# **Module (HTML) -1**

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

**Ans**:-   
**HTML Elements:** An HTML element is a building block of a webpage. It consists of a start tag, content, and an end tag. For example, `<p>` is the start tag of a paragraph element, and **`</p>`** is its end tag. The content goes between these tags.

**HTML Tags**: HTML tags are markers that define the start and end of HTML elements. Tags are made up of the element name enclosed in angle brackets. In the paragraph example, **`<p>`** is the opening tag, and **`</p>`** is the closing tag.

So, in short, HTML tags are the markup used to define the structure of HTML elements. Elements consist of tags and the content between them.

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

**Ans:-**

**1. Tags:**

**-Definition:** Tags are like labels that tell the browser where an element begins and ends.

**-Example:** `<p>` is the opening tag of a paragraph, and `</p>` is the closing tag. Everything in between is the content of the paragraph.

**2. Attributes:**

**- Definition:** Attributes are extra information we add to the opening tag to change or customize an element.

**- Example:** In`<a href="https://www.google.com">`, `href` is an attribute that provides the hyperlink reference (the link's destination).

So, tags define the structure of content, and attributes provide additional details about how elements should behave or appear.

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

**Ans**: Void elements in HTML are elements that don't have a closing tag. They are used for inserting content or providing specific functionalities, and they don't contain any content themselves. Instead of a closing tag, void elements are self-closed or closed with a trailing slash in XHTML.

* **<img>**: Represents an image.
* **<br>** : Represents a line break.
* **<hr>** : Represents a horizontal rule.
* **<input>**: Represents an input control.
* **<meta>**: Represents metadata about an HTML document.
* **<link>**: Represents an external resource link.

Note that void elements do not have a closing tag, and if you are working with XHTML, you can use a self-closing tag like <img />. In HTML5, the self-closing slash is optional, and <img> is equivalent to <img />.

1. **What are HTML Entities ?**

**Ans:** HTML entities are special codes used to represent reserved characters, symbols, and certain formatting in HTML (Hypertext Markup Language). In HTML, some characters have special meanings, such as the less-than sign ("<") and greater-than sign (">"), which are used to define HTML tags. If you want to display these characters as part of your content rather than as HTML code, you use HTML entities.

HTML entities are composed of the ampersand (`&`) followed by a specific code and terminated by a semicolon (`;`). Here are a few common HTML entities:

1. `&lt;` represents the less-than sign (`<`).

2. `&gt;` represents the greater-than sign (`>`).

3. `&amp;` represents the ampersand (`&`).

4. `&quot;` represents a double quotation mark (`"`).

5. `&apos;` represents an apostrophe or single quotation mark (`'`).

**For example,** if you want to display the text "3 < 5" in your HTML document, you would use the following code:

//html

3 &lt; 5

//

When the HTML page is rendered, the browser will display it as "3 < 5," and the less-than sign will not be interpreted as the beginning of an HTML tag.

Using HTML entities is essential when you want to display special characters without them being treated as part of the HTML markup.

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

Ans: In HTML, you can create different types of lists to organize and structure content. The three main types of lists are:

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

An ordered list is a list in which the order of the items matters. Each list item is marked with a number or another type of sequential marker.

- Use the `<ol>` tag to define an ordered list, and `<li>` tags to define each list item.

- **Example:**

html

<ol>

<li>First item</li>

<li>Second item</li>

<li>Third item</li>

</ol>

**This will be Output as**:

1. First item

2. Second item

3. Third item

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

- An unordered list is a list in which the order of the items does not matter. Each list item is marked with a bullet point or another type of marker.

- Use the `<ul>` tag to define an unordered list, and `<li>` tags to define each list item.

**-Example:**

**//html**

<ul>

<li>Item 1</li>

<li>Item 2</li>

<li>Item 3</li>

</ul>

**\\**

**This will be output as:**

- Item 1

- Item 2

- Item 3

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

- A definition list is a list of terms and their corresponding definitions. It is created using the `<dl>` tag for the list, `<dt>` for the term (definition term), and `<dd>` for the definition itself.

