**Web Designing Assignment**

**Term-1**

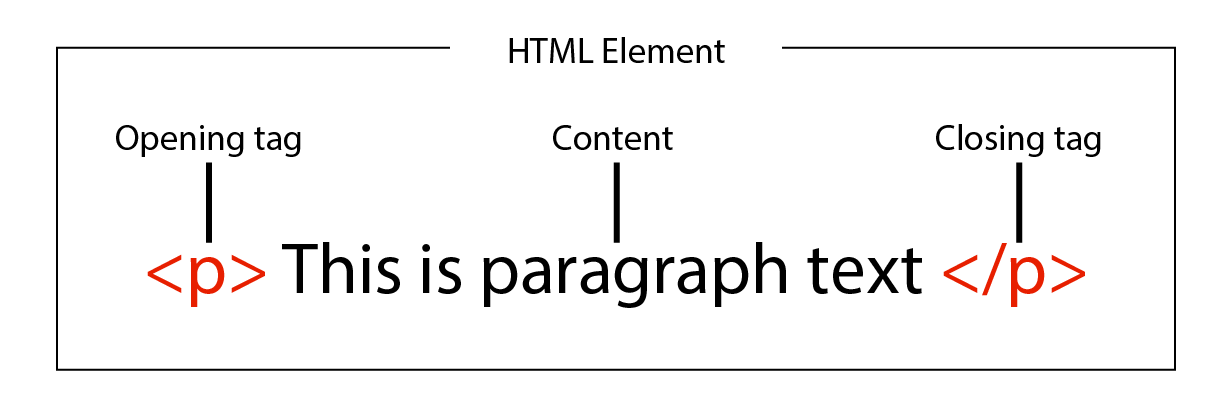
**Module (HTML) -1**

**1. Are the HTML tags and elements the same thing?**

HTML tags and elements are related but not exactly the same thing.

**HTML Tags**: Think of HTML tags as labels or instructions that tell the web browser how to display content. They are like the commands you give to format text or embed images. For example, `<p>` is a tag that tells the browser to start a new paragraph, and `</p>` is a tag that tells it to end the paragraph.

**HTML Elements**: HTML elements are made up of tags and the content they enclose. An HTML element consists of an opening tag, some content, and a closing tag. So, when you have `<p>Hello</p>`, the `<p>` is the opening tag, "Hello" is the content, and `</p>` is the closing tag. Elements are what you see on a webpage, like paragraphs, headings, images, links, and more.

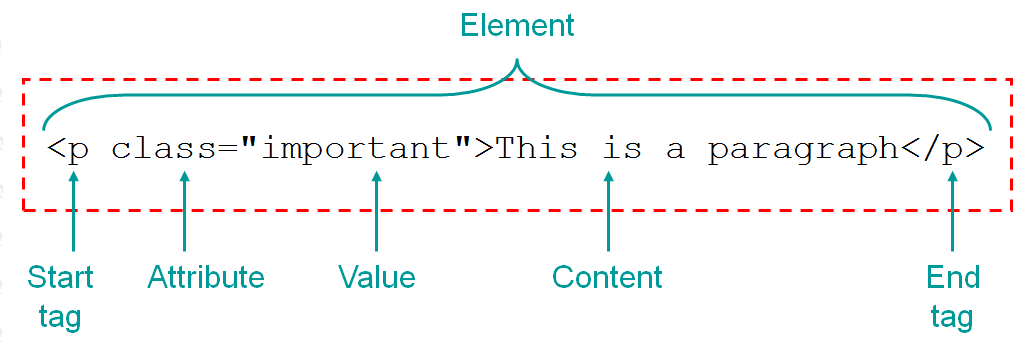
****

**2. What are tags and attributes in HTML?**

In HTML, tags and attributes are important components for creating web pages.

**Tags:** Think of HTML tags as labels or instructions that tell the web browser how to structure and display content. Tags are like commands that you put around your content to define its type or how it should be formatted. For example, the `<h1>` tag is used to create a top-level heading, and the `<p>` tag is used for paragraphs. Tags are enclosed in angle brackets, like `<tagname>`.

**Attributes**: Attributes provide additional information or settings for HTML tags. They are like extra instructions that you can add to a tag to control its behavior. Attributes are always placed inside the opening tag of an element and are typically written as name-value pairs. For example, the `src` attribute in an `<img>` tag specifies the source URL of an image, and the `href` attribute in an `<a>` tag defines the link's destination. Attributes help you customize how a tag works or how it's linked to other elements.

