UNIT-II

Coding Standards:

Validation

All HTML pages should be verified against the W3C validator to ensure that the markup is well formed. This in and of itself is not directly indicative of good code, but it helps to weed out problems that are able to be tested via automation. It is no substitute for manual code review. (For other validators, see HTML Validation in the Codex.)

Self-closing Elements

All tags must be properly closed. For tags that can wrap nodes such as text or other elements, termination is a trivial enough task. For tags that are self-closing, the forward slash should have exactly one space preceding it:

```
<br />
```

rather than the compact but incorrect:

```
<br/>
```

The W3C specifies that a single space should precede the self-closing slash (source).

Formatting

All HTML documents must use **two spaces** for indentation and there should be no trailing whitespace. HTML5 syntax must be used and all attributes must use double quotes around attributes.

```
<video autoplay="autoplay" poster="poster_image.jpg">
  <source src="foo.ogg" type="video/ogg">
</video>
```

HTML5 elements should be used where appropriate reserving `<div>` and `<span>` elements for situations where there is no semantic value (such as wrapping elements to provide styling hooks).

Doctype and layout

All documents must be using the HTML5 doctype and the `<html>` element should have a `"lang"` attribute. The `<head>` should also at a minimum include `"viewport"` and `"charset"` meta tags.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Example Site</title>
  </head>
  <body></body>
</html>
```

Forms

Form fields must always include a `<label>` element with a `"for"` attribute matching the `"id"` on the input. This helps accessibility by focusing the input when the label is clicked, it also helps screen readers match labels to their respective inputs.

```
<label for="field-email">email</label>
<input type="email" id="field-email" name="email" value="" />
```

Block Elements:

Every HTML element has a default display value, depending on what type of element it is.

The two most common display values are block and inline.

Block-level Elements

A block-level element always starts on a new line, and the browsers automatically add some space (a margin) before and after the element.

A block-level element always takes up the full width available (stretches out to the left and right as far as it can).

Two commonly used block elements are: <p> and <div>.

The <p> element defines a paragraph in an HTML document.

The <div> element defines a division or a section in an HTML document.

The <p> element is a block-level element.

The <div> element is a block-level element.

Example

```
<p>HelloWorld</p>
<div>Hello World</div>
```

HTML Coding Conventions:

Web developers are often uncertain about the coding technique and syntax to use in HTML as the HTML version changed from HTML to XHTML and finally HTML5. Although, HTML5 is far more flexible but sloppy in terms of code validation, the code itself is easier to understand. Since it supports bad coding, it is always a smart move to write HTML codes close to XHTML so that you do not need to rewrite the entire code in case new browsers or the newer versions of the existing ones stopped supporting bad coding standards.

Here are a few HTML coding conventions you might want to follow to maintain proper coding stardard.

1. Always keep your code clean and well-formed with proper indentation.

2. Always declare the document type as the first line in your document.
3. Use lowercase for element names
4. Close all HTML elements even the empty tags like **<hr />, <br />, <img />**
5. Use lowercase for attribute names
6. Always enclose attribute values with quotes. **(" ")**
7. Always add alt attribute to images.
8. Avoid long code lines
9. Do not add blank lines without a reason.
10. Make the **<title>** elemet as meaningful as possible
11. Always define the language and the character encoding **<meta>** in the document
12. Use lowercase for filenames

HTML and XHTML

XHTML stands for EXtensible HyperText Markup Language. It is almost identical to HTML but is stricter in coding standard. XHTML is supported by all major browsers. It is a markup language where documents must be marked up correctly. It was developed by combining the strengths of HTML and XML.

Bad HTML Coding Example

```
<html>

<head>

  <title>This is bad HTML</title>

<body>

  <h1>Bad HTML

  <p>This is a paragraph

</body>
```

Comments:

HTML comments are used to add notes or explanations in the HTML code that are not displayed by the browser.

- They are useful for documenting the code, making it easier to understand and maintain.
- To add a comment, use the syntax <!– your comment here –>.