**- Example:**

**//html**

<dl>

<dt>Term 1</dt>

<dd>Definition 1</dd>

<dt>Term 2</dt>

<dd>Definition 2</dd>

<dt>Term 3</dt>

<dd>Definition 3</dd>

</dl>

**\\\**

**This will be output as:**

Term 1

Definition 1

Term 2

Definition 2

Term 3

Definition 3

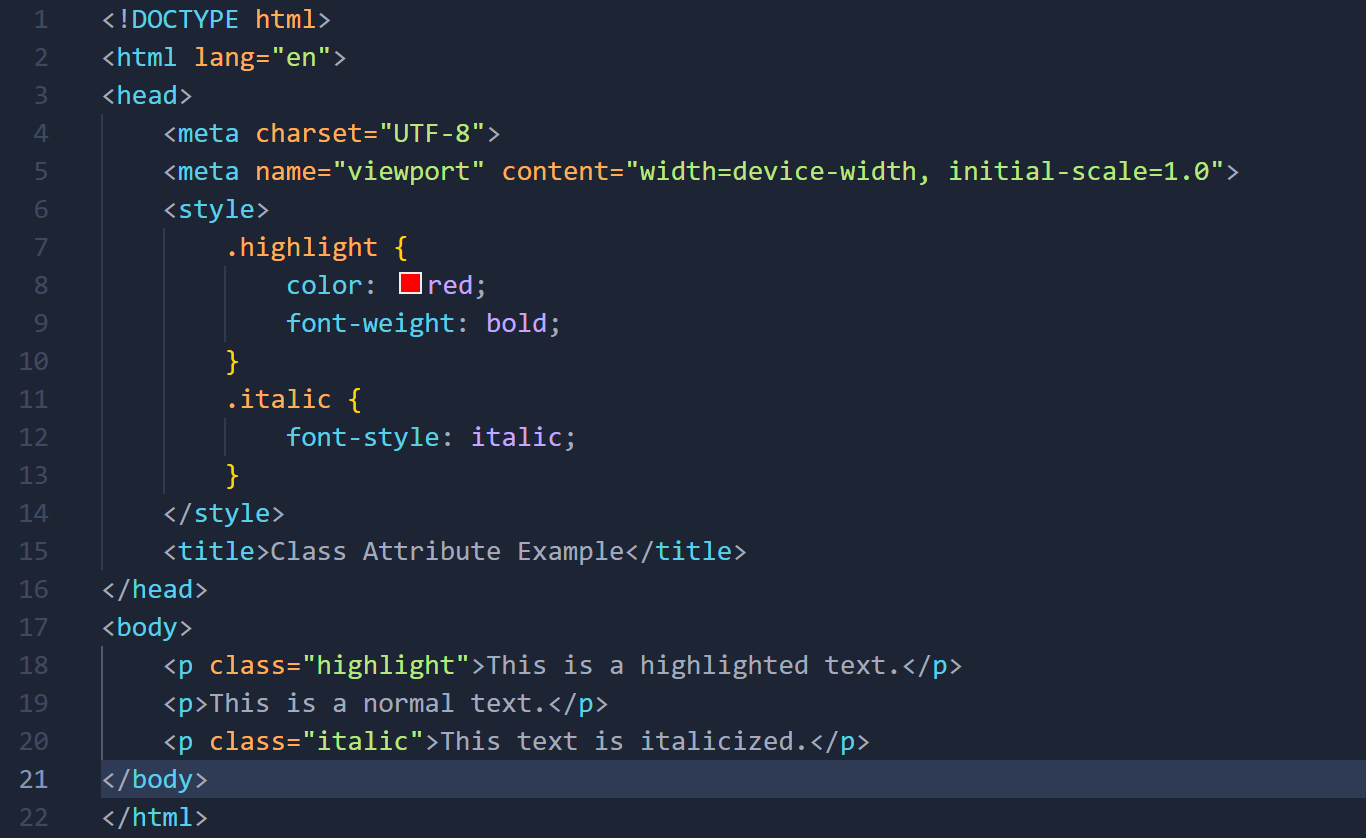
Remember, HTML lists are a way to organize and structure content on a webpage, and they can be styled using CSS to achieve different visual effects**.**

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

**Ans-** The “**Class**” attribute in HTML is used to assign one or more class names to an HTML element. Classes are a way to apply styles and other properties to multiple elements on a page without having to repeat the same attributes for each individual element.

The **“Class”** attribute is used with various HTML elements, and its value is a space-separated list of class names. **CSS (Cascading Style Sheets)** is then used to define the styles associated with each class.

**Here's a simple example:**



In this example, the **Class** attribute is used to assign two different classes, **"highlight"** and **"italic,"** to different paragraphs. The associated styles for these classes are then defined in the embedded **<style>** block. The first paragraph with the class **"highlight"** will have red and bold text, and the third paragraph with the class **"italic"** will be italicized.

By using the **Class** attribute, you can apply consistent styles to elements across your HTML document and make your styling more modular and maintainable. It also facilitates the separation of concerns between HTML (structure) and CSS (presentation).

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

**Ans-** In HTML, both the “**id**” and “**class**” attributes are used to identify and style elements, but they serve different purposes.

**1. “id” Attribute:**

1. The “id” attribute is used to uniquely identify a specific HTML element on a page.
2. The value assigned to the `id` attribute must be unique within the entire HTML document.
3. It is often used to target a specific element with CSS or JavaScript.
4. The “id” attribute is specified within the opening tag of an HTML element.

**Example:**

**\\\html**

<div id="header">

<h1>Welcome to my website</h1>

</div>

<p id="paragraph1">This is a paragraph.</p>

**Html///**

In this example, the **“id”** attribute is used to uniquely identify the header **“div”** and a paragraph, making it easy to apply specific styles or perform actions on these elements.

**2. “class” Attribute:**

1. The **“class”** attribute is used to group and style multiple HTML elements.
2. The value assigned to the **“class”** attribute can be used by multiple elements on the same page.
3. It is often used to apply common styles to a group of elements or to select elements with JavaScript or CSS.
4. The **“class”** attribute can be specified within the opening tag of an HTML element, and multiple elements can share the same class.

**Example:**

**\\\html**

<p class="important">This is an important paragraph.</p>

<p class="important">Another important paragraph.</p>

**Html///**

In this example, both paragraphs share the same class, allowing you to apply a common style to all elements with the **class "important.**"

In summary, the **`id`** attribute is used for unique identification of a single element, while the **`class`** attribute is used to group and style multiple elements. It's generally a good practice to use **`id`** for unique identifiers and **`class`** for styling and grouping related elements.

