Structure & Semantics

1. **What are semantic elements in HTML?**
   Semantic elements clearly describe their purpose and content, making the code easier to understand and accessible. Examples include:

```
<header>This is a header</header>

<article>This is an article</article>

<footer>This is a footer</footer>
```

2. **Why are semantic elements important?**
   Key reasons:
   - Improves readability: Easier for developers to understand the structure.
   - Enhances accessibility: Assistive technologies can interpret the content better.
   - Optimizes SEO: Search engines can better rank meaningful content.
     Example:

```
<section>
  <h2>Blog Posts</h2>
  <article>
    <h3>Post Title</h3>
    <p>Post content here...</p>
  </article>
</section>
```

3. **Difference between and <section>:**
   - : Generic container with no semantic meaning.
   - <section>: Represents a distinct section or thematic grouping of content.
     Example:

```
<div class="wrapper">
  <p>This is a non-semantic div container.</p>
</div>
<section>
  <h2>Section Title</h2>
  <p>This is a semantic section.</p>
```

```
</section>
```

4. **What is the purpose of the <meta> tag in HTML?**
   Provides metadata about the document for browsers and search engines.
   Example:

```
<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta name="description" content="This is a sample webpage.">
```

5. **Difference between id and class attributes in HTML:**
   - id: Unique, applied to one element.
   - class: Reusable, applied to multiple elements.
     Example:

```
<div id="unique-element">This has a unique ID.</div>

<div class="common-class">This shares a class.</div>

<div class="common-class">This also shares the same class.</div>
```

6. **Differences between inline, block, and inline-block elements:**
   - **Inline:** Elements do not start on a new line (e.g., , <a>).
   - **Block:** Elements start on a new line and take the full width (e.g., , <p>).
   - **Inline-block:** Combines properties of both; doesn't start a new line but allows width/height control.
     Example:

```
<span>This is inline.</span>

<div>This is block.</div>

<div style="display: inline-block; width: 100px;">This is inline-block.</div>
```

7. **How does the doctype declaration affect rendering in browsers?**
   It specifies the HTML version to the browser. A correct <!DOCTYPE> ensures standards mode rendering instead of quirks mode, improving compatibility.

8. **How does the browser handle invalid HTML?**
   Browsers attempt to correct and render invalid HTML using their error-recovery mechanisms. The layout may differ depending on the browser.

9. **Explain the role of <link> and <script> tags:**
   - <link>: Links external resources like stylesheets.
     Example: <link rel="stylesheet" href="styles.css">

- <script>: Embeds or links JavaScript files.
  Example: <script src="script.js"></script>

10. **How can you optimize an HTML page for SEO?**
    - Use semantic tags (<header>, <article>, etc.).
    - Add meta tags (<meta name="description">).
    - Use proper heading hierarchy (<h1> to <h6>).
    - Optimize images with alt attributes.
      Example:

    <meta name="description" content="Best recipes blog.">

    <h1>Top Recipes</h1>

    <img src="dish.jpg" alt="Delicious Dish">

Content & Attributes

1. **How does the <meta> tag work, and what are its common uses?**
   It provides metadata, influencing browser behavior and SEO.
   Example:

   <meta name="author" content="John Doe">

2. **New features in HTML5 vs. HTML4:**
   - New semantic tags (<section>, <article>).
   - Native multimedia support (<audio>, <video>).
   - APIs like Geolocation and Web Storage.

3. **Embed multimedia in HTML:**
   Example:

   <audio controls>

     <source src="audio.mp3" type="audio/mpeg">

   </audio>

   <video controls width="640">

     <source src="video.mp4" type="video/mp4">

   </video>

4. **Purpose of data-* attribute:**
   Stores custom data for JavaScript.
   Example:

   User Info

5. **Use of the <canvas> element:**
   Provides a drawing area for graphics.
   Example:

   <canvas id="myCanvas" width="200" height="100"></canvas>

6. **Lazy loading in HTML:**
   Defers loading of images or iframes using the loading attribute.
   Example:

   <img src="large-image.jpg" loading="lazy" alt="Image">

7. **Difference between and tags:**
   - : Bold text, no emphasis.
   - : Bold text with semantic emphasis.
     Example:

     Bold Text

     Important Text

8. **Use of the srcset attribute in <img>:**
   Defines multiple image sources for responsiveness.
   Example:

   <img src="small.jpg" srcset="large.jpg 1024w, medium.jpg 640w" sizes="(max-width: 600px) 100vw, 50vw" alt="Responsive Image">

Advanced Concepts

1. **What are web components (shadow DOM and custom elements)?**
   Web components are reusable encapsulated elements.
   - **Shadow DOM:** Provides scoped styling and DOM.
     Example:

     <template id="my-template">

       <style>p { color: red; }</style>

       Shadow content

     </template>

2. **How to create a progress bar in HTML without JavaScript:**
   Example:

   ```
   <progress value="70" max="100">70%</progress>
   ```

3. **Usage of the <dialog> tag in HTML5:**
   Creates native modals/dialogs.
   Example:

   ```
   <dialog id="myDialog">
     <p>This is a dialog.</p>
     <button onclick="document.getElementById('myDialog').close()">Close</button>
   </dialog>
   <button onclick="document.getElementById('myDialog').showModal()">Open Dialog</button>
   ```