```
<!-- This is a comment and will not be displayed on the webpage -->
<p>This is visible text.</p>
```

In this example:

- The text within the <!-- and --> tags will not appear on the webpage.
- These comments can include reminders, warnings, or explanations about the code, which can be useful for anyone reading or editing the HTML document.

HTML Elements:

See UNIT-I

Should Describe Web Page Content Accurately:

An overarching goal in web programming is to use appropriate HTML elements so your web page's content is described accurately. For example, if you have text that forms what would normally be considered a paragraph, then surround the text with a p element, not some other element (like div). Likewise, if you want to display words as a heading, use a heading element (h1-h6), not some other element

Key points about accurate web page content description:

Semantic HTML:

Utilize HTML tags that accurately reflect the meaning of the content, like using <header> for headers, <nav> for navigation, <article> for main content, and <aside> for sidebars.

Descriptive titles:

The title tag should clearly convey the main topic of the page.

Alt text for images:

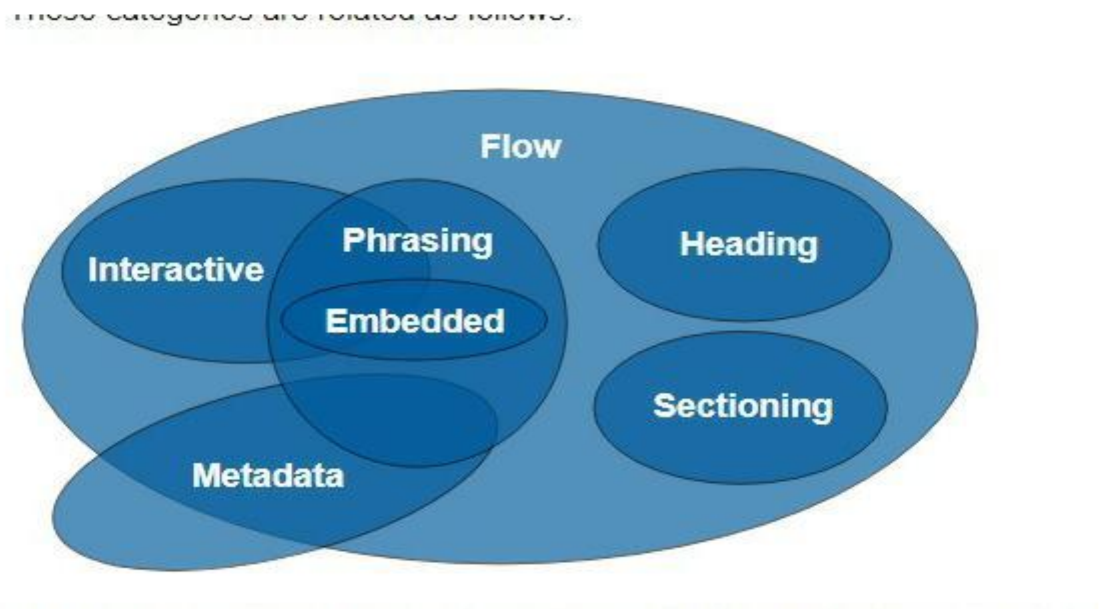
Provide descriptive text for images to assist visually impaired users and improve search engine optimization.

Heading hierarchy:

Use heading tags (H1, H2, H3, etc.) to properly structure the content hierarchy on the page.

Content Model Categories:

HTML content models are the rules that define what content can be included in each HTML element. They determine the types of elements that can be nested inside other elements.



How do content models work?

- Each HTML element has a content model.
- The content of an HTML element must match the requirements of its content model.
- Content models are important for accessibility and SEO. They identify the purpose or structure of content.

Examples of HTML elements:

- **<div>**: Groups other HTML content together.

<main>: Specifies the main content of a document.

<model>: Embeds interactive 3D models into a website.

Other HTML document categories:

Transitional: The most common type of HTML.

Strict: Returns rules to HTML to make it more reliable.

Frameset: Allows developers to create a mosaic of HTML documents and a menu system.

Blockquote Element:

The <blockquote> tag in HTML is used to define a section that is quoted from another source. It is typically displayed as an indented block to set it apart from the surrounding content.

- The <blockquote> tag should be used for longer quotations that require separation from the main text, while the <q> tag is more appropriate for shorter, inline quotations.
- Using the <blockquote> tag enhances the semantic structure of your HTML document, clearly indicating that a section of text is a quotation.
- The <blockquote> tag also supports the Global Attributes and Event Attributes in HTML.

