MODULE-1

Topics:

WEB BROWSERS, WEB SERVERS, MIME, URL HTTP

Introduction to HTML tags, Basic syntax and structure, Text Markups - paragraphs, line breaks, white spaces, headings, block quotations, font style and size, character entity. Images, Lists, Tables, Links, Frames, meta tag, span and div tags.

Q 1: Explain Web browsers. (OR) Write a note on Web browsers.

Ans: A **web browser** is a software application that allows users to access and navigate the internet. It interprets and displays web pages, which are written in HTML, CSS, and JavaScript. Browsers help users view content like text, images, videos, and interactive elements that make up websites.

Key Functions of a Web Browser:

1. **Rendering Web Pages**: A browser translates the HTML, CSS, and JavaScript of a website into a visual representation (the webpage) that users can interact with.
2. **Navigation**: Browsers let users navigate websites by entering a web address (URL) in the address bar, clicking on links, or using bookmarks.
3. **Managing Cookies and Sessions**: Browsers store cookies and session data that help websites remember users' preferences, login states, and other information across visits.
4. **Security**: Modern browsers help protect users from malicious websites, phishing attempts, and other online threats by using features like HTTPS (encrypted communication) and warning systems.
5. **Extensions/Add-ons**: Browsers allow users to install extra features or enhancements, such as ad blockers, password managers, or privacy tools.

Examples of the Popular Web Browsers:

1. **Google Chrome**: Known for its speed and extensive library of extensions.
2. **Mozilla Firefox**: Focuses on privacy and open-source development.
3. **Microsoft Edge**: Built on Chromium (the same engine as Chrome) but integrated with Microsoft services.
4. **Safari**: Apple's browser, optimized for macOS and iOS devices.
5. **Opera**: Known for its built-in VPN, ad blocker, and other features.
6. **Brave**: A privacy-centric browser that blocks ads and trackers by default.

Components of a Browser:

- **Address Bar**: Where users type the website URL.
- **Tabs**: Allows multiple web pages to be opened in the same window.
- **Bookmark Bar**: Quick access to frequently visited websites.
- **Back/Forward Buttons**: Let you navigate to the previous or next page.
- **Refresh/Reload Button**: Refreshes the current page.
- **Developer Tools**: For advanced users and developers to debug websites.

In summary, web browsers are essential tools for interacting with the internet, enabling users to access websites and digital content across different devices and platforms.

Q 2: Explain Web Servers. (OR) Write a note on Web servers

Ans: A **web server** is a system or software application that stores, processes, and serves content (like web pages, images, videos, or other media) to users over the internet. When you access a website using a browser, the browser sends a request to the web server, and the server responds by delivering the requested web content.

Following are the key functions of a web server:

1. **Receiving Requests**: When you type a URL into a web browser (e.g., https://www.example.com), the browser sends an HTTP or HTTPS request to the server that hosts the website.
2. **Processing Requests**: The web server processes the incoming request, which could involve retrieving a static file (like an HTML file) or executing server-side scripts (like PHP, Python, or Node.js) to generate dynamic content.
3. **Serving Content**: Once the web server processes the request, it sends the requested content back to the browser as an HTTP response. This content can be HTML, CSS, JavaScript, images, videos, or other types of files.
4. **Handling Static vs. Dynamic Content**:
    - **Static Content**: Files like HTML, CSS, and image files are served directly from the server's storage without any modification.
    - **Dynamic Content**: For more complex sites, the server may run code (like PHP, Python, or JavaScript on the server-side) to generate web pages dynamically based on user input, database queries, or other factors.

Common Web Server Software

- **Apache HTTP Server**: One of the most widely used web servers, known for its flexibility and extensive configuration options.
- **Nginx**: A high-performance web server, often used for its speed and efficiency, especially when handling large volumes of traffic.
- **Microsoft Internet Information Services (IIS)**: A web server for Windows-based systems.
- **LiteSpeed**: A commercial web server that focuses on speed and security.

