**: Transforms change an element's size, rotation, or position, like transform: scale(1.5); to make it 1.5 times bigger.

21. **Explain the purpose of responsive web design (RWD).**
    - **Answer**: RWD ensures websites look good on all devices, adapting to different screen sizes.
22. **How do media queries support responsive design?**
    - **Answer**: Media queries apply specific CSS based on device characteristics, like screen width, to change layouts for desktops, tablets, or phones.
23. **What is the difference between Flexbox and CSS Grid?**
    - **Answer**: Flexbox is for one-dimensional layouts (row or column), while CSS Grid is two-dimensional, controlling both rows and columns.
24. **Describe the structure of a Flexbox layout.**
    - **Answer**: Flexbox has a flex container (parent) and flex items (children). Items align in a row or column, and space is distributed evenly.
25. **What is the purpose of flex-wrap in Flexbox?**
    - **Answer**: flex-wrap allows items to wrap onto multiple lines if they can't fit in a single row or column.
26. **How does the justify-content property work in Flexbox?**
    - **Answer**: It aligns items along the main axis, like centering or spacing items across a row or column.
27. **What is a CSS Grid container, and what does it do?**
    - **Answer**: It's the parent with display: grid; that defines the grid structure for placing items in rows and columns.
28. **How do you create rows and columns in CSS Grid?**
    - **Answer**: Using grid-template-rows and grid-template-columns properties, you can define the layout structure.
29. **What is the purpose of a grid area in CSS Grid?**
    - **Answer**: Grid areas let you place elements in specific sections of the grid layout, making complex designs easier to manage.
30. **Explain the calc() function in CSS.**
    - **Answer**: calc() performs calculations within CSS, like width: calc(100% - 20px); to adjust size dynamically.

CSS Selectors and Specificity Questions

31. **What is the difference between class and ID selectors in CSS?**
    - **Answer**: IDs (#id) are unique for a single element, while classes (.class) can be used on multiple elements.
32. **How does the nth-child pseudo-class work?**
    - **Answer**: It selects elements based on their order, like :nth-child(2) for the second child or :nth-child(odd) for odd-numbered children.
33. **What is the negation pseudo-class in CSS?**
    - **Answer**: :not() excludes elements from a selector, like :not(.class) for elements without that class.
34. **How does the [attribute] selector work?**
    - **Answer**: It targets elements with specific attributes, like [target] for elements with a target attribute.

Practical Questions Based on Code

35. **Given CSS with an inline style, an ID selector, and a class selector, which style applies?**
    - **Answer**: The inline style has the highest specificity and will apply.
36. **How would you create a menu that displays items in a row using Flexbox?**
    - **Answer**: Set the container to display: flex; flex-direction: row;.
37. **How would you make a flex container responsive to switch from row to column on smaller screens?**
    - **Answer**: Use a media query to change flex-direction: row; to flex-direction: column; for smaller screens.
38. **What CSS would you use to add a gradient background with specific colors?**
    - **Answer**: background: linear-gradient(#E44444, #CD487B);.
39. **How would you center content within a div using CSS?**
    - **Answer**: Use display: flex; align-items: center; justify-content: center; on the parent container.

Write CSS to create a linear gradient from blue to green.

- **Answer**: background: linear-gradient(blue, green);

Modify the gradient to run diagonally from top-left to bottom-right.

- **Answer**: background: linear-gradient(45deg, blue, green);

How would you display elements in a two-row layout using CSS Grid?

- **Answer**: Use display: grid; grid-template-rows: repeat(2, 1fr); to define two rows in the container.

