# **Web Development (Module - 3 HTML)**

**(Q.1)** **What are tags and attributes in HTML?**

**(ANS):**

In HTML (Hypertext Markup Language), tags and attributes are fundamental components used to structure and define web documents. They play essential roles in specifying the structure and appearance of content within a webpage. Here's an overview of what tags and attributes are:

***1. Tags:***

* **Definition:** HTML tags are elements used to enclose and define different parts of content within an HTML document. Tags are enclosed within angle brackets (< and >).
* **Examples:** Common HTML tags include <html>, <head>, <body>, <p>, <a>, <img>, <div>, <ul>, <li>, <h1>, <table>, and many more.
* **Purpose:** Tags define the type of content enclosed within them and establish the structure of a web page. They indicate how elements should be displayed and often have semantic meanings.

***2. Attributes:***

* **Definition:** HTML attributes provide additional information about an HTML element. Attributes are always specified within the opening tag of an element and are followed by an equal sign (=) and a value enclosed in quotation marks.
* **Examples:** Common HTML attributes include src, href, alt, class, id, width, height, style, and target, among others.
* **Purpose:** Attributes modify the behavior or appearance of an HTML element. They can be used to supply information such as source URLs for images (src), hyperlinks (href), alternative text for images (alt), and CSS classes for styling (class). Attributes are key-value pairs that provide context and configuration for elements.

Here are some examples of HTML elements with attributes:

A computer code on a black background

Description automatically generated

In the examples above, the tags **(<a>, <img>, <div>, <p>)** define the type of content, while the attributes **(href, target, src, alt, width, height, class)** provide additional information and control how those elements function or appear.

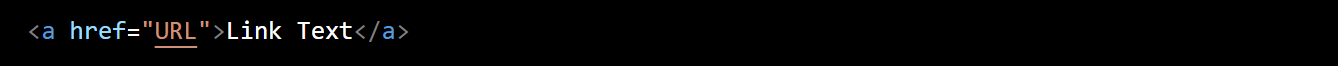
HTML tags and attributes are essential for creating structured, semantically meaningful, and visually appealing web pages. They are the building blocks of web development and are used in conjunction with CSS and JavaScript to create dynamic and interactive websites.

**(Q.2) How to create a Hyperlink in HTML? With Example.**

**(ANS):**

To create a hyperlink in HTML, you use the <a> (anchor) element, which allows you to link to other web pages, documents, or resources.

Here's the basic structure of an HTML hyperlink:



Let's break down the components of this code:

* **<a>:** This is the opening anchor tag, which marks the beginning of the hyperlink.
* **href="URL":** The href attribute specifies the destination URL or web address to which the link will take the user when clicked. Replace "URL" with the actual URL you want to link to.
* **Link Text:** This is the text that will be displayed as the clickable link. You can replace "Link Text" with any descriptive text you want to use for the link.

Here's an example of how to create a simple hyperlink in HTML:

A computer screen with a black background

Description automatically generated

In this example:

* The <a> element surrounds the word "website."
* The href attribute contains the URL **"https://www.example.com,"** which is where the user will be directed when they click the link.
* The text **"website"** is what the user will see as the clickable link.

When you open this HTML document in a web browser, you will see the text **"website"** as a clickable link. Clicking on it will take you to the URL specified in the **href** attribute, which, in this case, is **"https://www.example.com."**

You can use the **<a>** element to create links to other web pages, external websites, email addresses, files, or even specific sections within the same webpage (using anchor links). The **href** attribute determines what the link points to, and the text enclosed within the **<a>** tags becomes the visible link text.

**(Q.3) What are void elements in HTML? With Example.**

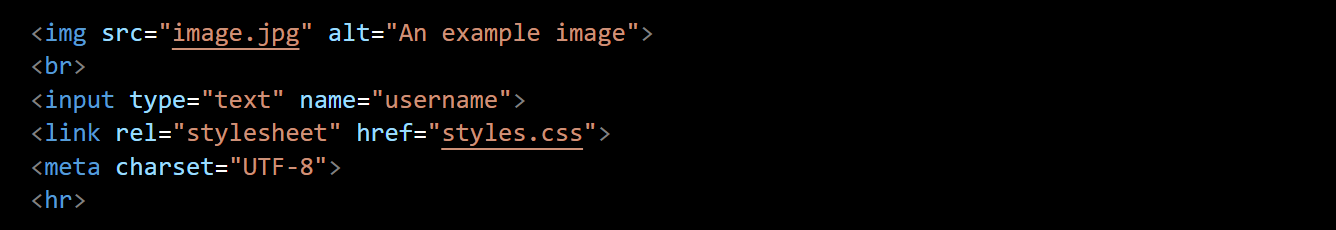
**(ANS):**

Void elements in HTML are elements that do not have a closing tag because they do not contain any content or have content that is considered self-closing.

Void elements are commonly used for inserting multimedia elements, line breaks, images, and other elements that do not require enclosed content. Some common void elements include:

1. **<img>:** Used for embedding images.
2. **<br>:** Represents a line break.
3. **<input>:** Used for creating input fields in forms.
4. **<link>:** Typically used to link external stylesheets.
5. **<meta>:** Contains metadata about the document, like character encoding.
6. **<hr>:** Represents a thematic break or horizontal rule.