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

**Ans -** HTML provides a variety of formatting tags that allow you to structure and style your content. Here are some common formatting tags along with examples:

**1.Heading Tags (`<h1>` to `<h6>`):**

**-Used to define headings of different levels.**

<h1>This is a Heading 1</h1>

------ <h1> to <h6>------

<h6>This is a Heading 6</h6>

**2.Paragraph Tag (`<p>`):**

**-Defines a paragraph.**

<p>This is a paragraph.</p>

**3.Bold Tag (`<b>`):**

**-Represents bold text.**

<p>This is <b>bold</b> text.</p>

**4.Italic Tag (`<i>`):**

-**Represents italicized text.**

<p>This is <i>italic</i> text.</p>

**5.Underline Tag (`<u>`):**

**-Represents underlined text.**

<p>This is <u>underlined</u> text.</p>

6.Strike Tag (`<s>` or `<strike>` or `<del>`):

**-Represents text that has been struck through.**

<p>This is <s>struck through</s> text.</p>

**or**

<p>This is <strike>struck through</strike> text.</p>

**or**

<p>This is <del>struck through</del> text.</p>

**7.Superscript Tag (`<sup>`):**

**-Represents superscript text.**

<p>This is <sup>superscript</sup> text.</p>

**8.Subscript Tag (`<sub>`):**

**-Represents subscript text.**

<p>This is <sub>subscript</sub> text.</p>

**9.Line Break Tag (`<br>`):**

**-Inserts a line break.**

<p>This is some text.<br> This is on a new line.</p>

**10.Horizontal Rule Tag (`<hr>`):**

**-Inserts a horizontal rule (a line).**

<p>This is some text.</p>

<hr>

<p>This is more text.</p>

These are just a few examples of formatting tags in HTML. HTML provides various other tags for text formatting, such as `<strong>` for strong importance, `<em>` for emphasized text, and more. Additionally, CSS is commonly used for more advanced and flexible styling of HTML content.

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

**Ans-** In the context of HTML tables, **`cellpadding**` and **`cellspacing`** are attributes that can be applied to the **`<table>`** element to control the spacing and padding of cells within the table.

**1. Cellpadding:**

**- The `cellpadding` attribute controls the space between the content of a cell and the cell border.**

**- It adds padding inside each cell.**

**\\\html**

<table cellpadding="10" border="1">

<tr>

<td>Cell 1</td>

<td>Cell 2</td>

</tr>

<tr>

<td>Cell 3</td>

<td>Cell 4</td>

</tr>

</table>

**///html**

**In this example, the `cellpadding` is set to 10. This means there will be a padding of 10 pixels between the content of each cell and the cell border.**

**2. Cellspacing:**

**- The `cellspacing` attribute controls the space between cells within the table.**

**- It adds spacing between adjacent cells**.

**\\\html**

<table cellspacing="10" border="1">

<tr>

<td>Cell 1</td>

<td>Cell 2</td>

</tr>

<tr>

<td>Cell 3</td>

<td>Cell 4</td>

</tr>

</table>

**///html**

In this example, the `cellspacing` is set to 10. This means there will be a spacing of 10 pixels between adjacent cells in the table.

**Remember, using inline styling attributes like `cellpadding` and `cellspacing` is considered outdated, and it's recommended to use CSS for styling. Here's an example using CSS:**

**\\\html**

<style>

table {

border-collapse: collapse;

}

td {

padding: 10px;

border: 1px solid black;

}

</style>

<table>

<tr>

<td>Cell 1</td>

<td>Cell 2</td>

</tr>

<tr>

<td>Cell 3</td>

<td>Cell 4</td>

</tr>

</table>

**///html**

**In this CSS example, `border-collapse: collapse;` is used to remove any default spacing between cells, and `padding: 10px;` is used to add padding to the cells.**

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

**Ans**- In HTML, you can use the **`colspan**` and **`rowspan**` attributes to merge cells and create a single cell that spans multiple columns or rows within a table. Here's how you can do it:

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

To merge two or more columns into a single column, use the `colspan` attribute. The value of `colspan` indicates how many columns the cell should span.

**\\\html**

<table border="1">

<tr>

<td>Cell 1</td>

<td colspan="2">Merged Cells</td>

<td>Cell 4</td>

</tr>

<tr>

<td>Cell 5</td>

<td>Cell 6</td>

<td>Cell 7</td>

<td>Cell 8</td>

</tr>

</table>

**///html**

**In this example, the second `<td>` has `colspan="2"`, indicating that it should span two columns.**

**2.Merging Rows (`rowspan`):**

To merge two or more rows into a single row, use the `rowspan` attribute. The value of `rowspan` indicates how many rows the cell should span.

**\\\html**

<table border="1">

<tr>

<td rowspan="2">Merged Cells</td>

<td>Cell 2</td>

<td>Cell 3</td>

<td>Cell 4</td>

</tr>

<tr>

<td>Cell 6</td>

<td>Cell 7</td>

<td>Cell 8</td>

</tr>

</table>

**///html**

**In this example, the first `<td>` has `rowspan=”2”`, indicating that it should span two rows.**

**In the easiest way:**

**- For Columns (Horizontal Merge):**

- Use the `colspan` attribute inside a `<td>` (table data) tag.