Web Server vs. Web Browser

- **Web Server**: A machine or software that hosts and delivers web content.
- **Web Browser**: A program on your device (like Google Chrome, Firefox, or Safari) that displays the content received from the web server.

In simple terms, the web server is the "host" that provides the content, and the web browser is the "client" that requests and views it.

Q 3: What is MIME? Explain.

MIME (Multipurpose Internet Mail Extensions) is a standard that extends the functionality of email by allowing the transfer of various types of data beyond plain text. Originally designed to address the limitations of traditional email, MIME enables the inclusion of multimedia content such as images, audio, video, and application files in email messages.

Key Features of MIME:

1. **Content Type Specification**: MIME defines a mechanism to specify the type of data being sent, such as text/plain, image/jpeg, or application/pdf.
2. **Encoding**: It includes methods like Base64 to encode binary data into a format suitable for transmission over text-based protocols like SMTP.
3. **Multipart Messages**: MIME allows messages to contain multiple parts, enabling the combination of text, attachments, and other data in a single email.
4. **Headers**: MIME introduces additional email headers, such as Content-Type and Content-Disposition, to describe the nature and handling of the content.

MIME is widely used not only in email systems but also in web applications to define content types for HTTP communication.

Q 4: Describe uniform resource locator (URL).

It is a standardized address used to locate resources on the internet. It serves as the foundation for navigating and accessing web content, whether it's a webpage, an image, a file, or a service. A URL specifies the location of a resource and the protocol used to access it.

Structure of a URL

A URL typically consists of several components:

1. **Protocol (Scheme)**: Defines how the resource is accessed. Common examples are:
   - http: HyperText Transfer Protocol
   - https: Secure version of HTTP
   - ftp: File Transfer Protocol

   Example: https://

2. **Domain (Host)**: Indicates the server hosting the resource. Example: www.example.com
3. **Port (Optional)**: Specifies the communication port (default for HTTP is 80 and HTTPS is 443). Example: :8080
4. **Path**: Points to the specific resource on the server. Example: /path/to/resource
5. **Query Parameters (Optional)**: Provides additional data in the form of key-value pairs, often used for dynamic content. Example: ?key1=value1&key2=value2
6. **Fragment (Optional)**: Refers to a specific section of the resource. Example: #section1

Example URL

https://www.example.com:8080/path/to/resource?key=value#section

- **Protocol**: https
- **Domain**: www.example.com
- **Port**: 8080
- **Path**: /path/to/resource
- **Query Parameters**: key=value
- **Fragment**: section

Importance of URLs

- URLs are essential for navigating the web.

- They allow users to directly access specific resources.
- Search engines rely on URLs to index and retrieve web pages.
- URLs ensure that resources can be uniquely identified and accessed globally.

Understanding the structure of URLs is critical for web development, search engine optimization (SEO), and troubleshooting network issues.

Q 5: Explain Hypertext Transfer Protocol (HTTP)

It is a protocol used for transferring data over the web. It is the foundation of data communication on the World Wide Web. HTTP defines how messages are formatted and transmitted and how web servers and browsers should respond to various commands.

Key Features of HTTP:

1. **Stateless**: HTTP is a stateless protocol, meaning each request from a client to a server is independent. The server does not retain any memory of previous requests.
2. **Client-Server Architecture**: HTTP operates on a client-server model where the client (usually a browser) sends a request, and the server processes it and sends back a response.
3. **Request-Response Mechanism**:
   - A **request** is sent by the client, specifying a method (e.g., GET, POST) and a resource (e.g., a webpage or file).
   - The **response** is sent by the server, containing the requested resource or an error code.
4. **Methods**:
   - **GET**: Retrieve data from a server.
   - **POST**: Submit data to a server.
   - **PUT**: Update a resource.
   - **DELETE**: Remove a resource.
   - Other methods include HEAD, OPTIONS, PATCH, etc.
5. **HTTP Versions**:
   - **HTTP/1.1**: The most widely used version, supporting persistent connections and other features.
   - **HTTP/2**: Introduced better performance with multiplexing and reduced latency.
   - **HTTP/3**: Uses QUIC protocol for even faster and more secure data transfer.
6. **URLs and Resources**: HTTP is closely tied to URLs (Uniform Resource Locators), which specify the address of the resource being accessed.