Here are examples of void elements in HTML:



**(Q.4) What are different types of lists in HTML? With Example.**

**(ANS):**

HTML provides several types of lists that you can use to organize and structure content. The most common types of lists in HTML include:

***1. Ordered Lists (<ol>):***

Ordered lists are used for items that have a specific order or sequence. Each list item is preceded by a number or another ordered marker (such as letters or Roman numerals).

Example of an ordered list:

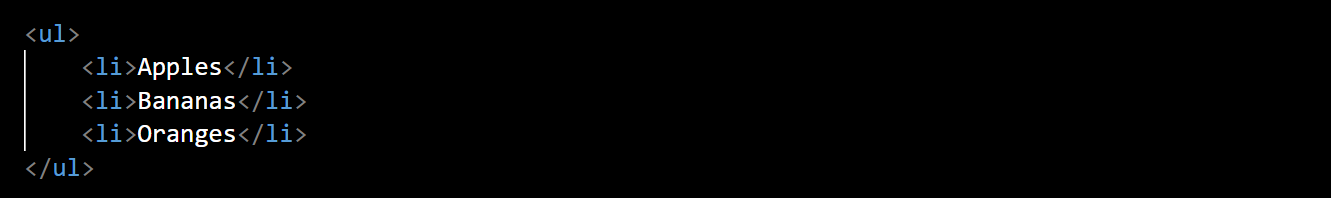
A black background with a black square

Description automatically generated with medium confidence

***2. Unordered Lists (<ul>):***

Unordered lists are used for items that do not have a particular sequence or order. Each list item is typically preceded by a bullet point.

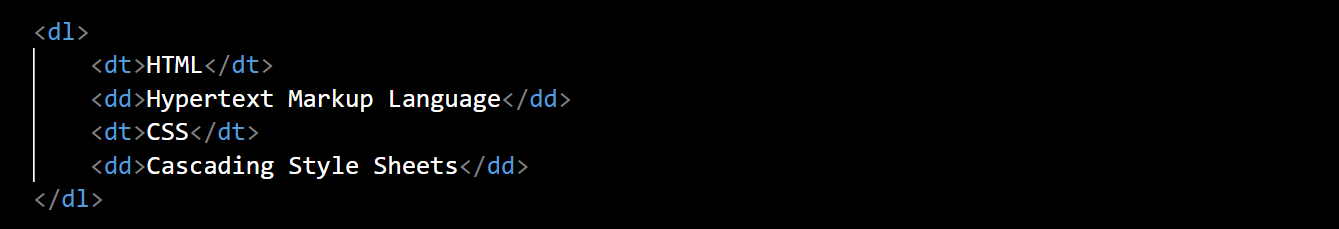
Example of an unordered list:



***3. Definition Lists (<dl>):***

Definition lists are used to group terms and their corresponding definitions. Each term is defined using a **<dt>** (definition term) element, and each definition is provided using a **<dd>** (definition description) element.

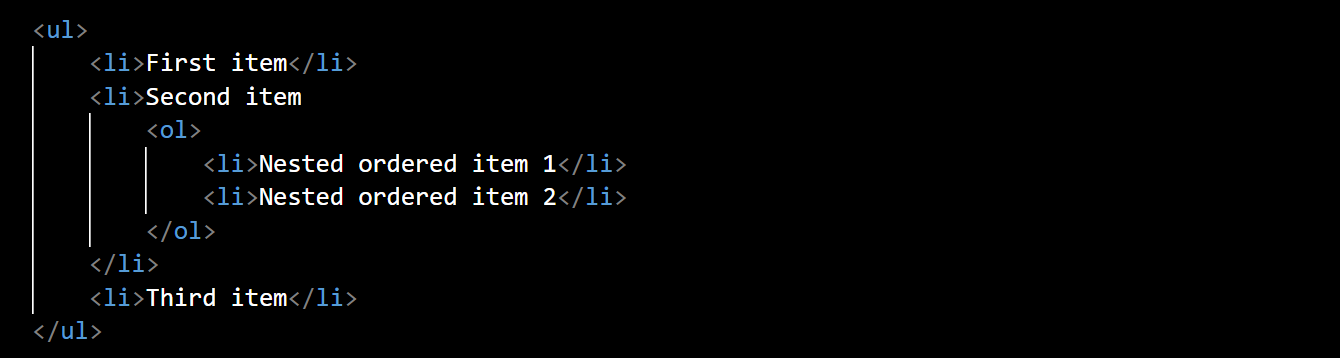
Example of a definition list:



***4. Nested Lists:***

You can also nest lists within each other to create more complex structures. For example, you can have an ordered list within an unordered list or vice versa.

Example of nested lists:



**(Q.5) Are the HTML tags and elements the same thing?**

**(ANS):**

TML tags and HTML elements are related but not the same thing. They are two distinct concepts in HTML.

1. ***HTML Tags:***

* **Definition:** HTML tags are the markup symbols that are used to define elements within an HTML document. Tags consist of opening tags, such as **<p>**, and closing tags, such as **</p>**, with content placed between them.
* **Purpose:** Tags provide instructions to the web browser on how to render the content enclosed within them. They define the beginning and end of an element and specify its type.

Example:



In this example, **<p>** is the opening tag, and **</p>** is the closing tag. Together, they define a paragraph element.

1. ***HTML Elements:***

* Definition: HTML elements are composed of tags, their content, and any attributes. An HTML element includes the opening tag, the content (if applicable), and the closing tag.
* Purpose: Elements are the building blocks of an HTML document. They represent the structure and content of the web page. Elements can be as simple as a single word enclosed in tags or complex, containing nested elements and attributes.