**- For Rows (Vertical Merge):**

- Use the `rowspan` attribute inside a `<td>` tag.

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

**Ans**- In HTML, the main difference between block-level elements and inline elements lies in their layout behavior and how they interact with surrounding elements:

**1. Layout Behavior:**

- **Block-level Elements:** Take up the full width available and start on a new line, creating a "block" in the layout. Examples include `<div>`, `<p>`, `<h1>` to `<h6>`.

- **Inline Elements**: Only take up as much width as necessary and do not start on a new line. They flow within the content. Examples include `<span>`, `<a>`, `<strong>`, `<em>`.

**2. Container:**

- **Block-level Elements:** Can contain other block-level and inline elements.

- **Inline Elements:** Typically appear within block-level elements and can only contain other inline elements or data.

**3. Default Styling:**

- Block-level Elements: Often have default styles that include starting on a new line and creating a clear visual separation from the content around them.

- Inline Elements: Typically do not force a new line, allowing them to flow within the content.

**Examples:**

**- Block-level element:**

**\\html**

<div>

This is a block-level element.

<p>This is a paragraph inside the block.</p>

</div>

**///html**

**- Inline element:**

**\\\html**

<p>This is an <span>inline</span> element within a paragraph.</p>

**///html**

Understanding these differences is crucial for controlling the layout and structure of HTML documents.

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

**Ans:-** In HTML, you can create hyperlinks using the `<a>` (anchor) element. The `<a>` element is used to define a hyperlink, and it typically includes an `href` attribute that specifies the URL of the linked resource. Here's an example:

**\\\html**

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

**///html**

**In this example:**

- `<a>` is the anchor element.

- `href` is the attribute that contains the URL (Uniform Resource Locator) of the linked resource.

- The text between the opening `<a>` tag and closing `</a>` tag, in this case, "Visit Example.com," is the clickable text that becomes the hyperlink.

You can also create links to local resources within your website:

\\\html

<a href="/about.html">About Us</a>

///html

In this example, the link points to a local file named "about.html" within the same website.

Additionally, you can create links that open in a new browser window or tab using the `target` attribute:

**\\\html**

<a href="https://www.example.com" target="\_blank">Visit Example.com in a new tab</a>

**///html**

In this example, `target="\_blank"` instructs the browser to open the linked resource in a new tab.

You can also link to email addresses using the `mailto` scheme:

**\\\html**

<a href="mailto:info@example.com">Send us an email</a>

**///html**

In this case, clicking the link will open the default email client with a new message addressed to the specified email address.

Remember to replace the URLs and text in the examples with your own content. Hyperlinks are a fundamental part of HTML, enabling navigation between pages and resources on the web.

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

**Ans:-** The `<iframe>` (inline frame) tag in HTML is used to embed another document or webpage within the current document. It's like a window into another world on your page.

Example:

**\\\html**

<iframe src="https://www.google.com" width="600" height="400" title="Example Website"></iframe>

**///html**

This example embeds the "https://www.goole.com" website within an iframe on your page. The `width` and `height` attributes determine the size of the iframe, and the `title` attribute provides a title for accessibility.

It's important to note a few things about **<iframe>**:

* **Security Concerns**
* **Responsive Design**
* **Accessibility**
* Loading External Content

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

**Ans:-** The `<span>` tag in HTML is a generic container that is used to apply styles or scripting to a specific part of the text within an HTML document. It doesn't have any inherent visual impact on its own but allows you to target and style specific portions of text or inline elements.

Here's a simple explanation with an example:

**\\html**

<p>This is <span style="color: red;">highlighted</span> text.</p>

**///html**

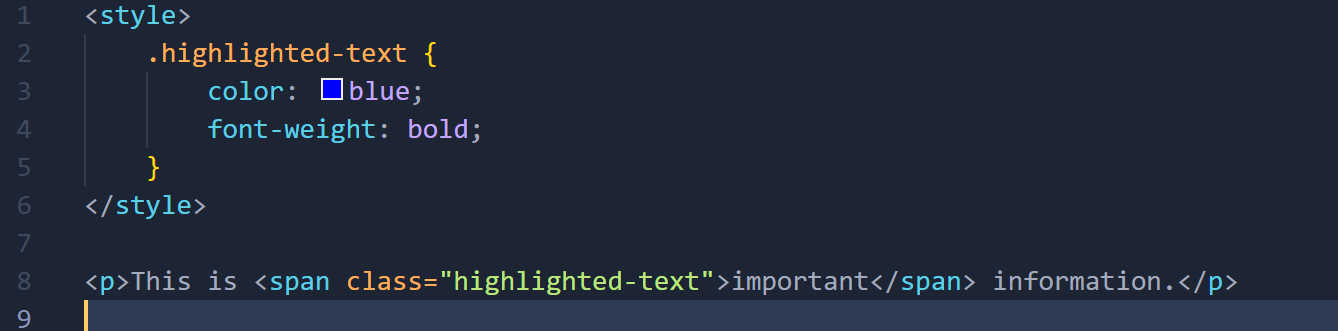
**In this example:**

- `<p>` is a paragraph tag.

- Inside the paragraph, the word "highlighted" is wrapped with a `<span>` tag.

