Introduction to HTML Elements and Tags

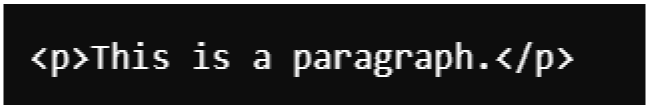
In HTML, **elements** and **tags** are the building blocks of every webpage. Understanding them is essential to create and structure content effectively.

HTML elements are the basic building blocks of a webpage, used to structure and present content such as text, images, and links.

Elements consist of:

1. Opening tag (<tag>)
2. Content (optional)
3. Closing tag (</tag>

For example:



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

Here, <p> is the start tag , This is a paragraph. is the content, and </p> is the end tag.

1) Standard HTML Elements:

An element is composed of a start tag, attributes (optional), content, and an end tag.

For example:

<h1 class="heading">Welcome to HTML</h1>

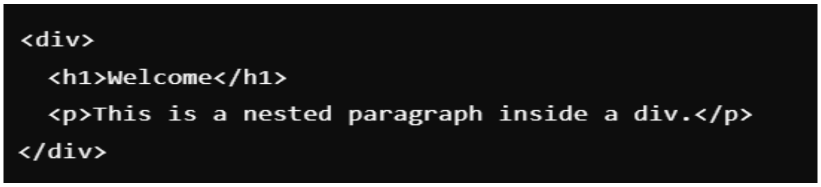
2) Void elements (empty elements):

Some elements in HTML do not have content or an end tag. These are known as void elements or unpaired tags.
Examples:
<br>: Inserts a line break.
<hr>: Inserts a horizontal line

3) Nested HTML Elements

HTML allows nesting, where one element is placed inside another. This is a common practice for structuring content.

For example:

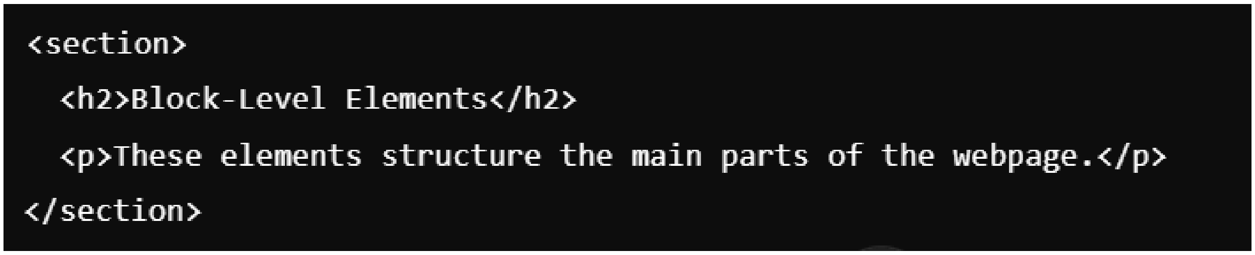


```
<div>
  <h1>Welcome</h1>
  <p>This is a nested paragraph inside a div.</p>
</div>
```

Block-level and Inline HTML elements
HTML elements are categorized into two types based on their default display behavior:

1) Block-Level Elements

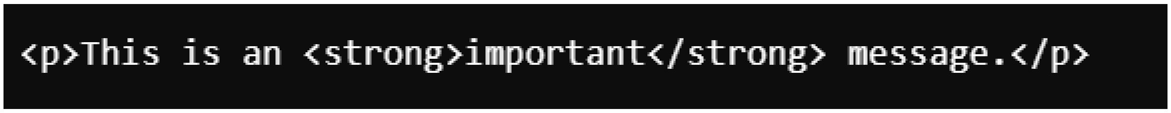
- These elements create a new block in the layout, starting on a new line and taking up the full width available (by default).
- They can contain other block-level elements as well as inline elements.
- **Examples of block-level elements:**
  <div>, <p>, <header>, <footer>, <section>, <table>, <form>, <ul>, <ol>,
  <h1> – <h6>.



```
<section>
  <h2>Block-Level Elements</h2>
  <p>These elements structure the main parts of the webpage.</p>
</section>
```

2) Inline Elements

- These elements are typically used inside block-level elements.
- They do not start on a new line and only take up as much width as required.
- **Examples of inline elements:**
  <a>, <b>, <i>, <img>, <span>, <strong>, <em>, <label>.



```
<p>This is an <strong>important</strong> message.</p>
```

Comparison of Block-Level and Inline Elements:-

Criteria	**Block-Level Elements**	**Inline Elements**
Display Behaviour	Occupy the full width available.	Take up only as much width as necessary.
Line Breaks	Always start on a new line	Do not start on a new line.
Width and Height	Can set width and height	Cannot set width and height
Containing Other Elements	Can contain both block-level and inline elements.	Can only contain other inline elements
Examples	<div> , <p> , <h1> , <ul> , <section>	<span> , <a> , <img> , <strong> , <em>

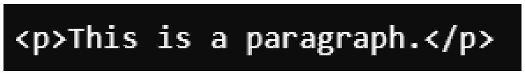
HTML Tags:-

HTML tags are keywords enclosed within angle brackets (< >) that define how web browsers should format and display content. Tags allow the browser to differentiate between HTML content and plain text, helping it render the webpage correctly.