Example:



In this example, **<a>** is an HTML element that represents a hyperlink. It consists of the opening tag **<a>**, the href attribute, the link text **("Visit Example")**, and the closing tag **</a>.**

In summary, while HTML tags are the individual markup symbols **(e.g., <p>, <a>, <div>)**, HTML elements are complete units that include tags, content, and attributes. Elements are used to structure and define the content of an HTML document, while tags are the syntax used to create those elements.

**(Q.6) What are the various formatting tags in HTML?**

**(ANS):**

In HTML, formatting tags are used to control the visual appearance and style of text and content on a web page. While HTML is primarily a structural language, these formatting tags are used in conjunction with CSS (Cascading Style Sheets) for precise control over the presentation of content. Here are some common formatting tags in HTML:

1. **<b> - Bold Text:** This tag is used to make text bold.



1. **<i> - Italic Text:** This tag is used to make text italic.



1. **<u> - Underlined Text:** This tag is used to underline text.



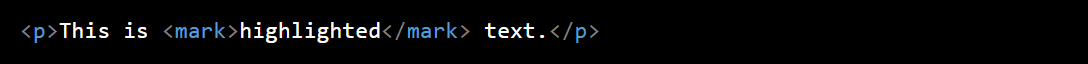
1. **<strong> - Strong Emphasis:** This tag is used to indicate strong emphasis, typically rendered as bold text.



1. **<em>** **- Emphasized Text:** This tag is used to indicate emphasis, typically rendered as italic text.



1. **<mark>** **- Highlighted Text:** This tag is used to highlight or mark text.



1. **<small> - Small Text:** This tag is used to render text in a smaller font size.



1. **<sub> - Subscript Text:** This tag is used to render text as subscript, typically for chemical formulas and mathematical expressions.



1. **<sup>** **- Superscript Text:** This tag is used to render text as superscript, typically for mathematical exponents.



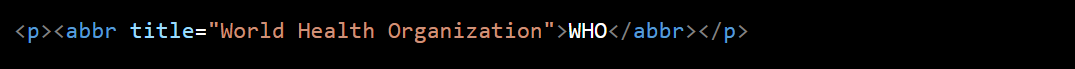
1. **<del>** **- Deleted Text:** This tag is used to indicate deleted or removed text.



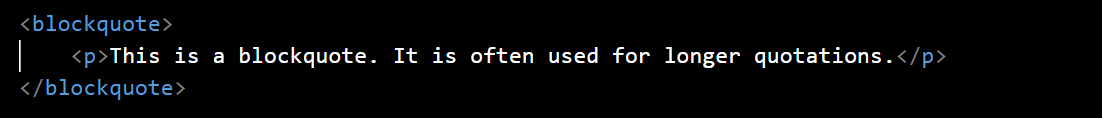
1. **<ins> - Inserted Text:** This tag is used to indicate inserted or added text.



1. **<abbr>** **- Abbreviations:** This tag is used for abbreviations or acronyms, providing an optional title attribute for expanded forms.



1. **<blockquote> - Block Quotation:** This tag is used to create block-level quotations.



1. **<q>** **- Inline Quotation:** This tag is used to mark inline quotations.



1. **<code>** **- Code Text:** This tag is used to represent computer code or code snippets.



1. **<pre>** - Preformatted Text: This tag is used for text that should be displayed in a fixed-width font and preserve whitespace and line breaks.

A black background with white text

Description automatically generated

Remember that while these formatting tags can influence the presentation of text, the recommended practice is to use CSS for styling and presentation, as it provides more flexibility and separation of content and style. HTML tags should primarily be used for semantic structure.

**(Q.7) What are HTML Entities? With Example.**

**(ANS):**

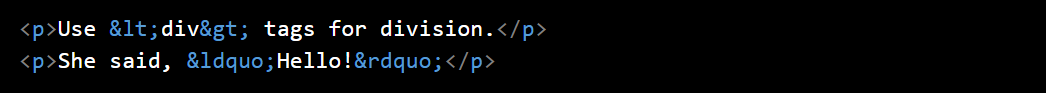
HTML entities are special codes or character references used to represent characters that have special meanings in HTML, or characters that cannot be easily typed or displayed using the standard keyboard or character encoding. HTML entities are especially useful when you need to display reserved characters, symbols, or characters with special purposes within your HTML document.

HTML entities are represented by an ampersand (&), followed by a specific code or name, and ending with a semicolon (;). Here are some common HTML entities:

1. ***Character Entities:***

* **&lt;:** Represents the less-than sign **(<).**
* **&gt;:** Represents the greater-than sign **(>).**
* **&amp;:** Represents the ampersand itself **(&).**
* **&quot;:** Represents double quotation marks **(").**
* **&apos; or &rsquo;:** Represents single quotation marks **(').**

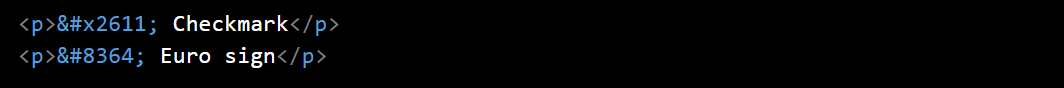
Example:



1. ***Numeric Character References:***

* **&#xHEX; or &#DEC;:** Represents a character using its hexadecimal or decimal Unicode code point. Replace HEX or DEC with the respective code.