How It Works:

1. A user enters a URL in their browser.
2. The browser sends an HTTP request to the server hosting the website.
3. The server processes the request and sends back an HTTP response containing the requested resource or an error message.
4. The browser displays the content to the user.

HTTP is a key component of how we access and interact with the web. Secure communication is often facilitated through HTTPS (HTTP Secure), which encrypts HTTP data using protocols like TLS (Transport Layer Security).

INTRODUCTION TO HTML TAGS

HTML tags are the fundamental building blocks of a webpage, used to define and structure content within an HTML document. They are enclosed in angle brackets (< >) and usually come in pairs: an opening tag and a closing tag, though some tags are self-closing.

Structure of HTML Tags

1. **Opening Tag**: Indicates the beginning of an element. Example: <p> (for a paragraph).
2. **Closing Tag**: Indicates the end of an element. Example: </p>.
3. **Self-Closing Tags**: Some tags don't need a closing counterpart. Example: <img /> (for images).

Basic Examples

1. **Heading Tags**: <h1> to <h6> (from largest to smallest headings):
   <h1>This is a heading</h1>

2. **Paragraph Tag**: <p>:
   <p>This is a paragraph.</p>

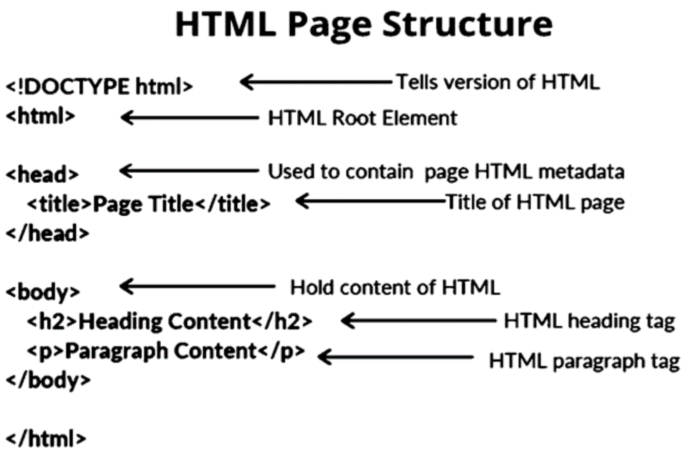
3. **Anchor Tag**: <a> for links:
   <a href="https://example.com">Visit Example</a>

4. **Image Tag**: <img> for images (self-closing):
   <img src="image.jpg" alt="Description of image" />

Q 6: What is HTML? Explain the basic syntax and structure of a HTML program.

HTML (HyperText Markup Language) is the standard markup language used to create web pages. It describes the structure of a web page using a series of elements and tags. Here is a basic overview of the structure of an HTML document:

An HTML document is divided into two main parts: the **head** and the **body**.



HTML Page Structure

```
<!DOCTYPE html>
<html>

<head>
  <title>Page Title</title>
</head>

<body>
  <h2>Heading Content</h2>
  <p>Paragraph Content</p>
</body>

</html>
```

- The <!DOCTYPE html> declaration defines that this document is an HTML5 document
- The <html> element is the root element of an HTML page

- The <head> element contains meta information about the HTML page
- The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
- The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.

Q 7: The PARAGRAPH TAG <p>

A **paragraph in HTML** is simply a block of text enclosed within the **<p> tag**. The <p> tag helps divide content into manageable, readable sections. It's the go-to element for wrapping text in a web page that is meant to be displayed as a distinct paragraph.

Syntax:
<p> Content**</p>**

How HTML Paragraphs are rendered?
In HTML, when you wrap text inside the <p> tag, it automatically:

- Adds space before and after the paragraph to visually separate it from other content.
- Breaks the text into a single block, creating an easy-to-read section.