Each tag generally consists of three parts:

1. **Opening tag**: Marks the beginning of an element..
2. **Closing tag**: Marks the end of an element (except for self-closing tags).

For example:



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

Here, <p> is the opening tag and </p> is the closing tag.

Key Points about HTML Tags:-

- All HTML tags must be enclosed within < > brackets.
- If a tag is opened (<tag>), it must generally be closed (</tag>), except for self-closing tags.
- HTML documents are read by browsers from top to bottom and left to right.

Unclosed HTML Tags:-

Some tags do not require a closing tag. These are known as **self-closing tags**.

Examples:

1) **<br> (Line Break)**:
   Inserts a line break in the content. It doesn't require a closing tag and simply forces the content that follows to start on a new line.

```
<!DOCTYPE html>

<html lang="en">

<head>

   <meta charset="UTF-8">

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

   <title>Line Break Example</title>

</head>

<body>

   <p>This is the first line.<br>This is the second line.<br>This is the third line.</p>
```

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

Output:-

```
This is the first line.
This is the second line.
This is the third line.
```

2) **<hr> (Horizontal Rule)**:

Creates a horizontal line across the webpage.

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Line Break Example</title>
</head>
<body>
   <p>This is the first line.<hr>This is the second line.</p>
</body>
```

```
</html>
```

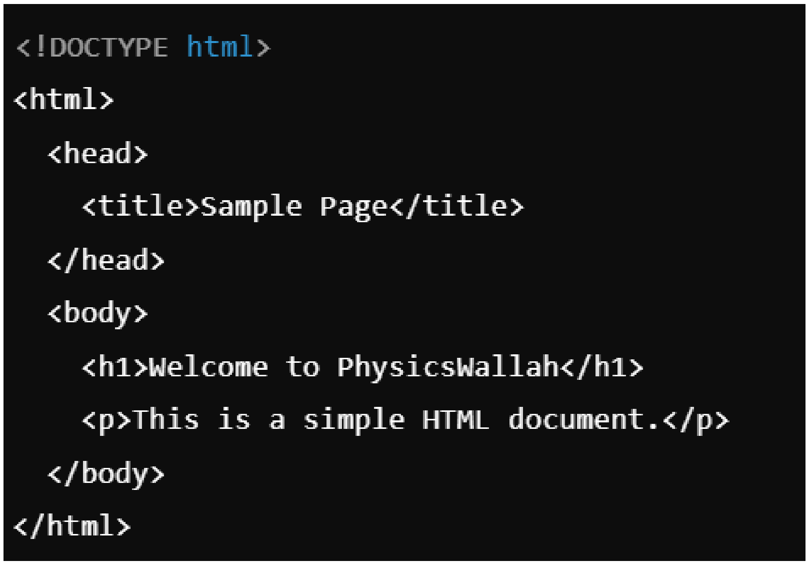
Output:-

```
This is the first line.

----------------------------

This is the second line.
```

HTML Tags:-

An HTML document must include some basic tags to define its structure and content:

Example:



```
<!DOCTYPE html>
<html>
  <head>
    <title>Sample Page</title>
  </head>
  <body>
    <h1>Welcome to PhysicsWallah</h1>
    <p>This is a simple HTML document.</p>
  </body>
</html>
```

Categories of HTML Tags

HTML tags are classified into various categories based on their functionality and usage. Below is an exploration of each category, with a brief definition of the tags and their purposes.

HTML Meta Tags:-

Used to provide metadata about the webpage, such as character encoding, author, or page description. These tags do not display content directly.

- **<!DOCTYPE>**: Declares the document type and version of HTML.
- **<title>**: Sets the title of the webpage, displayed on the browser tab.
- **<meta>**: Specifies metadata like keywords, description, and viewport settings.
- **<link>**: Links external resources like stylesheets to the document.
- **<style>**: Embeds CSS styles directly within the HTML document.