```
<!DOCTYPE html>
<html>
<body>
   <p>Here is a famous quote:</p>
   <blockquote cite="https://www.geeksforgeeks.org/html-blockquote-tag/">
      "The `<blockquote>` tag in HTML is used to define a block of text that
     is a quotation from another source."
   </blockquote>
</body>
</html>
```

Output:

```
Here is a famous quote:
              "The `
                     ` tag in HTML is used to define a block of text that is a quotation from
another source."
```

Whitespace Collapsing:

White space refers to empty or blank values in the code which the browser reads and renders. Html has a special feature of collapsing these white spaces. If you put extra/consecutive white spaces or newlines in the code it will regard it as one white space this is known as collapsing of white spaces. In this article, we will discuss the advantages of collapsing white space in HTML.

Advantages of collapsing white spaces :

- While you are writing the HTML for your web page you want the code to be more **understandable/readable** to the users.
- Collapsing white spaces **decreases the transmission time** between the server and the client because collapsing features remove unnecessary bytes that are occupied by the white spaces.
- By mistake, if you **leave extra** white **space**, the browser will ignore it and display the UI perfectly.

Example :

```
<!DOCTYPE html>
<html>
<head>
    <title>Page Title</title>
</head>
<body>
    <h1>Welcome to GFG.               GeeksforGeeks.</h1>
</body>
</html>
```

Pre Element:

The pre tag in HTML defines a block of preformatted text, preserving spaces, line breaks, and tabs. It displays text in a fixed-width font, which can be styled using CSS. This tag is useful for displaying code, formatted text, and preserving text layout.

```
<!DOCTYPE html>
<html>
<body>
  <pre>
    This    is      preformatted
    text.
    It preserves spaces and
```

```
      line breaks exactly.
   </pre>
</body>
</html>
```

The <pre> tag is a powerful tool for preserving the exact formatting of text within a web page. By using the <pre> tag, along with related tags like <samp>, <var>, <code>, and <kbd>, you can effectively display preformatted text, sample outputs, variables, code snippets, and keyboard inputs in a visually appealing and semantically meaningful way. Customizing the appearance of the <pre> element with CSS further enhances its versatility and integration into your web design.

Phrasing Elements:

Phrase Tag: In HTML, a phrase tag is used to indicate the structural meaning of a block of text. For example, abbr tag indicates that the phrase contains the abbreviation word. Some examples of phrase tags are abbr, strong, mark, . . . etc.

Emphasized Text: The em tag is used to emphasize the text and this tag displays the italic font in a browser. It simply means anything written within **em** tag is shown as emphasized Text.

Syntax:

```
<em> Text Content </em>
```

Example:

```
<!DOCTYPE html>
<html>
<body>

   <h1>GeeksforGeeks</h1>
   <!-- em tag is used in below paragraph-->
   <p><em>GeeksforGeeks</em> emphasized tag</p>
```