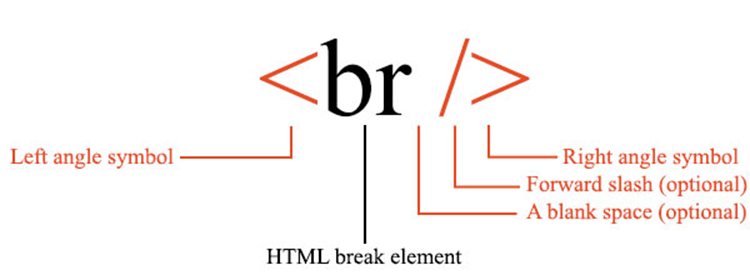


**3. What are void elements in HTML?**

Void elements in HTML are elements that don't have a closing tag. Instead, they stand alone as a single tag. These elements are used to insert something specific into a web page, like an image or a line break, and they don't contain any other content.

For example, the `<img>` tag is a void element used to display images. You don't need to write `<img></img>`, as you would with non-void elements. You simply write `<img>` to insert an image.

In simple terms, void elements in HTML are like one-stop instructions for adding certain things to a webpage, and they don't require a separate closing tag. They're handy for elements that don't have any content of their own and serve a specific purpose, like showing an image or creating a line break.



**4. What are HTML Entities?**

HTML entities are a way to represent special characters and symbols in HTML using a code or name. These special characters might include things like symbols (e.g., ©), foreign language characters (e.g., é), mathematical symbols (e.g., ±), or characters that have special meanings in HTML (e.g., < and &).

Instead of typing these characters directly into your HTML code, you use a special code or name that starts with an ampersand (&) and ends with a semicolon (;). For example, `&copy;` represents the copyright symbol ©, and `&lt;` represents the less-than sign (<).

HTML entities ensure that these characters are displayed correctly in web pages and don't cause confusion with HTML code. They are especially useful when you need to include characters that have special significance in HTML or are not easily accessible from your keyboard.

Here are some commonly used HTML entities:

1. &lt; - Less Than Sign (<):
2. &gt; - Greater Than Sign (>):
3. &amp; - Ampersand (&):
4. &quot; - Double Quotation Mark ("):
5. &apos; or &#39; - Single Quotation Mark ('):
6. &nbsp; - Non-Breaking Space:

**5. What are different types of lists in HTML?**

In HTML, you can create different types of lists to organize and display information on a web page. There are three main types of lists:

1. **Ordered Lists (`<ol>`):** Ordered lists are used when you want to present items in a specific sequence or order. Each item in an ordered list is preceded by a number or another type of marker, such as letters or Roman numerals. For example, a recipe with step-by-step instructions or a list of ranked items would typically use an ordered list. You create an ordered list using the `<ol>` tag.

<ol>

<li>Preheat the oven to 350°F.</li>

<li>Mix the ingredients in a bowl.</li>

<li>Bake for 30 minutes.</li>

</ol>

2. **Unordered Lists (`<ul>`):** Unordered lists are used when the order of items doesn't matter, and you want to display them with bullet points or other markers. Common examples include lists of features, benefits, or simple lists of items. You create an unordered list using the `<ul>` tag.

<ul>

<li>Apples</li>

<li>Bananas</li>

<li>Oranges</li>

</ul>

3. **Definition Lists (`<dl>`):** Definition lists are used when you want to define terms or items and provide a description or definition for each one. They consist of term-definition pairs. You create a definition list using the `<dl>` tag, with `<dt>` for terms (like a word) and `<dd>` for their corresponding definitions.

<dl>

<dt>HTML</dt>

<dd>HyperText Markup Language</dd>

<dt>CSS</dt>

<dd>Cascading Style Sheets</dd>

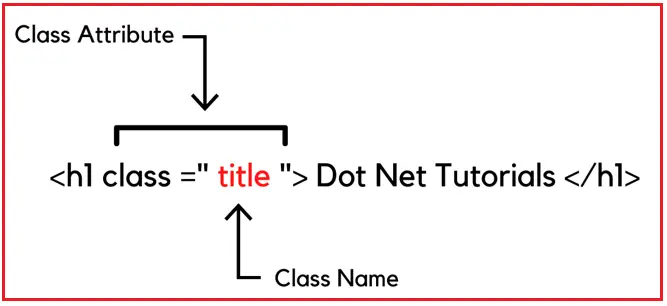
</dl>

**6. What is the ‘class’ attribute in HTML?**

In HTML, the `class` attribute is like a label or tag that you can give to HTML elements to group them together. It doesn't affect how the element looks or behaves by itself, but it's really useful when you want to style or manipulate multiple elements on a webpage in the same way.