Example:



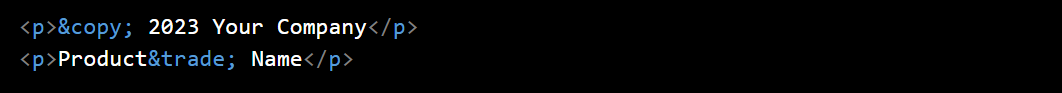
1. ***Special Characters:***

**&copy;:** Represents the copyright symbol **(©).**

**&reg;:** Represents the registered trademark symbol **(®).**

**&trade;:** Represents the trademark symbol **(™).**

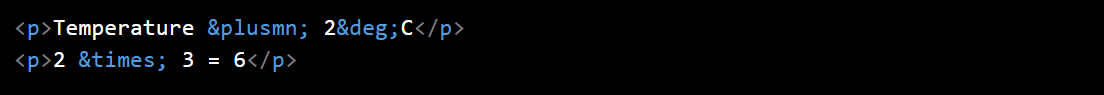
Example:



1. ***Mathematical Symbols:***

* **&plusmn;:** Represents the plus-minus symbol **(±).**
* **&times;:** Represents the multiplication symbol **(×).**
* **&divide;:** Represents the division symbol **(÷).**

Example:



Using HTML entities ensures that special characters and reserved symbols are displayed correctly in web browsers, regardless of the character encoding used. It's important to use entities when needed to avoid issues with character interpretation and rendering in HTML documents.

**(Q.8) How is Cell Padding different from Cell Spacing? With Example.**

**(ANS):**

cellpadding and cellspacing are attributes used in HTML <table> elements to control the spacing and padding around the content within table cells. They serve different purposes:

1. ***Cell Padding (cellpadding):***

* **Purpose:** Cell padding defines the space between the content of a table cell and the cell's borders. It adds spacing inside the cell.
* **Usage:** The cellpadding attribute is applied to the **<table>** element and sets the padding for all the cells in the table.
* **Value:** It takes a numeric value (usually in pixels) or a percentage to specify the amount of padding.

Example:

A black background with a black square

Description automatically generated with medium confidence

In this example, the cellpadding attribute is set to 10, so there will be 10 pixels of space between the content and the cell borders in all cells of the table.

1. ***Cell Spacing (cellspacing):***

* **Purpose:** Cell spacing defines the space between adjacent table cells, creating gaps between cells. It adds spacing between cells.
* **Usage:** The cellspacing attribute is applied to the **<table>** element and sets the spacing between all cells in the table.
* **Value:** It takes a numeric value (usually in pixels) or a percentage to specify the amount of spacing.

Example:

A black background with a black square

Description automatically generated with medium confidence

In this example, the cellspacing attribute is set to 5, so there will be 5 pixels of space between adjacent cells in the table.

To illustrate the difference further, consider a table with both cellpadding and cellspacing attributes set:

A black screen with text and numbers

Description automatically generated with medium confidence

In this case, you would have 10 pixels of padding inside each cell, creating space between the cell content and its border, and 5 pixels of spacing between adjacent cells, creating gaps between the cells themselves.

Adjusting cellpadding and cellspacing can help control the layout and spacing of tables in HTML, making them visually appealing and easier to read.

**(Q.9) How can we club two or more rows or columns into a single row or column in an HTML table? With Example.**

**(ANS):**

In HTML tables, you can merge two or more rows into a single row or columns into a single column using the rowspan and colspan attributes. These attributes allow you to create cells that span across multiple rows or columns. Here's how you can do it:

1. ***Merging Rows (rowspan):***

To merge multiple rows into a single row, you use the **rowspan** attribute on a **<td> (table data)** element. The **rowspan** attribute specifies how many rows the cell should span.

Example:

A computer code on a black background

Description automatically generated

In this example, the first cell in the second row spans two rows (from row 2 to row 3), creating a merged cell.

1. ***Merging Columns (colspan):***

To merge multiple columns into a single column, you use the **colspan** attribute on a **<td> (table data)** element. The colspan attribute specifies how many columns the cell should span.

Example:

A screen shot of a computer code

Description automatically generated

In this example, the second cell in the first row spans two columns (from column 2 to column 3), creating a merged cell.

Merging rows and columns is a useful technique when you need to create more complex table layouts or when you want to group cells together for better readability and presentation of tabular data.

**(Q.10) What is the difference between a block-level element and an inline element? (ANS):**

In HTML and CSS, elements are categorized into two main display **types: block-level elements** and **inline elements.** These display types determine how elements are visually rendered on a web page and how they interact with other elements. Here are the key differences between **block-level and inline elements:**

**Block-Level Elements:**

1. ***Display Type:***

* **Block-level elements** generate a block-level box, which means they create a rectangular box that spans the full width of their parent container. **Block-level elements** start on a new line and stack vertically on top of each other.

1. ***Width & Height:***

* By default, block-level elements take up the full available width of their parent container, unless a specific width is set using CSS. They also respect the height properties, and their height can be adjusted.

1. ***Examples:***

* Common block-level elements include **<div>, <p>, <h1> to <h6>, <ul>, <ol>, <li>, <table>, <form>, and many others.**

1. ***Use Cases:***

* Block-level elements are typically used for structural components like paragraphs, headings, lists, dividers, and containers for other content. They create distinct sections in the layout.