- The `style` attribute within the `<span>` tag applies a red color to the text.

In this case, the `<span>` tag is used to target and style only the word "highlighted" with a specific color. You could apply various styles such as color, font size, font weight, etc., using CSS within the `style` attribute or by applying a class or ID to the `<span>` element and styling it in a separate CSS block.

****

**In this example:**

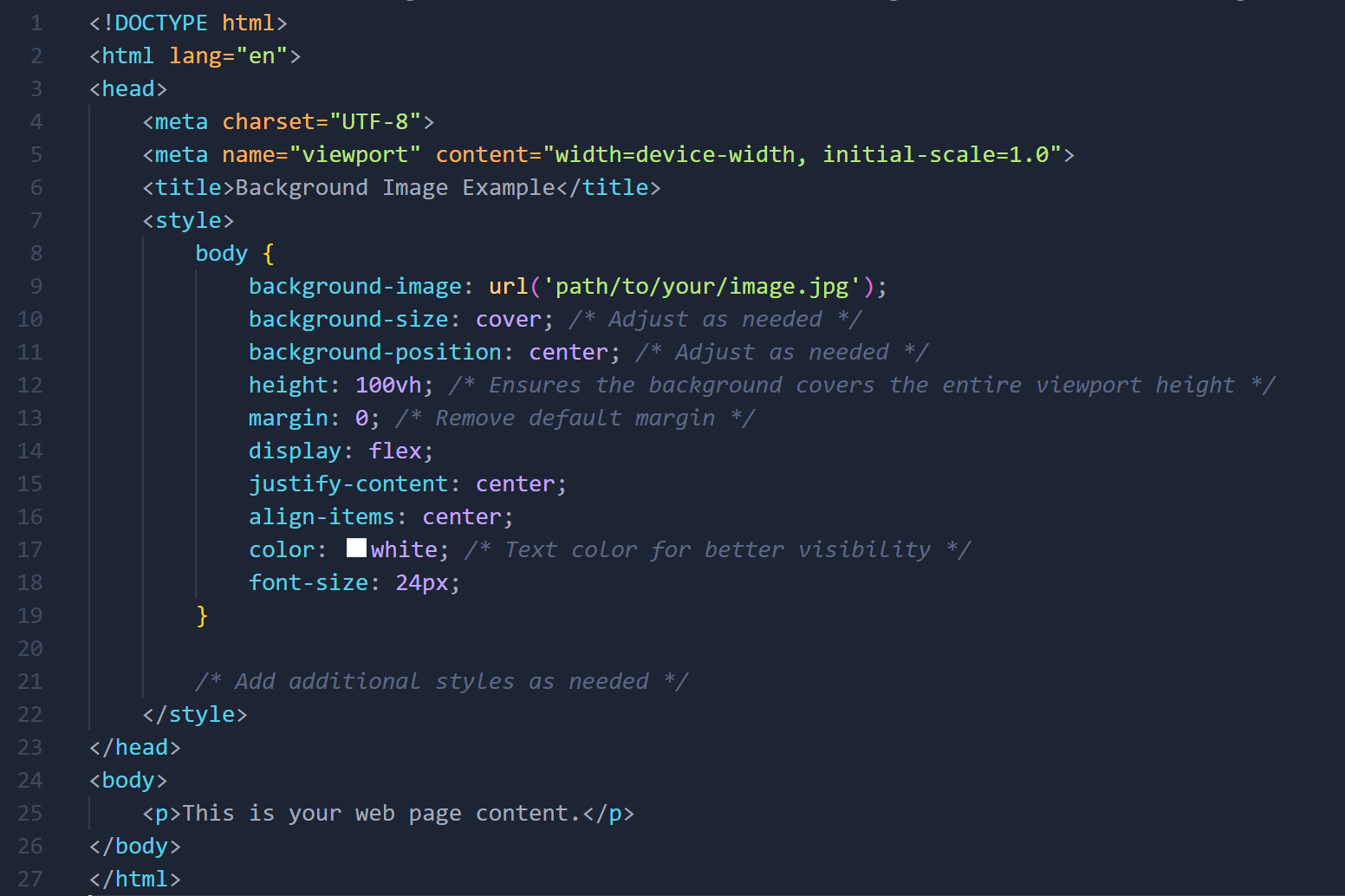
- The `<span>` tag now has a class attribute with the value "highlighted-text."

- The corresponding CSS block styles any element with the class "highlighted-text" with a blue color and bold font weight.

So, in summary, the `<span>` tag is a flexible container that allows you to target and apply styles to specific portions of text or inline elements within your HTML documents. It is often used in conjunction with CSS for styling purposes.

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

**Ans:-** To insert a picture into the background of a web page, you can use the CSS `background-image` property. The easiest way to do this is by applying the background image to the `<body>` element in your HTML document. Here's a simple example:

t

**In this example:**

- The `**body**` element is styled with a `**background-image**` property, setting the URL of the image you want to use.

- `**background-size: cover**;` ensures that the background image covers the entire viewport.

- `**background-position: center**;` centers the background image.

- `**height: 100vh;`** ensures that the background covers the entire viewport height.

- Additional styling like `**margin: 0**;` removes default margins for a cleaner look.

- The content of the page is styled with `**display: flex; justify-content: center; align-items: center;`** to horizontally and vertically center it.