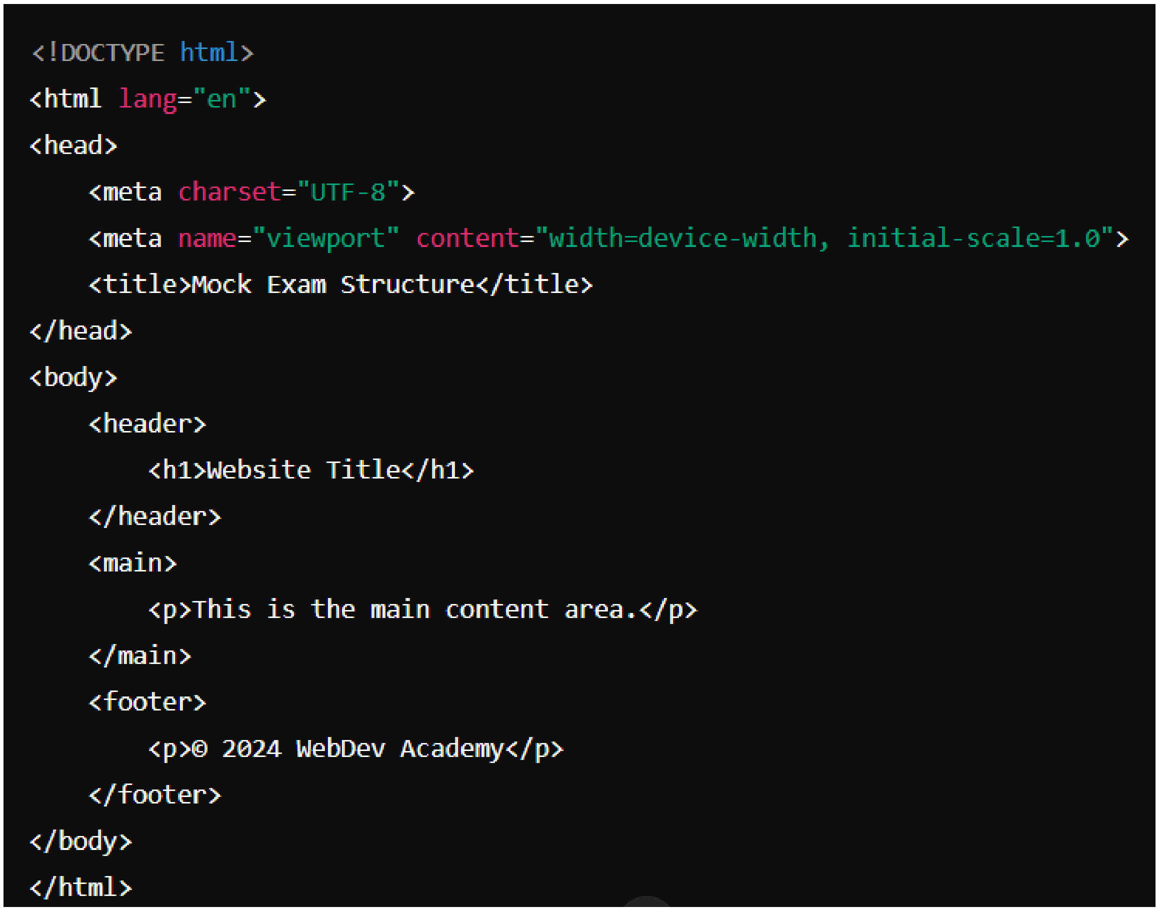
Mock Exam Questions

Question 1: HTML5 Semantic Elements (10 Marks)

(a) What are semantic elements in HTML5? Explain why they are important for web development. Provide at least 3 examples of semantic elements. (5 marks)

- **Answer**: Semantic elements in HTML5 clearly define their purpose in the code, making it easier for browsers, developers, and screen readers to understand the content. This improves accessibility, SEO, and the overall readability of the code.
  - **Examples**: <header>, <main>, <footer>

(b) Write a simple HTML5 structure using at least 3 semantic elements to define the header, main content, and footer of a web page. (5 marks)