```
</body>
</html>
```

Editing Elements:

You can edit HTML elements using HTML editors, such as Firefox, Sublime Text, and Adobe Dreamweaver. You can also use JavaScript to change HTML elements.

Using HTML editors

Firefox

Double-click the text you want to edit, change it, and press Enter to see the changes

Sublime Text

A text editor that's popular for HTML editing

Adobe Dreamweaver

Supports HTML, CSS, and JavaScript, and lets you choose between a text-based or WYSIWYG editor

Using JavaScript

contenteditable attribute: Add the contenteditable="true" attribute to an element to make it editable

innerHTML property: Use the innerHTML property to modify the content of an HTML element

type attribute: Use JavaScript or jQuery to dynamically modify the type attribute of an <input> element

Other HTML editing tips

You can add new HTML, change existing elements, or change the element's tag

You can use HTML formatting tags to bold, italicize, or underline text

You can launch an HTML file within the text editor you created it in, or view it in a web browser.

q and Cite Elements:

The **HTML <q> cite attribute** specifies the source of a short inline quotation. It contains a URL pointing to the original document or reference, though it is not displayed to users. It helps in citing the source of a quote.

Syntax

```
<q cite="URL">
```

Attribute Values

It contains a single value **URL** which is used to specify the URL of the quotation. The possible value of the URL is:

- **Absolute URL:** It is used to point out other websites (like cite =”http://www.w3schools.com”)
- **Relative URL:** It is used to point out the page within the website. (like cite=”geeks.htm”).

Example: In this example we creates a webpage with centered text, displaying a heading “GeeksForGeeks”, a subheading about the <q> cite attribute, and a paragraph quoting “GeeksforGeeks” with a citation link.

```
<!DOCTYPE html>
<html>
<head>
   <title>HTML q cite Attribute</title>
</head>
<body style="text-align:center;">
   <h1>GeeksForGeeks</h1>
   <h2>
      HTML <q>cite Attribute
   </h2>

   <p>
      <q cite="https://www.geeksforgeeks.org">
         GeeksforGeeks
      </q>
      A computer science portal for geeks
   </p>
```

```
</body>
</html>
```

dfn ,abbr, and Time elements:

dfn:

The **<dfn> tag** in HTML represents a definition element and is used to represent a defining instance in HTML. Generally, the defining instance is the first use of a term in a document. The <dfn> tag requires a starting as well as an ending tag.

```
<!DOCTYPE html>
<html>
<body>
   <p>
      <dfn>Geeksforgeeks</dfn>
      is a portal for geeks.
   </p>
</body>
</html>
```

Syntax

```
<dfn> Content ...</dfn>
```

The content inside the <dfn> tag can be any of the following:

Term Definition Inside <dfn>: Placing the term's meaning directly within the <dfn> element.

```
<p><dfn>Learn</dfn>HTML from GeeksforGeeks</p>
```

abbr:

The <abbr> tag in HTML is **used to represent abbreviations** and provides additional information about them through the title attribute, which displays a tooltip when hovered over. It helps improve accessibility and SEO by offering context for the abbreviated text.

- It makes text clearer by explaining abbreviations.
- Improves accessibility with helpful tooltips.
- Works with global and event attributes for flexible use.

Syntax:

```
<abbr title=""> Short form </abbr>
```

- title: Displays the full form as a tooltip when hovering over the abbreviation.

```
<!DOCTYPE html>
<html>
<body>
   <p>
      <!-- HTML abbr tag is used here -->
      <abbr title="GeeksforGeeks">GFG</abbr>: A computer science portal for geeks.
   </p>
</body>
</html>
```

- The <abbr> tag defines the abbreviation "GFG" with the full form "GeeksforGeeks" provided in the title attribute.
- When users hover over "GFG," a tooltip displaying "GeeksforGeeks" appears, offering clarity.

Code-Related Elements:

HTML provides a set of **elements tailored for displaying computer code** so that it is easily distinguishable from other text on a webpage. These elements help in formatting and presenting source code in a readable and syntactically correct manner.

The code Tag

The `<code>` tag in HTML is designed to display computer code snippets with fixed formatting for optimal readability. It renders the code in a monospace font, preserving the original spacing and layout. The `<code>` tag also supports both global attributes and event attributes, allowing for flexible styling and interaction.

Syntax:

```
<code> Computer code contents... </code>
```

Note: The program that is written inside the <code> tag has some different font sizes and font types to the basic heading tag and paragraph tag.

Example: The <code> tag displays a **C program within a <pre> tag**, preserving whitespace and formatting. The C program includes the stdio.h library and a main function that prints “Hello Geeks”.