```html
<!DOCTYPE html> <!-- Declares the document type as HTML5 -->
<html lang="en"> <!-- Specifies that the language of the document is English -->

<head>
  <meta charset="UTF-8"> <!-- Specifies the character encoding for the document -->
  <meta name="viewport" content="width=device-width, initial-scale=1.0"> <!-- Ensures responsiveness -->
  <title>HTML Document</title> <!-- Sets the title of the webpage -->
</head>

<body>

  <h1>Welcome to the HTML Document Example</h1>
  <p>This page is using HTML5 as its document type, as declared by the &lt;!DOCTYPE&gt; tag.</p>

</body>

</html>

```

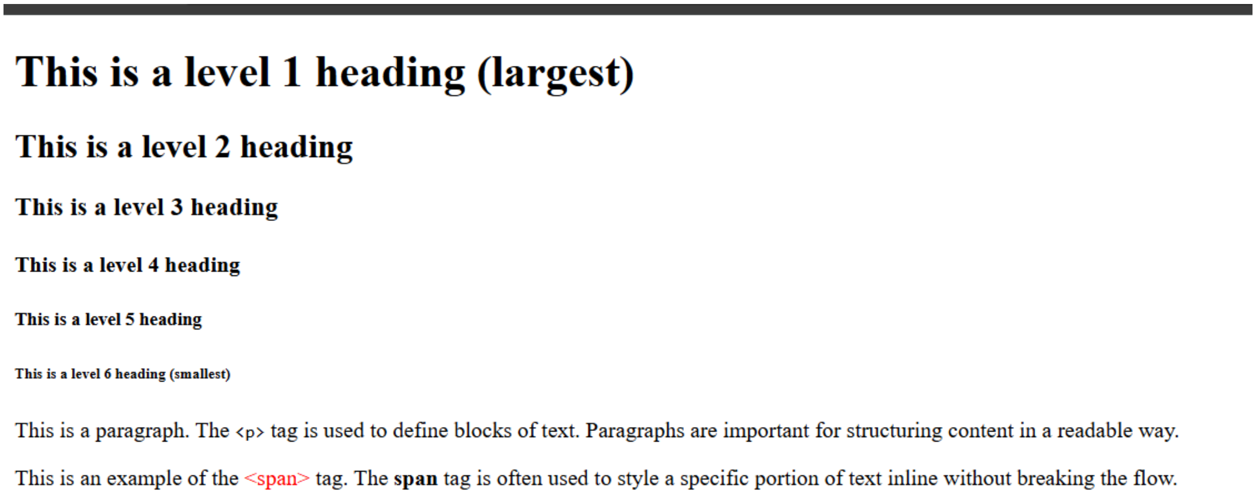
HTML Text Tags

Used to format and style text content, making it more readable and structured.

- **<p>**: Defines a paragraph.
- **<h1>–<h6>**: Represents headings, with <h1> being the largest and <h6> the smallest.

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>HTML Text Tags Example</title>
</head>
<body>

  <!-- Headings Example -->
  <h1>This is a level 1 heading (largest)</h1> <!-- Main heading -->
  <h2>This is a level 2 heading</h2> <!-- Second largest heading -->
  <h3>This is a level 3 heading</h3> <!-- Third largest heading -->
  <h4>This is a level 4 heading</h4> <!-- Smaller heading -->
  <h5>This is a level 5 heading</h5> <!-- Even smaller heading -->
  <h6>This is a level 6 heading (smallest)</h6> <!-- Smallest heading -->

  <!-- Paragraph Example -->
  <p>This is a paragraph. The <code>&lt;p&gt;</code> tag is used to define blocks of text. Paragraphs are important for structuring content in a readable way.</p>

  <!-- Span Example -->
  <p>This is an example of the <span style="color: red;">&lt;span&gt;</span> tag. The <span style="font-weight: bold;">span</span> tag is often used to style a specific portion of text inline without breaking the flow.</p>

</body>
</html>

```

Output:-



This is a level 1 heading (largest)

This is a level 2 heading

This is a level 3 heading

This is a level 4 heading

This is a level 5 heading

This is a level 6 heading (smallest)

This is a paragraph. The <p> tag is used to define blocks of text. Paragraphs are important for structuring content in a readable way.

This is an example of the <span> tag. The **span** tag is often used to style a specific portion of text inline without breaking the flow.

HTML Link Tags

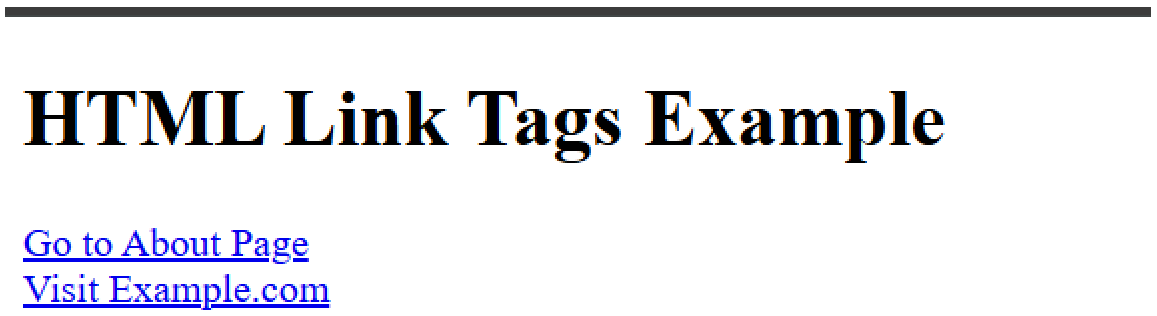
Used to create hyperlinks or define base URLs for relative links.