For example, you might have several paragraphs on a page, but you want some of them to have a special style, like a different background color. You can assign the same `class` attribute to those paragraphs, and then use CSS (Cascading Style Sheets) to apply the desired styles to all elements with that class.

So, the `class` attribute helps you organize and control the appearance and behavior of elements on your webpage, especially when you want to apply the same rules to multiple elements without repeating the same instructions for each one.



**7. What is the difference between the ‘id’ attribute and the ‘class’ attribute of HTML elements?**

The 'id' attribute and the 'class' attribute in HTML serve similar purposes in that they both help you target and style specific elements on a webpage, but they have key differences:

**1. 'id' Attribute:**

**Uniqueness:** The 'id' attribute is used to uniquely identify a single HTML element on a page. This means that no two elements on the same page should have the same 'id'.

**Styling:** You can use 'id' to apply specific styles or behaviors to a single element. It's typically used when you have a unique or distinct element that needs to be styled or manipulated differently from others.

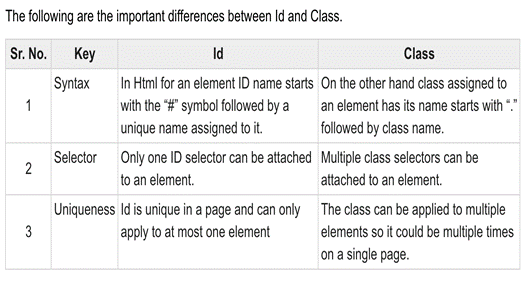
**Example:** If you have a unique header at the top of your page, you might give it an 'id' like `<div id="header">...</div>`, and then apply custom styles or JavaScript functions specifically to this header.

**2. 'class' Attribute:**

- **Multiplicity:** The 'class' attribute can be applied to multiple elements on a page. Multiple elements can share the same 'class'.

- **Styling**: It's used to group elements together. You can apply a 'class' to multiple elements to style or format them in a consistent way. It's handy for styling multiple elements similarly without having to repeat styles for each one.

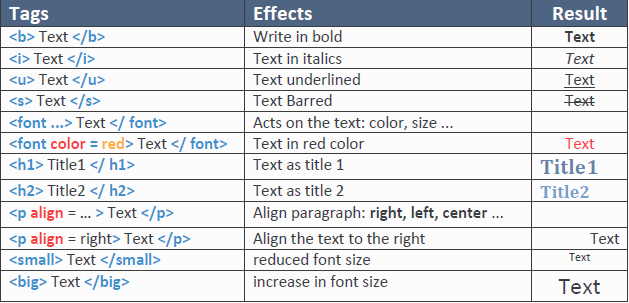
- **Example:** If you want to style all the paragraphs on your page with a specific font, you might use `<p class="special-font">...</p>` for each paragraph, and then apply the font style to all elements with the 'special-font' class.



**8. What are the various formatting tags in HTML?**

Formatting tags in HTML are used to change the appearance or structure of text and elements on a web page. Here are some common formatting tags explained in simple words:

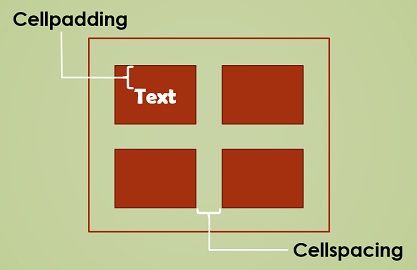
1. **Heading Tags (<h1> to <h6>):** These tags are used to create headings and subheadings. **<h1>** is the largest and most important, while **<h6>** is the smallest. They help organize and give hierarchy to your content.
2. **Paragraph Tag (<p>):** The **<p>** tag is used to create paragraphs of text. It adds space before and after the text, making it easy to separate and format different blocks of content.
3. **Bold Tag (<b> or <strong>):** These tags make text bold. **<b>** is used for stylistic bolding, while **<strong>** is used to indicate strong importance, which is often rendered as bold by browsers.
4. **Italic Tag (<i> or <em>):** These tags make text italic. **<i>** is used for stylistic italicization, while **<em>** is used to emphasize text, often rendered as italic by browsers.
5. **Underline Tag (<u>):** The **<u>** tag is used to underline text. However, it's not commonly used because underlined text is often associated with hyperlinks.
6. **Strike-Through Tag (<s> or <del>):** These tags create a strikethrough effect on text. **<s>** is used for stylistic purposes, while **<del>** is used to indicate deleted or removed text.
7. **Subscript Tag (<sub>):** The **<sub>** tag is used to create subscript text, like in chemical formulas where numbers appear below the text line.
8. Superscript Tag (<sup>): The <sup> tag is used to create superscript text, like in mathematical exponents or footnotes.



**9. How is Cell Padding different from Cell Spacing?**

In HTML tables, both cell padding and cell spacing affect the space around table cells, but they have different purposes:

1. Cell Padding:
   * Purpose: Cell padding controls the space between the content (text or other elements) inside a table cell and the cell's border.
   * Effect: Increasing cell padding adds space between the content and the cell's border, making the content appear more spaced out from the cell edges.
   * Example: If you set cellpadding="10" on a table, it adds 10 pixels of space between the content and the cell border for all cells in the table.
2. Cell Spacing:
   * Purpose: Cell spacing controls the space between adjacent cells in a table.
   * Effect: Increasing cell spacing adds space between the borders of neighboring cells, creating a gap or padding between cells.
   * Example: If you set cellspacing="5" on a table, it adds 5 pixels of space between the borders of adjacent cells, creating a visible gap between cells.

I

**10. How can we club two or more rows or columns into a single row or column in an HTML table?**

To combine two or more rows or columns into a single row or column in an HTML table, you can use the **rowspan** and **colspan** attributes. Here's how it works in theory:

**Rowspan (rowspan):**

* If you want to merge multiple rows into a single row, you typically start by creating the cell in the first row where you want the merging to begin.
* In that cell, you add the **rowspan** attribute, which tells how many rows you want to span or merge together.
* This cell will occupy the space of the specified number of rows below it, effectively merging them into one.

Top of Form

**Example:**

<table border="1">

<tr>

<td rowspan="3">This spans 3 rows</td>

<td>Row 1, Cell 2</td>

</tr>

<tr>

<td>Row 2, Cell 2</td>

</tr>

<tr>

<td>Row 3, Cell 2</td>

</tr>

</table>

In this example, the first cell spans three rows, merging them into a single row.

**Colspan (colspan):**

* If you want to merge multiple columns into a single column, you typically create the cell in the first column where you want the merging to begin.
* In that cell, you add the colspan attribute, specifying how many columns you want to span or merge.
* This cell will occupy the space of the specified number of columns to its right, effectively merging them into one.

**Example:**

<table border="1">

<tr>

<td colspan="3">This spans 3 columns</td>

</tr>

<tr>

<td>Row 2, Cell 1</td>

<td>Row 2, Cell 2</td>

<td>Row 2, Cell 3</td>

</tr>

</table>

In this example, the first cell spans three columns, merging them into a single column.

**11. What is the difference between a block-level element and an inline element?**

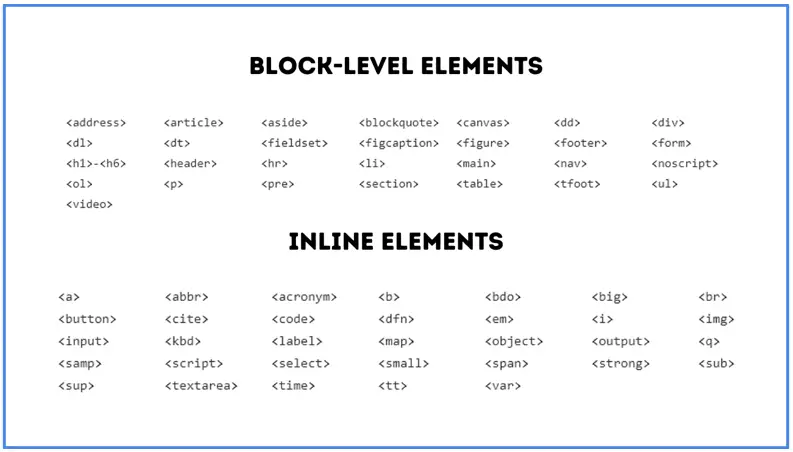
Block-level elements and inline elements are two different types of HTML elements that behave differently in terms of their layout and how they interact with other elements on a webpage. Here's a simple explanation of the key differences:

**Block-Level Elements:**

* Block-level elements create a new "block" or "box" on the webpage. They typically start on a new line and stretch across the full width of their container (like a new paragraph or a heading).
* Block-level elements are used for structuring the layout and organizing content into sections or blocks.
* Examples of block-level elements include **<div>**, **<p>**, **<h1>**, **<ul>**, and **<li>**.

**Inline Elements:**

* Inline elements do not create new blocks; they flow within the content of a block-level element. They appear on the same line as the surrounding text or elements and only take up as much width as necessary.
* Inline elements are often used for styling or adding small bits of content within a block of text, like links or emphasis (italic or bold text).
* Examples of inline elements include **<span>**, **<a>**, **<strong>**, **<em>**, and **<img>**.



**12. How to create a Hyperlink in HTML?**

Creating a hyperlink (a clickable link) in HTML is straightforward. You use the **<a>** tag, which stands for "anchor," and you specify the destination URL within it. Here's how to create a hyperlink in simple words:

1. **Open the <a> Tag:** Start by typing **<a>**.
2. **Specify the Destination URL:** Inside the **<a>** tag, include the **href** attribute and set it equal to the URL you want the link to point to. This URL could be a web address, a file path, or another web page within your website.

Example: To link to a website like Google, you would write **<a href="https://www.google.com">**.

1. **Add Link Text:** After the **href** attribute, type the text you want to display as the clickable link. This is what users will see and click on.

Example: **<a href="https://www.google.com">Go to Google</a>**.

1. **Close the <a> Tag:** End the hyperlink by typing **</a>**.

Syntax:

<a href="https://www.example.com">Visit Example.com</a>

For another html file.

<a href="example.html ">Visit Example.com</a>

**13. What is the use of an iframe tag?**

The **<iframe>** tag in HTML is used to embed another web document or webpage within the current webpage. It allows you to display external content within a designated area on your webpage. Here are some common use cases:

1. **Embedding External Content:** You can use an **<iframe>** to embed content from other websites, such as maps, videos, social media posts, or external web applications, directly within your own webpage.
2. **Creating Inline Frames:** **<iframe>** creates a separate "frame" or "window" within your webpage, isolating the embedded content from the rest of the page. This means that the styles and behaviors of the embedded content won't affect your main webpage.
3. **Displaying PDFs and Documents:** You can embed PDF files or other document types using **<iframe**. This is often used to display documents like product manuals, reports, or application forms.
4. **Interactive Widgets:** **<iframe>** is used to integrate interactive widgets or components (like a calendar, chatbox, or calculator) from other services or platforms into your webpage.

**Syntax:**

<a href="https://www.example.com">Visit Example.com</a>

For another html file.

<a href="example.html ">Visit Example.com</a>

**14. What is the use of a span tag? Explain with example?**

The **<span>** tag in HTML is used to apply inline styling or to group and target specific portions of text or inline elements within a block of text. It doesn't add any visual formatting by itself but can be used in combination with CSS (Cascading Style Sheets) to style and manipulate text or inline elements.

Here's how it works with an example:

Let's say you have a paragraph of text, and you want to make a specific word or phrase stand out by changing its color:

**Example:**

<p>This is a <span style="color: blue;">blue</span> word in the sentence.</p>

In this example:

* **<span>** is used to create a "span" or a small inline container around the word "blue."
* **style="color: blue;"** is an inline style applied to the **<span>** element, specifying that the text within this **<span>** should appear in blue color.
* As a result, only the word "blue" within the paragraph will be displayed in blue, while the rest of the text remains unchanged.

**15. How to insert a picture into a background image of a web page ?**

We can use the background attribute in the body tag to set an image as the background of the webpage. We will need to specify the URL or the location of the image which we want to set to the background attribute of the body tag.