Make sure to replace `**'path/to/your/image.jpg'**` with the actual path or URL of your image. You can use different image formats such as JPEG, PNG, or GIF. Adjust the styles according to your design preferences and needs. This is a basic example, and you can customize it further based on your specific requirements.

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

**Ans:-**

In the context of HTML and web development, **"active links"** and **"normal links"** are not standard terms. However,

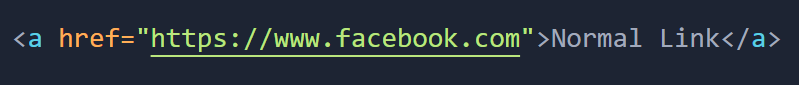
**Here's an explanation with examples:**

**1.Normal Links:**

- Normal links are the default appearance of links before any interaction.

- By default, they are often styled with an underline and a default color.

**- Here's an example:**



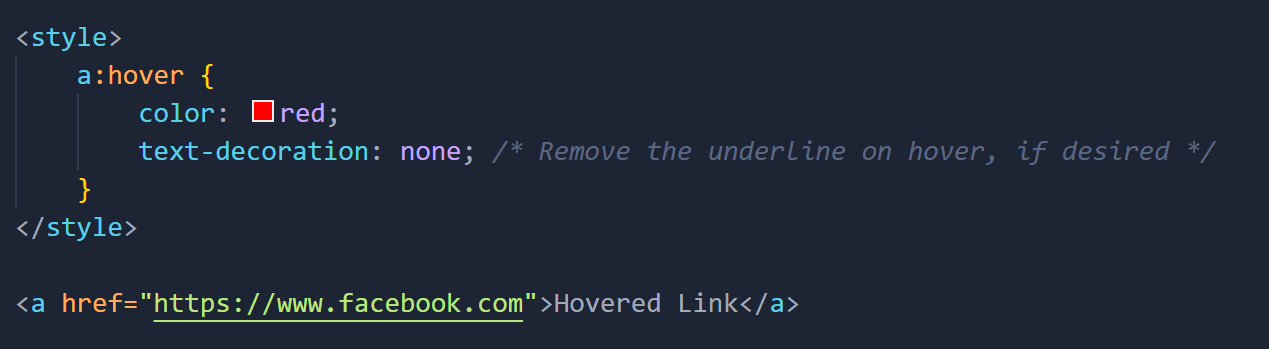
- The appearance of the link is determined by the browser's default styles unless overridden with CSS.

**2. Hovered Links:**

- Hovered links are the appearance of links when the user hovers their cursor over the link.

- They can have different styles to provide visual feedback.

**- Here's an example:**

****

- In this example, the link will turn red when hovered over.

**3. Visited Links:**

- Visited links are the appearance of links that the user has previously visited.

- They are often styled differently to indicate that the user has already clicked on them.

**- Here's an example:**



- In this example, the link will turn purple after the user has visited it.

**4. Active Links:**

- Active links are the appearance of links when they are being clicked or activated.

- This state is typically very brief, as it represents the moment when the link is being clicked.

**- Here’s an example:**



-In this example, the link will turn green during the click.

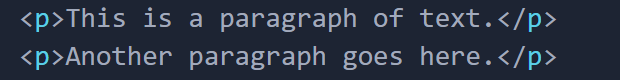
These different states can be styled using **CSS** to provide a consistent and visually appealing user experience. The styles can be customized based on your design preferences.

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

**Ans:-** In HTML, various tags are used to separate sections of text based on their semantic meaning. Here are some common tags:

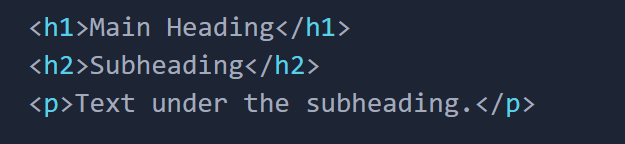
**1. Paragraphs (`<p>`): Used for separating blocks of text.**

**Example:**

****

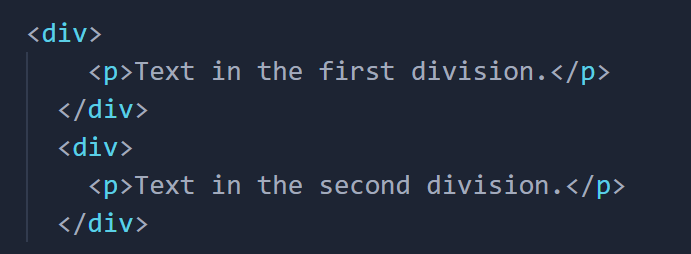
**2. Headings (`<h1>` to `<h6>`): Used for creating hierarchical headings.**

**Example:**

****

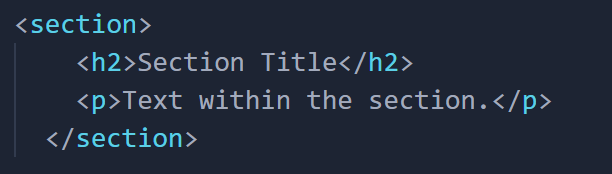
**3. Divisions (`<div>`): Used as a generic container to group content.**