- **<a>**: Creates a hyperlink to navigate to another page or resource.
- **<base>**: Sets the base URL for all relative links in the document.

```
<!DOCTYPE html>

<html lang="en">

<head>

 <meta charset="UTF-8">

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

```
  <title>HTML Link Tags Example</title>


  <!-- Setting the base URL for relative links -->

  <base href="https://www.example.com/" target="_blank">

</head>

<body>


  <h1>HTML Link Tags Example</h1>


  <!-- Hyperlink to an external webpage -->

  <a href="about.html">Go to About Page</a> <!-- This will navigate to https://www.example.com/about.html -->


  <br>


  <!-- Hyperlink to an external website -->

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


</body>

</html>
```

Output:-



HTML Link Tags Example

Go to About Page
Visit Example.com

Syntax for Using HTML Tags

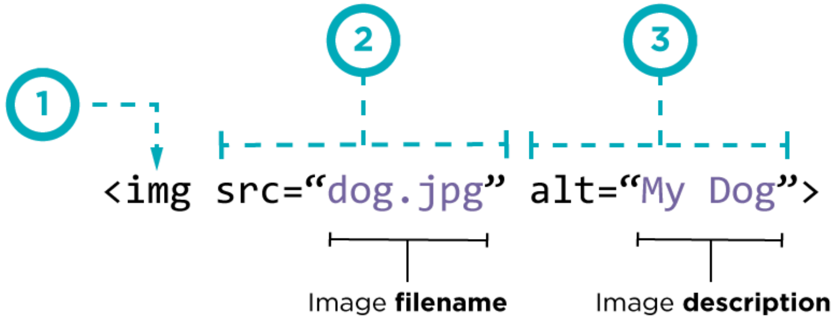
<tagname> Content goes here. </tagname>

Example:

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

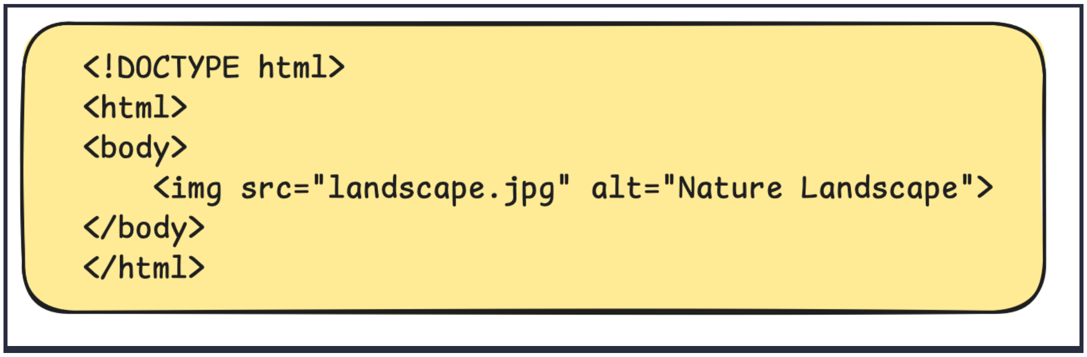
HTML Image Tag and its Attributes

Integrating images into web pages is fundamental for enhancing visual appeal and user engagement. In HTML, the <tag> serves as the conduit for incorporating images. Let’s delve into the nuances of working with HTML images and explore additional features like image maps.



HTML Images Syntax

The basic syntax for embedding an image is straightforward:



```
<!DOCTYPE html>
<html>
<body>
    <img src="landscape.jpg" alt="Nature Landscape">
</body>
</html>
```

Here, the **<img>** tag is an empty element, necessitating two crucial attributes: **src** (specifying the image path) and **alt** (providing alternate text).

The Src Attribute

```
<img src="cityscape.jpg" alt="Urban Cityscape">
```

The **src** attribute pinpoints the image's location, defining the source.

The Style Attribute

Adjusting image dimensions can be achieved through the **style** attribute:

```
<img src="sunset.jpg" alt="Breathtaking Sunset" style="width:800px;height:400px;">
```

Alternatively, the **width** and **height** attributes can be used:

```
<img src="sunset.jpg" alt="Breathtaking Sunset" width="800" height="400">
```

Specifying dimensions is crucial to prevent page flickering during image loading.

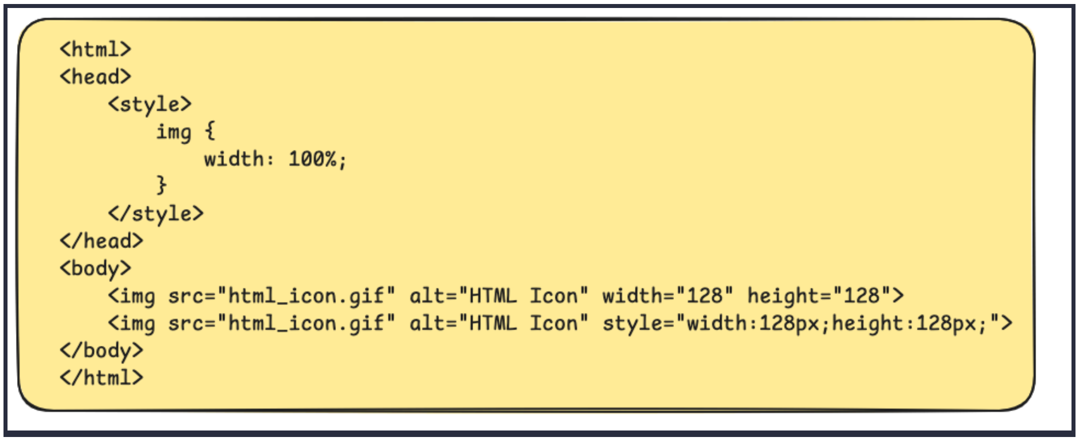
Width and Height, or Style?

Both approaches are valid; however, the **style** attribute is recommended to prevent stylesheets from altering image sizes.

```
<!DOCTYPE html>
<img src="floral.jpg" alt="Beautiful Floral Arrangement">
```

The alt attribute provides text for images, which helps with accessibility and appears if the image doesn't load.

Image Size – Width and Height



```
<html>
<head>
    <style>
        img {
            width: 100%;
        }
    </style>
</head>
<body>
    <img src="html_icon.gif" alt="HTML Icon" width="128" height="128">
    <img src="html_icon.gif" alt="HTML Icon" style="width:128px;height:128px;">
</body>
</html>
```

Images in Another Folder

Include the folder name for images in a sub-folder:

```
<img src="/images/scenery.jpg" alt="Scenic View" style="width:500px;height:300px;">
```

Images on Another Server/Website

For images on external servers, use an absolute URL:

```
<img src="<https://www.example.com/images/beach.jpg>" alt="Sunny Beach">
```

Exercise caution with copyright and potential changes to external images.

Animated Images

Animated GIFs are supported in HTML:

```
<img src="animated_logo.gif" alt="Animated Logo" style="width:64px;height:64px;">
```

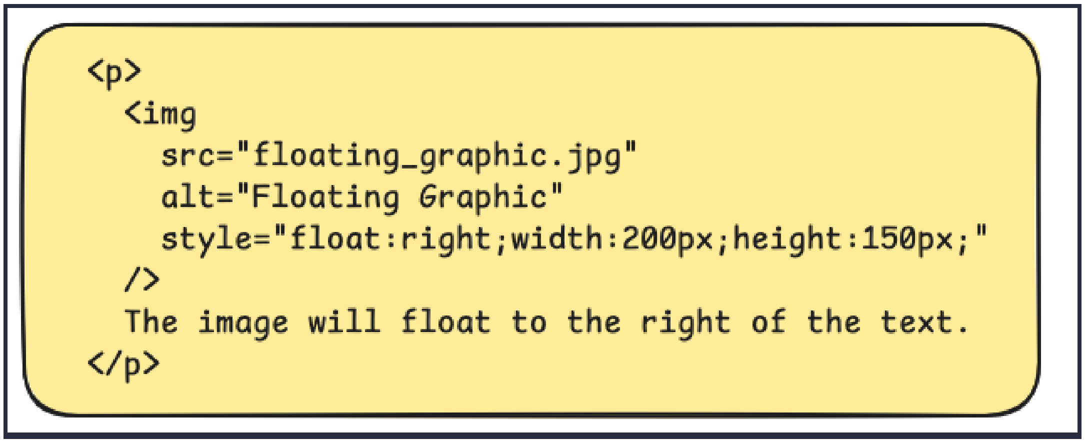
Image as a Link

Embed an image within an **<a>** tag to create a clickable link:

```
<a href="destination.html">
   <img src="button_graphic.jpg" alt="Click Me">
</a>
```

Image Floating:

Utilize the CSS **float** property for image alignment:



```
<p>
  <img
    src="floating_graphic.jpg"
    alt="Floating Graphic"
    style="float:right;width:200px;height:150px;"
  />
  The image will float to the right of the text.
</p>
```

Common HTML Image Formats

Supported image formats include APNG, GIF, ICO, JPEG, PNG, and SVG.

Canvas Tag and its Attributes

Use the **HTML <canvas> element** with either the canvas scripting API or the WebGL API to draw graphics and animations.

Attributes

This element's attributes include the global attributes.

height

The height of the coordinate space in CSS pixels. Defaults to 150.

moz-opaque

Let the canvas know whether translucency will be a factor. If the canvas knows there's no translucency, painting performance can be optimized. This is only supported by Mozilla-based browsers; use the standardized canvas.getContext('2d', { alpha: false }) instead.

width

The width of the coordinate space in CSS pixels. Defaults to 300.

What is a Grouping Tag?

A **grouping tag** in HTML is an element that is used to group multiple other elements together, allowing for better organization, easier styling, and manipulation of those elements.

Popular Grouping Tags

1. **<div>**: The <div> tag is one of the most commonly used grouping tags. It is a block-level element that can wrap any other HTML elements, allowing for flexible structuring of content.
2. **<section>**: Defines sections of content that are thematically related, often used for organizing different parts of the webpage (like articles or features).
3. **<article>**: Represents independent, self-contained content, often used for blog posts or news articles.
4. **<header>**: Used to define introductory content, typically the heading, navigation, or logo of a page.
5. **<footer>**: Defines the footer section of a webpage, typically containing information like copyright, contact details, or links to privacy policies.
6. **<nav>**: Defines navigation links on a webpage.

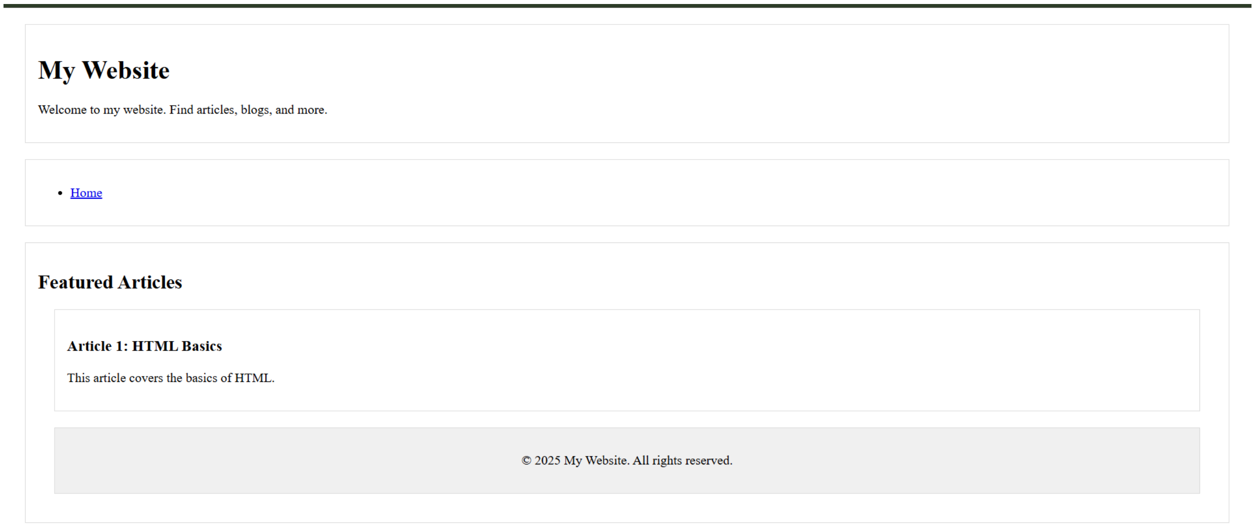
Example:-

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Grouping Tags Example</title>
  <style>
    /* Simple styles for each section */
    header, footer, nav, section, article {
      margin: 20px;
      padding: 15px;
      border: 1px solid #ddd;
    }
    footer {
      background-color: #f1f1f1;
      text-align: center;
    }
  </style>
</head>
<body>

  <!-- Header section -->
  <header>
```

```html
  <h1>My Website</h1>

  <p>Welcome to my website. Find articles, blogs, and more.</p>

</header>



<!-- Navigation section -->

<nav>

  <ul>

    <li><a href="#home">Home</a></li>

  </ul>

</nav>



<!-- Main content section -->

<section>

  <h2>Featured Articles</h2>



  <!-- Article 1 -->

  <article>

    <h3>Article 1: HTML Basics</h3>

    <p>This article covers the basics of HTML.</p>

  </article>
  <!-- Footer section -->
  <footer>
    <p>&copy; 2025 My Website. All rights reserved.</p>
  </footer>


</body>
</html>
```

Output:-



My Website

Welcome to my website. Find articles, blogs, and more.

- Home

Featured Articles

Article 1: HTML Basics

This article covers the basics of HTML.

© 2025 My Website. All rights reserved.