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>The p tag</title>
</head>
<body>
  <p>A Computer Science portal for geeks.</p>
  <p>It contains well written, well thought articles.</p>
</body>
</html>
```

Q 8: Describe LINE BREAKS in HTML.

The **<br> tag** is used to create a line break in HTML. It allows you to break text into a new line without starting a new paragraph or block. This is helpful when you need to format text in a specific way, like in poems or addresses. For adding a line break in HTML, simply insert the <br> tag where the text should continue on the next line.

How to Use the <br> Tag in HTML?
Using the <br> tag is simple and effective method. To insert a line break, simply place the **<br>** tag at the point in your content where you want the text to start on a new line.
Syntax
<br>
Note: *No closing tag is required i.e. the <br> tag is self-closing.*
Example

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

Output

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

Q 9: Write a note on WHITE SPACES in HTML.

Inserting spaces in HTML involves using the non-breaking space character (`&nbsp;`). It ensures spaces are displayed and not collapsed by browsers. Multiple `&nbsp;` can create fixed-width spaces. CSS margin and padding properties are preferred for layout spacing over excessive use of `&nbsp;`.

Using Non-Breaking Space
The abbreviation of ***&nbsp;*** is a Non-Breaking Space entity used in HTML to insert a space between characters that prevents line breaks.
Syntax:
&nbps;

Using Multiple Non-Breaking Spaces
We can achieve multiple non-breaking spaces in different ways including &nbsp;, &emsp;, and &ensp;.
Syntax:
&ensp; Or $emsp;
Example

```
<h1>
      Welcome
      to&ensp;To BITM
   </h1>
<h3>Hello&emsp;BITM</h3>
```

Q 10: Explain different levels of heading tags in HTML with an example.

HTML Headings are used to **define the content hierarchy and structure of a webpage**. They range from **<h1> to <h6>**, with <h1> being the most important heading and <h6> the least important. Proper use of headings helps improve readability, Search Engine Optimization and accessibility.

Levels of HTML Heading Tags
HTML offers **six levels of heading tags**, each serving a different purpose in structuring your content:
<h1> – Main Heading (Largest)
- Represents the primary focus of the page, usually used for the main title.
- Use only one <h1> tag per page for the best SEO practices.
- Makes it clear to both users and search engines what the main topic is.

<h2> – Subheadings
- Ideal for dividing the content into major sections.
- If the content has further subsections, use <h3> to create a logical flow.

<h3> to <h6> – Smaller Headings
- These heading levels are used for finer subdivisions, gradually decreasing in size and importance.
- <h3> is used for subsections under <h2>, while <h4> to <h6> are used for additional, less important subdivisions.
- <h6> defines the least important heading.

  Example

  ```
  <html>
  ```

```
<body>
   <h1>This is the Main Heading</h1>
   <h2>This is a Subheading</h2>
   <h3>This is a Smaller Subheading</h3>
   <h4>This is a Sub-Subheading</h4>
   <h5>This is a Minor Subheading</h5>
   <h6>This is the Smallest Heading</h6>
</body>
</html>
```

Q 11: Explain the <blockquote> tag in HTML with an example.

The **<blockquote>** HTML element indicates that the enclosed text is an extended quotation. Usually, this is rendered visually by indentation. A URL for the source of the quotation may be given using the cite attribute, while a text representation of the source can be given using the <cite> element.

Example

```
<body>
   <h2>Example of blockquote tag</h2>
   <p>A Great Motivational Quote :</p>
  <blockquote cite="https://www.myquotes.com/quotes1">
   <p>
    At the end of the day, you are solely responsible for your success and your failure. And the sooner you realize that, you accept that, and integrate that into your work ethic, you will start being successful. As long as you blame others for the reason you aren't where you want to be, you will always be a failure.
    </p>
  </blockquote>
<cite>Erin Cummings</cite>
</body>
```

Q 12: Write a note on Font style and font size in HTML.

<u>Font Style in HTML</u>

To style fonts in HTML, you can use the CSS properties within the `style` attribute or in a separate CSS file. This allows you to change the font family, size, weight, style, and color of your text.

Example: Changing Font Family and Size