**Example:**

****

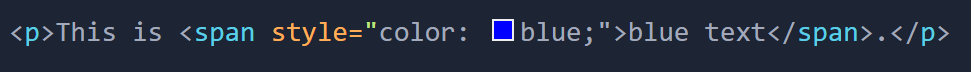
**4. Sections (`<section>`): Used to define sections in a document.**

**Example:**

****

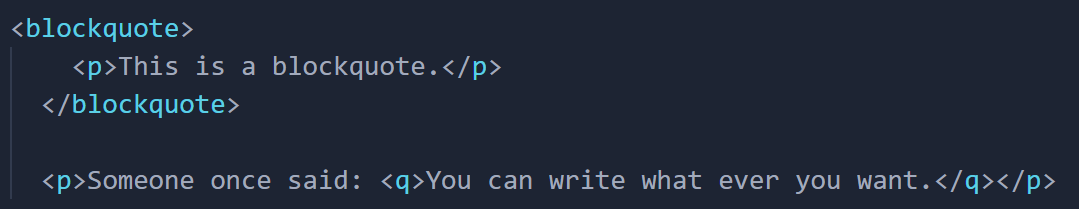
**5. Spans (`<span>`): Used to apply styles or scripting to a specific part of text.**

**Example:**

****

**6. Quotes (`<blockquote>` and `<q>`): Used for quotations.**

**Example:**

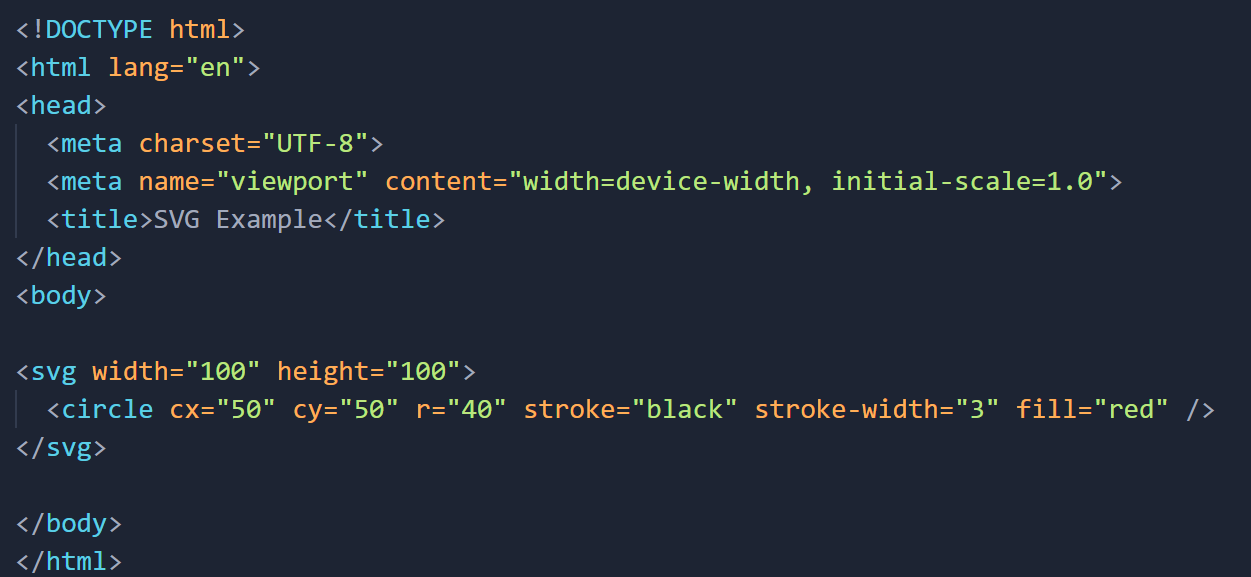
****

These tags help structure and format text content in a meaningful way on web pages. Choose the appropriate tag based on the intended semantic meaning of the content.

1. **What is SVG?**

**Ans:-** **SVG (Scalable Vector Graphics)** is an XML-based markup language for describing two-dimensional vector graphics. In HTML, SVG is used to create vector graphics, such as shapes, lines, and text, that can be scaled and styled without losing quality. SVG is particularly useful for creating graphics like logos, icons, and interactive graphics on the web.

**Here's a simple example of an SVG element in HTML:**



**In this example:**

* The `<svg>` element creates an SVG container.
* The `width` and `height` attributes set the dimensions of the SVG canvas.
* The `<circle>` element is used to draw a circle.
* `cx` and `cy` specify the center of the circle.
* `r` specifies the radius.
* `stroke` and `stroke-width` set the border color and width.
* `fill` sets the fill color.

When you open this HTML file in a browser, you'll see a red circle with a black border. SVG provides a powerful and flexible way to create graphics that can be manipulated using CSS and JavaScript.

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

**Ans:-HTML (Hypertext Markup Language)** and **XHTML (Extensible Hypertext Markup Language)** are both markup languages used to structure content on the web. The primary difference between them lies in their syntax and rules for writing markup.