**Inline Elements:**

1. ***Display Type:***

* **Inline elements** do not generate a block-level box. They flow within the content of a block-level element or other **inline elements** and do not start on a new line. Inline elements stack horizontally, next to each other.

1. ***Width & Height:***

* Inline elements only occupy as much width as necessary to contain their content. They do not have a width or height property that can be adjusted.

1. ***Examples:***

* Common inline elements include **<a>, <span>, <strong>, <em>, <img>, <br>, <input>, and <button>.**

1. ***Use Cases:***

* Inline elements are typically used for styling and formatting within the text content. They emphasize or modify specific portions of text, insert images or links, and provide finer-grained control over the appearance of content within block-level elements.

It's important to note that the default display type of an HTML element can be modified using CSS. For example, you can change an inline element to behave like a block-level element by applying the display: block; CSS property, and vice versa. This flexibility allows developers to create custom layouts and styling while adhering to the inherent behaviors of block-level and inline elements.

**(Q.11) What is the use of an iframe tag? With Example.**

**(ANS):**

The **<iframe>** (short for inline frame) tag in HTML is used to embed another HTML document or web page within the current document. It allows you to display content from an external source within a designated area on your webpage. This is particularly useful for integrating third-party content, like maps, videos, or social media feeds, into your site.

Here's how you can use the **<iframe>** tag with an example:



In this example:

* The **<iframe>** element is used to embed content. It has a src attribute that specifies the source URL of the content you want to embed. You can set the width and height attributes to define the size of the iframe.
* The first **<iframe>** embeds a Google Map of a specific location, and the second one embeds a YouTube video. You can replace the src attribute with the URL of any web page or content you want to embed.
* The **frameborder="0"** attribute is used to remove the border around the iframe, and the allowfullscreen attribute enables fullscreen mode for the embedded content (in the case of the YouTube video).
* You can style the iframe using inline **CSS** or link to an **external stylesheet**.

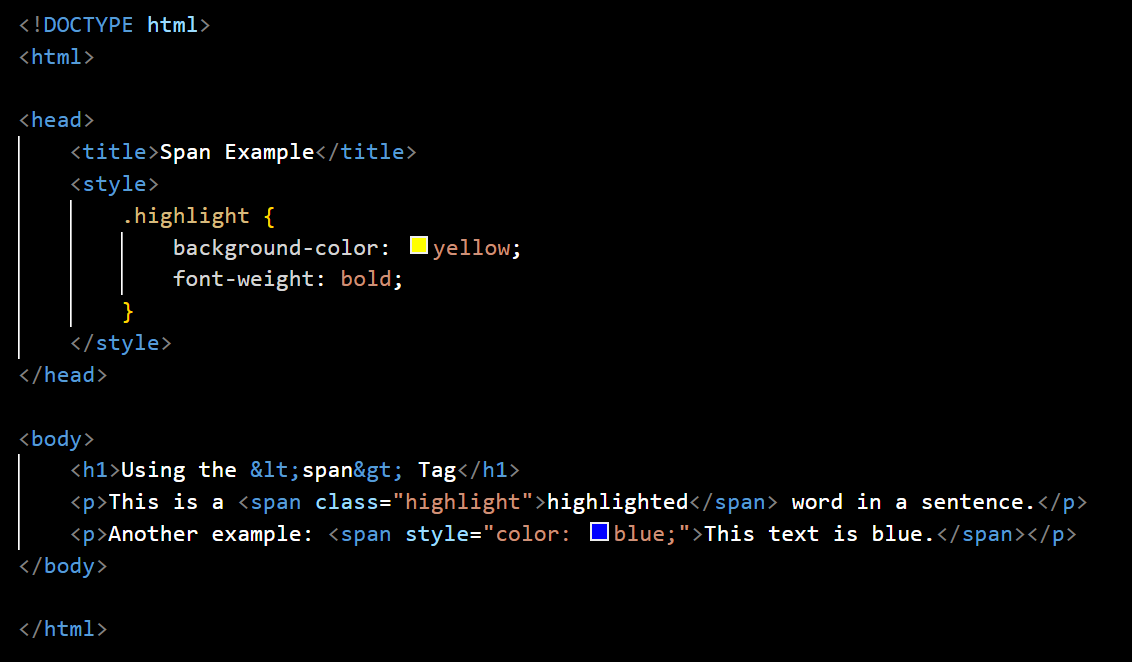
The **<iframe>** tag is a powerful tool for integrating external content into your web pages while maintaining isolation and security boundaries between your site and the embedded content. However, be cautious when embedding content from untrusted sources, as it can potentially introduce security risks.

**(Q.12) What is the use of a span tag? Explain with example?**

**(ANS):**

The **<span>** tag in HTML is an **inline-level element** used for applying styling, scripting, or targeting specific portions of text or inline content within a larger block of text. It doesn't add any inherent formatting or structure to the content but serves as a container for applying CSS styles or JavaScript interactions. The **<span>** element is commonly used to apply CSS classes, inline styles, or JavaScript event handlers to specific text or inline elements.

Here's how you can use the <span> tag with an example:



In this example:

* We have a **<style>** block in the **<head>** section where we define a CSS class named .highlight. This class specifies that the background color should be yellow and the font weight should be bold.
* In the first **<p>** element, we use the **<span>** element with the class attribute to apply the .highlight class to the word **"highlighted."** This makes the word appear with a **yellow** **background and bold text**.
* In the second **<p>** element, we use the **<span>** element with the style attribute to apply inline CSS. This changes the color of the text within the **<span>** element to blue.