```
<!DOCTYPE html>
<html>
<body>
   <pre>
      <code>
         #include <stdio.h>
         int main() {
            printf("Hello Geeks");
         }
      </code>
   </pre>
</body>
</html>
```

The kbd Tag

The `<kbd>` tag is used to define keyboard input. The text between the `<kbd>` tags represents text that should be typed on a keyboard. This text is typically displayed in the browser’s default monospace font, though a richer effect can be achieved with CSS. The `<kbd>` tag has no specific attributes.

Syntax:

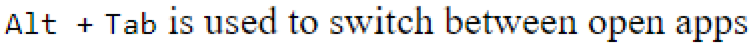
```
<kbd> Contents... </kbd>
```

Example: To demonstrate the implementation of the <kbd> Tag. The <kbd> tag displays keyboard keys “Alt”, “+”, and “Tab” within the styled text.

```
<!DOCTYPE html>
<html>
<head>
   <title>The kbd tag</title>
```

```
  <style>
    body {
      text-align: center;
    }
  </style>
</head>
<body>
  <kbd>Alt </kbd>
  <kbd>+</kbd>
  <kbd>Tab</kbd>
  <span>
    is used to switch between open apps
  </span>
</body>
</html>
```

Output:



```
Alt + Tab is used to switch between open apps
```

br and wbr Elements:

br:

The **<br> tag in HTML** is used to create a line break in the text, causing the content that follows it to appear on a new line.

Note: It is a self-closing tag, meaning it does not need a separate closing tag.

```
<!DOCTYPE html>
<html>
<body>
  <p>Dear Reader, <br>This is printed after a line break.</p>
</body>
</html>
```

wbr:

The **<wbr> tag in HTML** stands for word break opportunity and is used to define the position within the text which is treated as a line break by the browser. It is mostly used when the used word is too long and there are chances that the browser may break lines at the wrong place for fitting the text.

Syntax:

```
<wbr>
```

Note: The <wbr> tag is new in HTML5 and it does not require end tag.

Below examples illustrate the <wbr> tag in HTML 5:

Example 1:

```
<!DOCTYPE html>
<html>
<body>
   <h1>GeeksforGeeks</h1>
   <h2>&lt;wbr&gt; Tag</h2>
   <!--  It is mostly used when the used word is too long and
          there are chances that the browser may break lines at
          the wrong place -->

   <p>GFGstandsforGeeksforGeeksanditis<wbr>acomputerscienceportalforgeeks.</p>
</body>
</html>
```

Text Elements:

HTML provides many predefined elements that are used to change the formatting of text. The formatting can be used to set the text styles (like – bold, italic, or emphasized, etc.), highlight the text, make text superscript and subscript, etc.

Text Formatting Elements:

<b> and <strong> Tags: Both tags are used to make text bold. The text content of the tag is shown as important information on the webpage.

```
<!DOCTYPE html>
```

```
<html>

<head>
   <title>Bold and strong</title>
</head>

<body>
   <!--Normal text-->
   <p>Normal Text</p>
   <!--Text in Bold-->
   <p><b>Bold Text</b></p>
   <!--Text in Strong-->
   <p><strong> Strong Text</strong></p>
</body>