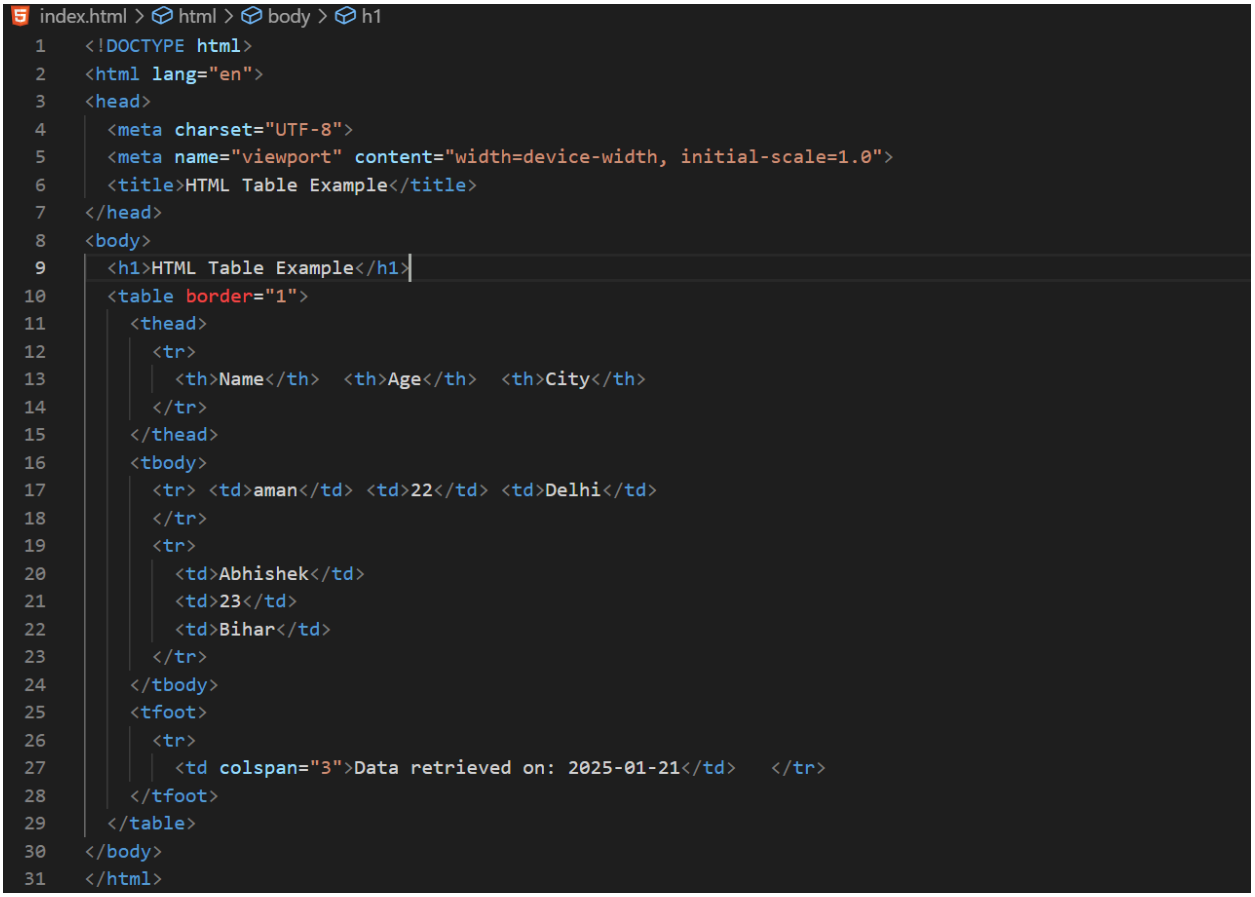
HTML Tables

Tables in HTML are used to display data in a structured format with rows and columns. Let's explore how to create and work with tables step-by-step.

Let's Take an Example

Follow these steps to practice creating a table in HTML:

1. **Open a Code Editor**
   Use a text editor like Visual Studio Code, Sublime Text, or Notepad.
2. **Create a New File**
   Save the file with the name html_table_example.html.
3. **Copy and Paste the Code Below**



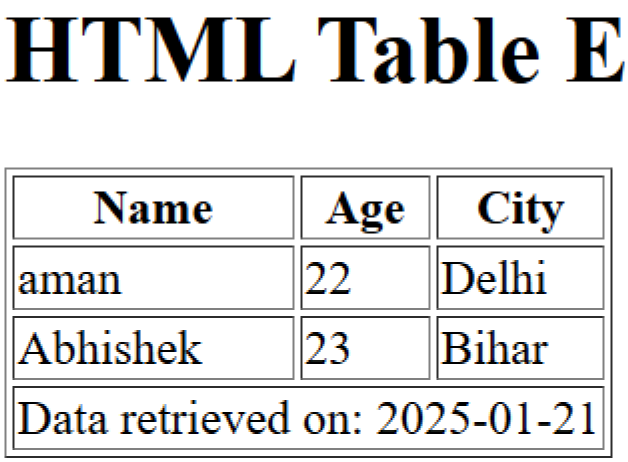
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>HTML Table Example</title>
</head>
<body>
  <h1>HTML Table Example</h1>
  <table border="1">
    <thead>
      <tr>
        <th>Name</th>  <th>Age</th>  <th>City</th>
      </tr>
    </thead>
    <tbody>
      <tr> <td>aman</td> <td>22</td> <td>Delhi</td>
      </tr>
      <tr>
        <td>Abhishek</td>
        <td>23</td>
        <td>Bihar</td>
      </tr>
    </tbody>
    <tfoot>
      <tr>
        <td colspan="3">Data retrieved on: 2025-01-21</td>   </tr>
    </tfoot>
  </table>
</body>
</html>
```

Open the File in a Browser

- Double-click the saved html_table_example.html file or open it in your preferred browser.

Observe the Table

- You will see a table with headers, rows, and a footer.



HTML Table Example

Name	Age	City
aman	22	Delhi
Abhishek	23	Bihar
Data retrieved on: 2025-01-21		

Table Structure

- <table>: Creates the table.
- border="1": Adds a border to the table (for visibility).

Table Header

- <thead>: Contains the table’s header rows.
- <tr>: Represents a table row.
- <th>: Defines header cells, which are bold and centered by default.