The <span> element is particularly useful when you need to apply styling or JavaScript functionality to only a portion of the text or inline content within a larger block of text. It provides a way to target specific elements without affecting the entire block, allowing for fine-grained control over presentation and interaction.

**(Q.13) What is the use of a span tag? Explain with example?**

**(ANS):**

In HTML, there are several tags and elements you can use to separate and structure different sections of text within a web page. These tags help organize content and provide semantic meaning to the text. Here are some common tags used to separate sections of text:

1. ***Headings (<h1> to <h6>):***

* Headings are used to define section titles and hierarchies within a document.

**<h1>** represents the highest-level heading, while **<h6>** is the lowest-level heading.

Example:

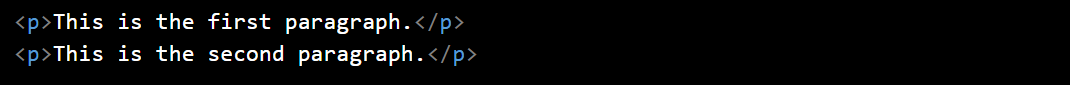
A screen shot of a computer

Description automatically generated

1. ***Paragraphs (<p>):***

* The **<p>** element is used to define paragraphs of text. It separates text into distinct blocks.

Example



1. ***Divisions (<div>):***

* The **<div>** element is a generic container used to group and separate content. It is often styled with CSS for layout purposes.

Example:

A screenshot of a computer screen

Description automatically generated

1. ***Sections (<section>):***

* The **<section>** element is used to define thematic sections of content within a document. It provides semantic meaning to the content.

Example

A screen shot of a computer code

Description automatically generated

1. ***Article (<article>):***

* The **<article>** element is used to define self-contained and independent pieces of content, such as news articles or blog posts.

Example:

A blue and white text on a black background

Description automatically generated

1. ***Headers (<header>) and Footers (<footer>):***

* The **<header>** and **<footer>** elements are used to mark the beginning and end of header and footer sections, respectively.

Example:

A computer screen with a black background

Description automatically generated

These HTML tags help structure and organize content into meaningful sections within a web page, making it easier to style and format the text and improving accessibility and search engine optimization (SEO)

**(Q.14) What is difference between HTML and XHTML?**

**(ANS):**

**HTML (Hypertext Markup Language)** and **XHTML (Extensible Hypertext Markup Language)** are both markup languages used to create and structure content on the web. However, there are some key differences between the two:

1. ***Syntax Rules:***

* **HTML:** HTML has a more forgiving syntax. It allows for certain deviations, such as unclosed tags or mixed case tag names, without causing major issues in rendering
* **XHTML:** XHTML follows a stricter XML-based syntax. All elements and attributes must be properly nested, closed, and written in lowercase. Failure to adhere to these rules may result in errors.

1. ***Document Structure:***

* **HTML:** In HTML, there is often more flexibility in terms of document structure. It allows for a looser structure with optional elements.
* **XHTML:** XHTML enforces a stricter document structure. Elements must be properly nested, and elements like **<head>, <body>, and <html>** are required.

1. ***Quotation Marks:***

* **HTML:** HTML allows for the use of single or double quotation marks around attribute values.
* **XHTML:** XHTML requires the use of double quotation marks around attribute values.

Example (HTML):



Example (XHTML):



1. ***Self-Closing Tags:***

* **HTML:** In HTML, self-closing tags like **<br>, <img>, and <input>** are often written without a closing slash, and browsers understand them.
* **XHTML:** XHTML requires self-closing tags to include a closing slash **(e.g., <br />, <img />, <input />).**

Example (HTML):



Example (XHTML):



1. ***Error Handling:***

* **HTML:** Browsers tend to be forgiving of minor errors in HTML documents and will often attempt to render the page even if there are syntax issues.
* **XHTML:** XHTML documents must be well-formed, and browsers are less forgiving of errors. A small mistake can lead to a page not rendering correctly.

1. ***MIME Type:***

* **HTML:** HTML documents are served with the MIME type **text/html.**
* **XHTML:** XHTML documents are served with the MIME type **application/xhtml+xml or application/xml.**

1. ***Script Handling:***

* **HTML:** In HTML, scripts (JavaScript, for example) can be embedded directly within the document using script tags.
* **XHTML:** In XHTML, it's recommended to use external scripts, and script tags should be placed within the <head> section.

While XHTML was designed to be more strict and XML-compliant, HTML5 has become the dominant standard for web development due to its ease of use and better support for modern web features. HTML5 incorporates many features of both HTML and XHTML, and it's widely adopted for creating web pages today.

**(Q.15) What is SVG?**

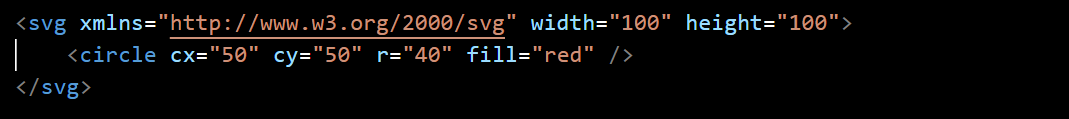
**(ANS):**

**SVG**, which stands for Scalable Vector Graphics, is an XML-based vector image format used for displaying two-dimensional vector graphics on the web. Unlike raster image formats like **JPEG or PNG**, which use a grid of pixels to represent images, SVG uses mathematical descriptions of shapes, paths, and curves to define graphics. This makes SVG images resolution-independent and ideal for displaying graphics that need to be resized or scaled without losing quality.