```
<html>
<head>
<style>
.example {
font-family: "Arial", sans-serif;
font-size: 20px;
}
</style>
</head>
<body>
```

```
<p class="example">This is an example of styled text.</p>
</body>
</html>
```

Font size in HTML

To change the font size in HTML, you can use CSS's `font-size` property. This can be done through inline, internal, or external styling.

Example: Inline CSS

Inline CSS allows you to apply styles directly within HTML tags using the `style` attribute.

```
<h1 style="font-size: 4em;">Hello World!</h1>

<p style="font-size: 14px;">Any text whose font we want to change</p>
```

Q 13: Describe Character entities in HTML.

HTML character entities are basically a set of characters (entity) used to represent few characters reserved by the HTML, especially invisible characters or characters difficult to type out using a regular keyboard. HTML provides some entity names and entity numbers to use these symbols.

Example: '<' is already reserved in HTML language. Sometimes this character needs to display on the web page that creates ambiguity in code. Along with these are the character which is normally not present in basic keyboard (£, ¥, €, ©), etc.

Syntax:
&entity_name; (or) &#entity_number;

Symbol	*Entity Name*	*Entity Number*	*Details*
	&	&#160;	non-breaking space
<	&lt;	&#60;	less than
>	&gt;	&#62;	greater than
®	&reg;	&#174;	registered trademark
©	&copy;	&#169;	copyright
₹		&#8377;	India rupee sign
€	&euro;	&#8364;	euro
¥	&yen;	&#165;	yen
$	&dollar;	&#36;	dollar

Symbol	*Entity Name*	*Entity Number*	*Details*
£		&#163;	pound
¢	&cent;	&#162;	cent
‘	&apos;	&#39;	single quotation mark (apostrophe)
“	&quot;	&#34;	double quotation mark
&	&amp;	&#38;	ampersand

Reserved Characters: Reserved characters are those characters that are either reserved for HTML or those which are not present in the basic keyboard.
Example: As Entity numbers are easy to learn. Here, a few of the reserved characters are discussed:

Reserved Character	**Entity Name**
&	&amp;
<	&lt;
>	&gt;
“	&quot;

Example

```
<!DOCTYPE html>
<html>
 <head>
  <title>HTML Character entities</title>
 </head>
 <body>
  <h3>Indian rupee sign = ₹</h3>
  <h3>Euro sign = €</h3>
  <h3>Dollar sign = $</h3>
  <h3>Pound sign = £</h3>
 </body>
</html>
```

Q 14: Explain <img> tag with an example.

The **HTML <img> tag** is used to embed an image in web pages by linking them. It creates a placeholder for the image, defined by attributes like src, width, height, and alt, and does not require a closing tag.
Syntax:

```
<img src= "url">
```

In the above syntax, <img> is tag src is used to define the specify the image file's location and Url includes the address of image. The url may be relative or absolute. **For example,** if you want to insert an image file on a webpage residing in the same folder/directory on your page then only image filename is needed to be specified as follows:

```
<h2>HTML Image Example</h2>
<img src="good_morning.jpg" alt="Good Morning Friends"/>
```

Some of the attributes of <img> tag are explained below:

src

It is a necessary attribute that describes the source or path of the image. It instructs the browser where to look for the image on the server. The location of image may be on the same directory or another server.

alt

The alt attribute defines an alternate text for the image, if it can't be displayed. The value of the alt attribute describe the image in words.

```
<img src="animal.png" height="180" width="300" alt="animal image">
```

Height and width

If we want to display image according to our requirement, then we can set height and width attributes of image. Both of these attributes are optional. If you do not specify these attributes, the browser will show an image width its original size. You can set the value of height and width attribute in pixels or in percentage ranging from 1 to 100.