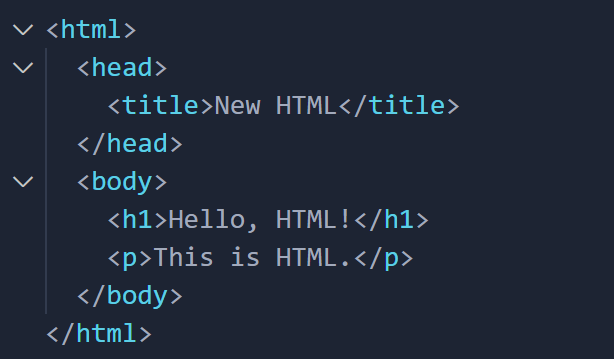
**HTML (HyperText Markup Language):**

**1. Syntax:** HTML has a more forgiving syntax, and it allows for certain deviations and errors without causing the document to be rejected.

**2. Parsing:** HTML parsers are generally more lenient, and they can interpret code even if it's not perfectly structured.

**3. Closing Tags:** In HTML, some tags can be left unclosed (e.g., `<br>`), and attribute values do not need to be enclosed in quotes.

**Example:**



**XHTML (Extensible HyperText Markup Language):**

**1. Syntax:** XHTML has a stricter syntax based on XML rules. It requires well-formed documents with properly nested tags and case-sensitive tag names.

**2. Parsing:** XHTML parsers are less forgiving. Even minor syntax errors can cause the entire document to fail.

**3. Closing Tags:** All tags in XHTML must be properly closed, and attribute values should be enclosed in quotes.

**Example:**



**Analogy:** Imagine HTML as a language where you can communicate in a relaxed and flexible manner, sometimes using colloquial expressions. On the other hand, XHTML is like a formal language where every rule must be strictly followed, and any deviation might lead to miscommunication.

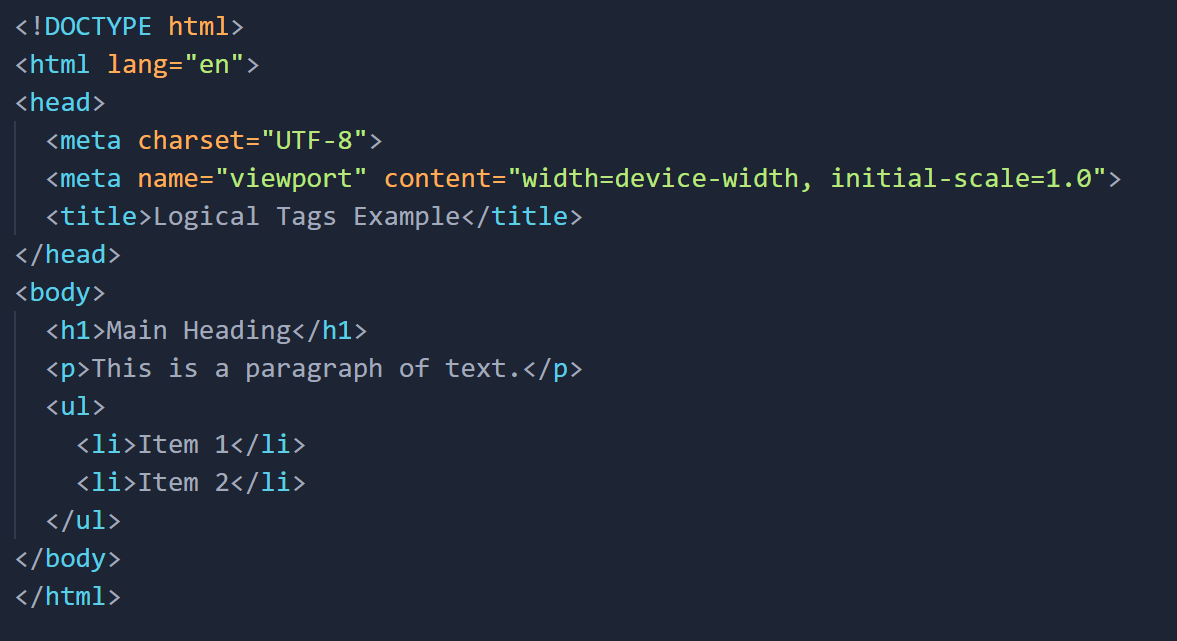
In recent web development, HTML5 has become the standard, and its syntax is more lenient like traditional HTML, while still incorporating some of the stricter rules from XHTML. Most modern websites use HTML5.

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

**Ans:-** In HTML, **"logical tags"** and **"physical tags"** are terms that were historically used to describe the semantic and presentational aspects of markup, respectively. These terms were more relevant in earlier versions of HTML, especially before the adoption of **CSS (Cascading Style Sheets)** as the preferred method for styling web pages.

* **Logical Tags (Semantic Markup):** Logical tags are focused on describing the structure and meaning of content without specifying its appearance. They are more concerned with the semantic interpretation of the content. **Examples of logical tags include `<h1>` for headings, `<p>` for paragraphs, `<ul>` for unordered lists, and `<table>` for tables.**

**Example:**



**In this example, `<h1>`, `<p>`, and `<ul>` are logical tags that convey the meaning and structure of the content.**

* **Physical Tags (Presentational Markup):** Physical tags, on the other hand, were used to define how the content should be presented visually. These tags specified the appearance or formatting of the content. **Examples of physical tags include `<font>`, `<b>` for bold text, `<i>` for italic text, and `<center>` for centering content.**

**Example:**



**In this example, `<font>`, `<b>`, `<i>`, and `<center>` are physical tags that directly control the appearance of the content.**