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Mock Exam Structure</title>
</head>
<body>
    <header>
        <h1>Website Title</h1>
    </header>
    <main>
        <p>This is the main content area.</p>
    </main>
    <footer>
        <p>© 2024 WebDev Academy</p>
    </footer>
</body>
</html>
```

Question 2: Responsive Design (10 Marks)

(a) Explain the concept of responsive design. Why is it essential in modern web development? (4 marks)

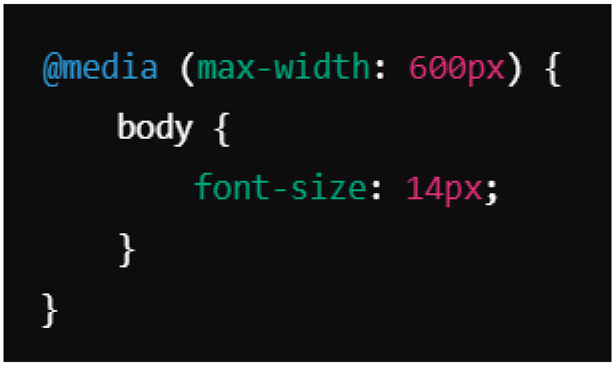
- **Answer**: Responsive design ensures that a website's layout and elements adjust to fit different screen sizes and devices. It's essential because it provides a consistent user experience on desktops, tablets, and mobile devices, improving accessibility and usability.

(b) Name and describe two elements/attributes that are important for building responsive websites. (4 marks)

- **Answer**:
  - **Viewport meta tag**: Controls the viewport's scaling and dimensions, making the layout responsive to different screen sizes.
  - **Flexible grid layouts (e.g., Flexbox/Grid)**: These allow layouts to adapt to screen size by adjusting the position and alignment of elements.

(c) What is a media query in CSS, and how is it used for responsive design? Provide a basic example. (2 marks)

- **Answer**: A media query applies styles based on device characteristics, like screen width. It's used to make elements responsive.



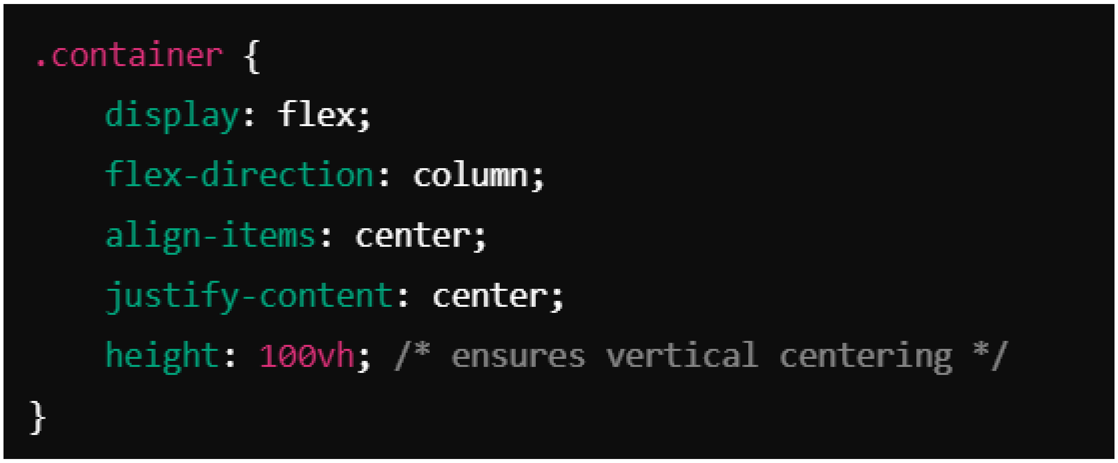
```
@media (max-width: 600px) {
    body {
        font-size: 14px;
    }
}
```

Question 3: Flexbox Directions (10 Marks)

(a) In Flexbox, what are the flex-direction, wrap, order, and basis properties? Explain how each affects the layout of the flex items. (5 marks)

- **Answer**:
  - **flex-direction**: Controls the direction of items in the container (row, row-reverse, column, column-reverse).
  - **wrap**: Allows items to wrap onto multiple lines within the flex container.
  - **order**: Sets the order of flex items, overriding their natural placement.
  - **basis**: Sets the initial size of a flex item before space is distributed based on flex-grow or flex-shrink.

(b) Given a flex container, write CSS to display its child elements in a column layout and align them in the center both horizontally and vertically. (5 marks)



```css
.container {
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    height: 100vh; /* ensures vertical centering */
}
```

Question 4: Specificity Rules (10 Marks)

(a) Explain how CSS specificity works. Provide an example of how specificity affects the application of styles in a web page. (5 marks)

- **Answer**: CSS specificity determines which styles apply when multiple rules target the same element. Specificity is calculated based on selectors (ID > Class > Element). For example, if #header and .nav target the same element, the ID selector (#header) has higher specificity and will apply.

(b) Calculate the specificity of the following selectors and explain which one has higher specificity:

- i) .header .nav-item
- ii) #main-content .article h1
- iii) div p
- **Answer**:
    - i) .header .nav-item = 0, 0, 2, 0
    - ii) #main-content .article h1 = 0, 1, 1, 1
    - iii) div p = 0, 0, 0, 2
    - **Explanation**: #main-content .article h1 has the highest specificity due to the ID selector (#main-content).

Question 5: Simple Gradients (10 Marks)

(a) What is a gradient in CSS? Explain the difference between a linear gradient and a radial gradient. (4 marks)

- **Answer**: A gradient is a smooth transition between colors.
    - **Linear gradient**: Colors blend along a straight line (horizontal, vertical, diagonal).
    - **Radial gradient**: Colors blend outward from a center point, creating a circular or elliptical effect.

(b) Write a CSS rule to create a background that transitions from blue to green using a linear gradient. (3 marks)

```
background: linear-gradient(blue, green);
```

(c) How can you control the direction of a linear gradient? Modify your answer in (b) to create a diagonal gradient. (3 marks)

```
background: linear-gradient(45deg, blue, green);
```