```
<thead>
  <tr>
    <th>Name</th>
    <th>Age</th>
    <th>City</th>
  </tr>
```

```
</thead>
```

Table Body

- <tbody>: Contains the main content rows of the table.
- <td>: Defines table data cells.

```
<tbody>
  <tr>
    <td>Aman</td>
    <td>22</td>
    <td>Delhi</td>
  </tr>
</tbody>
```

Table Footer

- **<tfoot>: Contains footer information, often summary or additional notes.**
- **colspan: Combines multiple columns into one cell.**

```
<tfoot>
  <tr>
    <td colspan="3">Data retrieved on: 2025-01-21</td>
  </tr>
</tfoot>
```

Key Attributes of Table Elements:-

1. **colspan:**
   - Combines multiple columns into one cell.
   - Example: <td colspan="3">Merged 3 Cell</td>
2. **rowspan:**
   - Combines multiple rows into one cell.
   - Example: <td rowspan="2">Merged 2 Rows</td>

Let’s explore how to create and work with tables step-by-step.

Step 1: Open Your Code Editor

Use a text editor like Visual Studio Code, Sublime Text, or Notepad++.

Step 2: Create a New File

Step 3: Copy and Paste the Code Below

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>HTML Table with Colspan and Rowspan</title>
</head>
<body>
  <h1>Table with Colspan and Rowspan</h1>
  <table border="1" style="width: 100%; border-collapse: collapse;">
    <!-- Table Header -->
    <thead>
      <tr>
        <th>Category</th>
        <th>Product</th>
        <th>Details</th>
      </tr>
    </thead>
    <!-- Table Body -->
    <tbody>
      <tr>
        <td rowspan="2">Electronics</td> <!-- Merges 2 rows under "Electronics" -->
        <td>Laptop</td>
        <td>Intel i7, 16GB RAM</td>
      </tr>
      <tr>
```

```
        <td>Smartphone</td>
        <td>128GB, 5G Support</td>
      </tr>
      <tr>
        <td colspan="3" style="text-align: center;">End of Data</td> <!-- Spans 3 columns -->
      </tr>
    </tbody>
  </table>
</body>
</html>
```

Output:-

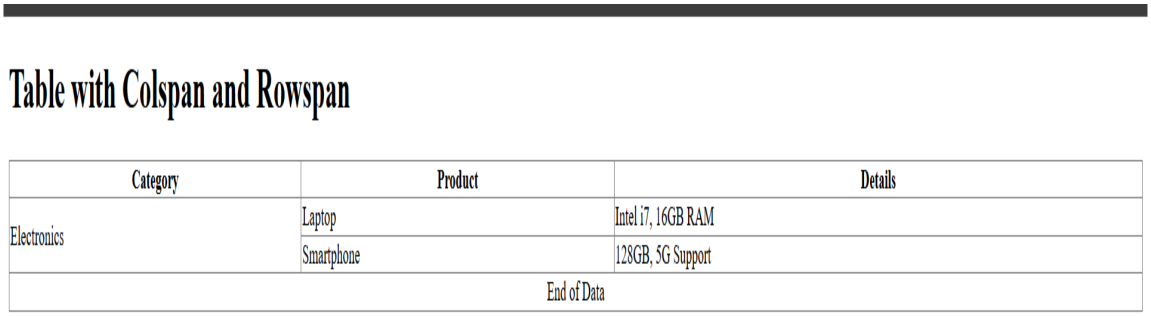


Table with Colspan and Rowspan

Category	Product	Details
Electronics	Laptop	Intel i7, 16GB RAM
	Smartphone	128GB, 5G Support
End of Data		

Observe the Table:-

1. **Rowspan Example:**
    - The "Electronics" cell spans across two rows, grouping the category with its products.
2. **Colspan Example:**
    - The "End of Data" cell spans all three columns, creating a visually clean footer.

Key Points:-

- <tr> defines the structure (rows), not the cells.
- <td> and <th> define the content and allow for merging cells.
- By using colspan and rowspan wisely in <td> or <th>, you can structure your tables in a more meaningful and visually appealing way