</html>
```

HTML <i> and <em> Tags: Both tags are used to make the text italic and emphasized. Both the element has opening and closing tags.

Character references:

An HTML **character reference** is an escape sequence of characters that is used to represent another character in the rendered web page.

Character references are used as replacements for characters that are reserved in HTML, such as the less-than (<) and greater-than (>) symbols used by the HTML parser to identify element tags, or " or ' within attributes, which may be enclosed by those characters. They can also be used for invisible characters that would otherwise be impossible to type, including non-breaking spaces, control characters like left-to-right and right-to-left marks, and for characters that are hard to type on a standard keyboard.

There are three types of character references:

Named character references These use a name string between an ampersand (&) and a semicolon (;) to refer to the corresponding character. For example, &lt; is used for the less-than

(<) symbol, and &copy; for the copyright symbol (©). The string used for the reference is often a camel-cased initialization or contraction of the character name.

Decimal numeric character references These references start with &#, followed by one or more ASCII digits representing the base-ten integer that corresponds to the character's Unicode code point, and ending with ;. For example, the decimal character reference for < is &#60;, because the Unicode code point for the symbol is U+0003C, and 3C hexadecimal is 60 in decimal.

Hexadecimal numeric character reference These references start with &#x or &#X, followed by one or more ASCII hex digits, representing the hexadecimal integer that corresponds to the character's Unicode code point, and ending with ;.

Sup,Sub,S,mark:

In HTML, **the <sub> tag** is used for subscript, making text appear slightly below the normal line, while the **<sup> tag** is used for superscript, positioning text slightly above the normal line

SUB

HTML Subscript

The <sub> tag displays text slightly below the regular text baseline. It is commonly used in

- **Chemical formulas**: H2O, CO2
- **Mathematical notations**: x1, y2
- **Technical references**: Aij

```
{...}
  <p>The chemical formula for water is H<sub>2</sub>O.</p>
  <p>The mathematical notation for x squared is x<sub>2</sub>.</p>


{...}
```

In this example <sub>2</sub> makes the “2” appear as a subscript in the formula H2O and x.

SUP

HTML Superscript

The <sup> tag text is slightly above the regular text baseline. It is commonly used in

- **Exponents in mathematics:** x2
- **Chemical isotopes:** 14C
- **Ordinals:** 1st, 2nd

```
{...}
  <p>The formula for squaring a number is x<sup>2</sup>.</p>
  <p>Euler's constant is approximately 2.718<sup>e</sup>.</p>


{...}
```

In this example

- **<sup>2</sup>** makes the “2” appear as a superscript in the formula x^2.
- **<sup>e</sup>** places the “e” in superscript for the approximation of Euler’s constant 2.718^e.

S

- The <s> tag specifies text that is no longer correct, accurate or relevant. The text will be displayed with a line through it.
- The <s> tag should not be used to define deleted text in a document, use the <del> tag for that.

```
<!DOCTYPE html>
<html>
<head>
<style>
s {
  text-decoration: line-through;
}
</style>
</head>
<body>
<p>An s element is displayed like this:</p>
<p><s>Some no-longer correct text.</s></p>
<p>Change the default CSS settings to see the effect.</p>
</body>
</html>
```

Mark:

The <mark> tag in HTML is used to define the marked text. It is used to highlight the part of the text in a paragraph. The <mark> tag is new in HTML 5.

Syntax:

```
<mark> Contents... </mark>
```

```
<!DOCTYPE html>
<html>
<body>
   <h1 style="color: green;">GeeksforGeeks</h1>
   <h2>HTML mark Tag</h2>
   <p>
      <mark>GeeksforGeeks:</mark> It is a
      <mark>computer science</mark> portal for geeks
   </p>
</body>
</html>
```

Small Elements:

The <small> tag in HTML is used to define smaller text, reducing font size. It decreases the font size by one size (from medium to small, from x-large to large). It has a display property of inline.

The <small> tag also supports the Global Attributes and Event Attributes in HTML. This tag is not deprecated, but it is possible to achieve better results or nearly the same effect with CSS.

Syntax:

```
<small> Contents... </small>
```

Example:

```
<html>
<head>
   <title>HTML small Tag</title>
</head>
<body>
   <h1>
      GeeksforGeeks
   </h1>
   <h2>
      &lt;small&gt; Tag
   </h2>
   <!-- html small tag is used here -->
```