**Example.**

**< style="background-image: url('example.jpg');">.**

<img src="Leaf 2.png" alt="" height="100px" width="100px">

**16. How are active links different from normal links?**

Active links and normal links, in the context of web design and user experience, refer to links that have different appearances or behaviors based on the user's interaction with them. Here's how they differ:

1. **Normal Links:**
   * Normal links are the default appearance of hyperlinks on a webpage.
   * They typically appear as underlined or differently colored text, depending on the website's design.
   * When a user hovers their mouse pointer over a normal link, it may change color or have a subtle animation to indicate interactivity.
   * Clicking on a normal link typically leads the user to a new page or a different location within the same page (e.g., scrolling to an anchor point).
2. **Active Links:**
   * Active links, on the other hand, refer to links that change their appearance or behavior when they are currently being clicked or interacted with.
   * This change is often used to provide visual feedback to the user, confirming that their click or interaction has been registered.
   * For example, when a user clicks on an active link, it might briefly change color or have a distinct styling to show that it's currently being activated.

**17. What are the different tags to separate sections of text?**

**Ans.** There are many tags in html that are used to saprate sections of text.

**Examples:**

Section tag <section>: it makes a saction in web page to saprate its content from other.

**<saction>**

**<h1>Section 1</h1>**

**<p>Content of section 1</p>**

**</saction>**

Paragraph tag <p>:it makes a paragraph of text. It is mostly used tag for making sactions of text in web pages.

**<p>Content of section 1</p>**

* Line break tag <br>: it makes a break in line or creates a gap(whitespace) in a web page. Main purpose of this is to break in line and start from other line.

**<p>Content of <br>section 1</p>**

Content in after <br> will be written

In next line.

These are some tags to make sections in a web page. there are many more tags to do this like <div>,<h1>,<hr> and article etc.

**18. What is SVG?**

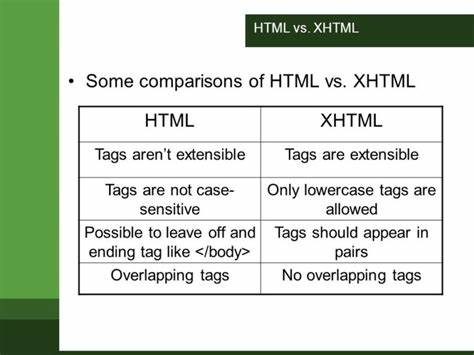
SVG stands for Scalable Vector Graphics. It is a widely used XML-based format for creating two-dimensional vector graphics, which can be both static and dynamic (interactive). SVG is designed to describe graphics in a way that is independent of resolution or display size, making it ideal for displaying images and illustrations on the web.

**19. What is difference between HTML and XHTML?**

HTML (Hypertext Markup Language) and XHTML (Extensible Hypertext Markup Language) are both markup languages used for creating web pages and defining their structure. However, there are some key differences between them:

- **HTML:** HTML has more forgiving syntax rules, which means it can be less strict about closing tags and proper nesting. For example, in HTML, it's common to omit closing tags for certain elements like `<p>` and `<li>`.

- **XHTML:** XHTML has stricter syntax rules and enforces well-formed XML syntax. All elements must be properly nested and closed, and attribute values must be enclosed in quotes. For example, in XHTML, you must write `<p></p>` instead of just `<p>`.



**20. What are logical and physical tags in HTML?**

In HTML, there's a distinction between logical tags and physical tags, and it's important to understand their differences:

1. **Logical Tags:**
   * Logical tags describe the purpose or meaning of the content, rather than its appearance or how it should be styled.
   * They are used to structure and define the content in a semantically meaningful way.
   * Logical tags are typically used for organizing and conveying the structure and hierarchy of the content.
   * Examples of logical tags include **<header>**, **<nav>**, **<section>**, **<article>**, **<aside>**, **<footer>**, **<h1>** to **<h6>**, **<p>**, **<ul>**, **<ol>**, **<li>**, and others.
2. **Physical Tags:**
   * Physical tags describe how the content should be presented or styled on the web page.
   * They are used to control the appearance and layout of the content.
   * Physical tags are often associated with CSS (Cascading Style Sheets) to apply styles, such as colors, fonts, spacing, and positioning.
   * Examples of physical tags include **<b>**, **<i>**, **<u>**, **<font>**, **<span>**, **<br>**, **<hr>**, and others. 