***Key features and characteristics of SVG include:***

1. **Vector Graphics:** SVG is designed for representing vector graphics, which are images defined by geometric shapes like lines, curves, and polygons. This allows for smooth scaling and resizing of images without pixelation.
2. **XML-Based:** SVG is based on **XML (Extensible Markup Language),** which means SVG images are text-based and can be created and edited with text editors or generated dynamically using scripts.
3. **Scalability:** SVG images can be scaled up or down without loss of quality. This is especially useful for responsive web design, where images need to adapt to different screen sizes and resolutions.
4. **Interactivity:** SVG supports interactivity through the use of JavaScript. You can add event handlers to SVG elements, making it possible to create interactive graphics and animations.
5. **Accessibility:** SVG can be made accessible to people with disabilities by including descriptive text and metadata, which screen readers and assistive technologies can use to convey information to users.
6. **Small File Sizes:** SVG files are typically smaller in size compared to equivalent raster images, which can lead to faster page loading times.
7. **CSS Styling:** SVG elements can be styled using CSS, allowing for a high degree of control over the appearance of graphics.
8. **Wide Browser Support:** SVG is supported by all modern web browsers, making it a reliable choice for web graphics.
9. **Use Cases:** SVG is commonly used for a wide range of web graphics, including icons, logos, charts, diagrams, maps, and interactive graphics.

Here's a simple example of an SVG image that draws a red circle:



In this example, the **<svg>** element defines the **SVG** canvas, and the **<circle>** element is used to draw a red circle. The **cx** and **cy** attributes specify the circle's center coordinates, and the **r** attribute defines the radius. The fill attribute sets the **fill** color to red.

SVG is a versatile and powerful format for creating and displaying graphics on the web, and it's widely used in web development and design.

**(Q.16) What is the ‘class’ attribute in HTML? With Example.**

**(ANS):**

In HTML, the **class** attribute is used to assign one or more class names to an HTML element. Class names are used as identifiers or labels that can be used to apply CSS styles, JavaScript functionality, or other processing to specific elements on a web page. The **class** attribute allows you to group and style multiple elements in a consistent way or target them with CSS and JavaScript.

Here's how the **class** attribute is used with an example:

A screen shot of a computer code

Description automatically generated

In this example:

* The **<style>** block in the **<head>** section defines two CSS classes: **.highlight** and **.warning**. The **.highlight** class sets the background color to yellow and the font weight to bold, while the **.warning** class sets the text color to red.
* The **<h1>** element is assigned the **class** attribute with the value **"highlight"**. This means that the heading will have a yellow background and bold text because it inherits the styles defined in the **.highlight** class.
* The **<span>** element within the second **<p>** element is also assigned the **class** attribute with the value **"highlight"**. This results in the word **"highlighted"** having the same styling as the **<h1>** element.
* The third **<p>** element is assigned the **class** attribute with the value **"warning"**, causing its text to be displayed in red.

The **class** attribute is a fundamental part of web development and is widely used to apply styles and behaviors to specific elements on a web page. It allows for the separation of content and presentation, making it easier to maintain and style complex web pages. Additionally, it plays a crucial role in JavaScript and can be used to select and manipulate elements using JavaScript functions and libraries.

**(Q.17) What is the difference between the ‘id’ attribute and the ‘class’ attribute of HTML elements? With Example.**

**(ANS):**

The **id** and **class** attributes in HTML are both used to identify and target specific elements for styling with CSS or for manipulation with JavaScript. However, there are some key differences between the two attributes:

***id Attribute:***

1. ***Uniqueness:***

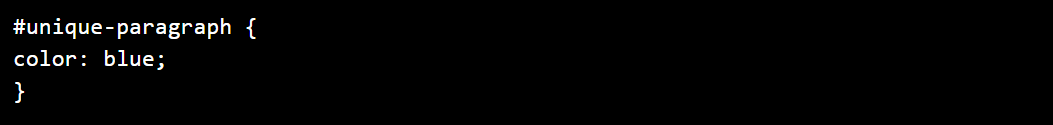
* The id attribute must be unique within a single HTML document. It can only be assigned to one element on the page.
* It is used to uniquely identify a single element, and no other element in the document should have the same id.

1. ***Selector:***

* When styling or selecting elements using CSS or JavaScript, you can directly target an element by its **id** using the **#** selector.

Example:

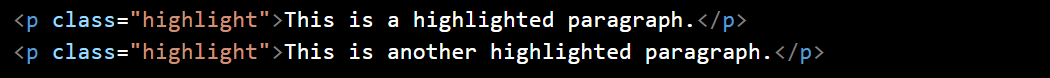


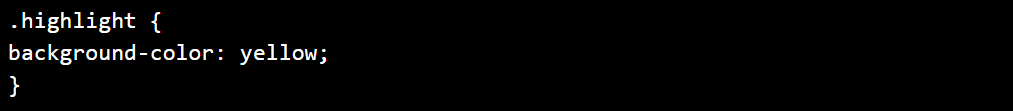


***class Attribute:***

1. ***Multiple Elements:***
   * The **class** attribute can be assigned to multiple elements on the same page. Multiple elements can share the same class.
   * It is used to apply common styles or behaviors to a group of elements.
2. ***Selector:***
   * When styling or selecting elements using CSS or JavaScript, you can target multiple elements with the same class using the **.** selector.