```
    <small>
        Welcome to GeeksforGeeks!
    </small>
</body>
</html>
```

Strong Tag:

The HTML <strong> Tag is the parsed tag and is used to show the importance of the text. The content inside it is generally in bold format. The primary purpose of <strong> tag is to provide semantic meaning to the content, indicating its importance to both browsers and developers.

Note: This tag supports the global attributes and event attributes in HTML.

Syntax

```
<strong> Contents... </strong>
```

Example

```
<!DOCTYPE html>
<html>
<body>
    <h1>
        GeeksforGeeks
    </h1>
    <h2>
        &lt;strong&gt; Tag
    </h2>
    <!-- HTML Strong Tag used here -->
    <strong>
        Welcome to geeksforGeeks!
    </strong>
</body>
</html>
```

em, b , u and I Elements:

EM:

HTML <em> tag is used to emphasize text, typically rendering it in italics. It conveys importance or stress semantically and is widely employed for emphasis within paragraphs or sentences.

Note: The <em> tag in HTML supports both Global Attributes and Event Attributes, enabling versatile usage for emphasizing text content.

Syntax:

```
<em> Contents... </em>
```

HTML <em> tag Examples

Example:

```
<!DOCTYPE html>
<html>
<body>
   <h2>HTML em Tag</h2>
   <em>Emphasized text content</em>
</body>
</html>
```

B:

HTML <em> tag is used to emphasize text, typically rendering it in italics. It conveys importance or stress semantically and is widely employed for emphasis within paragraphs or sentences.

Note: The <em> tag in HTML supports both Global Attributes and Event Attributes, enabling versatile usage for emphasizing text content.

Syntax:

```
<em> Contents... </em>
```

HTML <em> tag Examples

Example:

```
<!DOCTYPE html>
<html>

<body>
    <h2>HTML em Tag</h2>
    <em>Emphasized text content</em>
</body>

</html>
```

U:

HTML <em> tag is used to emphasize text, typically rendering it in italics. It conveys importance or stress semantically and is widely employed for emphasis within paragraphs or sentences.

Note: The <em> tag in HTML supports both Global Attributes and Event Attributes, enabling versatile usage for emphasizing text content.

Syntax:

```
<em> Contents... </em>
```

HTML <em> tag Examples

Example:

```
<!DOCTYPE html>
<html>

<body>
    <h2>HTML em Tag</h2>
    <em>Emphasized text content</em>
</body>

</html>
```

I:

HTML <em> tag is used to emphasize text, typically rendering it in italics. It conveys importance or stress semantically and is widely employed for emphasis within paragraphs or sentences.

Note: The <em> tag in HTML supports both Global Attributes and Event Attributes, enabling versatile usage for emphasizing text content.

Syntax:

```
<em> Contents... </em>
```

HTML <em> tag Examples

Example:

```
<!DOCTYPE html>
<html>

<body>
    <h2>HTML em Tag</h2>
    <em>Emphasized text content</em>
</body>

</html>
```

Span Element:

The **HTML <span> tag i**s an inline container that is used to group and apply styles or scripts to specific parts of text or elements within a document.

While it doesn’t affect the layout or appearance on its own, it serves as a target for CSS styling and JavaScript interactions, making it ideal for customizing small portions of content without disrupting the flow of surrounding elements.

Syntax

```
<span class="">Some Text</span>
```

Web page with Character References:

A web page with character references uses codes to represent characters that can't be typed directly into HTML. These codes are also known as character entities or entity codes.

Why use character references?

- To represent characters that are reserved in HTML, such as < and >
- To represent invisible characters, such as non-breaking spaces
- To represent characters that are hard to type on a standard keyboard, such as ©

Examples of character references

- To display a less than symbol, use the character reference &lt; instead of the symbol itself
- To force the browser not to break up space-separated words when wrapping content, use the non-breaking space character entity reference, &nbsp;
- To display a double quotation mark, use the character entity reference &quot;