Example:





***Comparison Example:***

Here's an example that illustrates the difference between the **id** and **class** attributes:



In this example:

* The first paragraph has the **id** attribute set to **"unique-paragraph"**. It is uniquely identified and styled with blue text.
* The second and third paragraphs both have the **class** attribute set to **"highlight"**. They share the same class and are styled with a yellow background.

In summary, the **id** attribute is used to uniquely identify a single element on a page, while the **class** attribute is used to group multiple elements together for common styling or manipulation. It's important to use **id** when you need to uniquely target a specific element and use **class** when you want to apply styles or behaviors to multiple elements with shared characteristics.

**(Q.18) How to insert a picture into a background image of a web page? With Example.**

**(ANS):**

To insert an image into the background of a web page, you can use CSS to set the background image property. Here's an example of how to do it:

A computer screen shot of text

Description automatically generated

In this example:

* We use the **<style>** block within the **<head>** section to define CSS rules for the page's styling.
* The **background-image** property is applied to the **body** element, setting the background image to **'background.jpg'**. You should replace **'background.jpg'** with the actual URL or path to your background image.
* The **background-size: cover;** property ensures that the background image covers the entire viewport, scaling as needed.
* **background-repeat: no-repeat;** prevents the background image from repeating.
* **background-attachment: fixed;** fixes the background image in place, so it doesn't scroll with the content.

Additionally, we have a **.content** CSS class that adds a semi-transparent background to the content on top of the background image.

Remember to replace **'background.jpg'** with the actual path or URL of the image you want to use as the background. Adjust the other CSS properties to achieve the desired appearance.

This example demonstrates a simple way to insert an image as the background of a web page. You can further customize the styling and layout as needed to match your design preferences.

**(Q.19) How are active links different from normal links?**

**(ANS):**

Active links and normal links refer to the state of hyperlinks on a web page, and they are part of the user interface and interaction design of websites. Here's how active links differ from normal links:

1. ***Normal Links:***

* Normal links, also known as "regular links" or "default links," are the standard hyperlinks that you see on web pages.
* These links have a default appearance, which often includes an underline and a distinct color (typically blue) to indicate that they are clickable.
* When you hover your mouse pointer over a normal link, it may change appearance (e.g., the color may change) to provide visual feedback to the user that the link is interactive.

1. ***Active Links:***

* Active links, on the other hand, represent links that are currently being interacted with. They reflect the link's state when it's clicked or activated by the user.
* The active state of a link is usually a temporary change in appearance to indicate that the link has been clicked or is currently being pressed (for touchscreen devices). This change helps users understand that their action has been registered.
* Common changes for active links include altering the link's color, removing the underline, or adding a subtle shadow to simulate a button press.

The sequence of states for a typical link interaction includes:

* ***Normal (Default):*** The link appears in its default state.
* ***Hover (Mouseover):*** The link's appearance changes when the mouse pointer is placed over it.
* ***Active (Clicked/Pressed):*** The link's appearance changes temporarily when it's clicked or pressed.

Here's an example of a simple CSS rule that changes the appearance of a link in its active state:

A black background with white text

Description automatically generated

In this example, when a user clicks on the link, it will momentarily turn red and lose its underline to indicate the active state. Once the user releases the mouse button, it will return to its default state.

Active links play an essential role in providing visual feedback to users during interactions with web pages. They help users understand that their action has been recognized and can improve the overall user experience by making navigation more intuitive.

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

**(ANS):**

In HTML, there is a distinction between logical tags and physical tags, although this distinction is somewhat historical and less relevant in modern web development. These terms were more significant in older versions of HTML, such as HTML 4, and they refer to how elements are styled and how they relate to the document's structure:

1. ***Logical Tags:***

* Logical tags are HTML elements that describe the structural or semantic meaning of content without specifying how the content should be presented.
* They are focused on defining the purpose or meaning of content rather than its appearance.
* Logical tags are often used to create well-structured and semantically meaningful web documents that are accessible to assistive technologies (e.g., screen readers).
* Examples of logical tags include **<em>** (emphasis), **<strong>** (strong importance), **<abbr>** (abbreviation), **<blockquote>** (block quotation), and **<cite>** (citation).

Example:



1. ***Physical Tags:***

* Physical tags, on the other hand, are HTML elements that specify the presentation or visual formatting of content.
* They are more concerned with how content should appear on the screen rather than its semantic meaning.
* Physical tags were commonly used for layout and styling purposes in older versions of HTML (e.g., HTML 3.2 and HTML 4).
* Examples of physical tags include **<font>** (for setting font properties), **<b>** (for bold text), **<i>** (for italic text), and attributes like **align**, **bgcolor**, and **width**.

Example:



It's important to note that in modern web development, there has been a shift away from physical tags and presentational attributes in favor of using Cascading Style Sheets (CSS) to control the presentation of content. CSS separates the structure (HTML) from the presentation, allowing developers to apply styles consistently and create responsive designs.

HTML5, the current HTML standard, encourages the use of semantic, logical tags to describe content while leaving the styling to CSS. This approach improves accessibility and the overall maintainability of web documents.

In summary, logical tags focus on the meaning and structure of content, while physical tags are more concerned with visual formatting. In modern web development, it's recommended to use logical tags and CSS for styling to create accessible and well-structured web pages.