```
<img src="animal.jpg" height="180" width="300" alt="animal image">
```

hspace

The hspace attribute is used to inserts an equal space on the left and right side of an image.

vspace

The vspace attribute is used to inserts an equal space on the top and bottom of an image.

```
<html>

<body>

  <div>

    <img src="image123.png" alt="Demo_image" width="100" height="106">

  </div>

</body>

</html>
```

Q 15: Different types of Lists in HTML

Lists in HTML

HTML lists are used to group related items in a structured manner. They help in organizing content on web pages, making it more readable and accessible. There are three main types of lists in HTML: unordered lists, ordered lists, and description lists.

Unordered Lists

Unordered lists are used to display items where the order does not matter. They are created using the <ul> tag, and each item within the list is defined using the <li> tag. By default, the list items are marked with bullets (small black circles).

Example:

```
<ul>
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ul>
```

In this example, the list items "Coffee", "Tea", and "Milk" are displayed with bullet points.

Ordered Lists

Ordered lists are used to display items where the order does matter. They are created using the <ol> tag, and each item within the list is defined using the <li> tag. By default, the list items are marked with numbers.

Example:

```
<ol>
<li>First item</li>
<li>Second item</li>
<li>Third item</li>
</ol>
```

In this example, the list items "First item", "Second item", and "Third item" are displayed with numbers

Description Lists

Description lists are used to display a list of terms and their descriptions. They are created using the <dl> tag. Each term is defined using the <dt> tag, and each description is defined using the <dd> tag.

Example:

```
<dl>
<dt>Coffee</dt>
<dd>- black hot drink</dd>
<dt>Milk</dt>
<dd>- white cold drink</dd>
</dl>
```

In this example, the terms "Coffee" and "Milk" are described with their respective descriptions

Attributes for Lists

Unordered Lists (<ul>): compact: Renders the list smaller (not supported in HTML5). type: Specifies the type of marker (e.g., disc, circle, square).

Ordered Lists (<ol>): compact: Defines the list should be compacted (not supported in HTML5). reversed: Defines that the order will be descending. start: Defines the starting number or letter. type: Defines the type of numbering (e.g., 1, A, a, I, i).

Example with Attributes:

```
<ol type="i" start="5" reversed>
<li>HTML</li>
```

<li>CSS</li>
<li>JavaScript</li>
</ol>

In this example, the ordered list starts from 5, uses Roman numerals, and is displayed in descending order.

HTML lists provide a way to display a structured collection of items, making the content more organized and readable. They are essential for creating well-structured web pages.

Q 16: Explain Table creation in HTML.

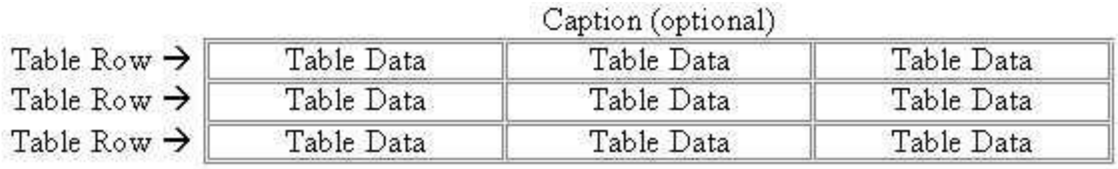
HTML table tag is used to display data in tabular form (row * column). There can be many columns in a row. We can create a table to display data in tabular form, using <table> element, with the help of <tr> , <td>, and <th> elements.

In each table, table row is defined by <tr> tag, table header is defined by <th>, and table data is defined by <td> tags.

HTML tables are used to manage the layout of the page e.g. header section, navigation bar, body content, footer section etc. But it is recommended to use div tag over table to manage the layout of the page.

Parts of a Table

The table below is a generic table with its parts labelled.



Caption (optional)

Table Row →	Table Data	Table Data	Table Data
Table Row →	Table Data	Table Data	Table Data
Table Row →	Table Data	Table Data	Table Data

Attributes Of An HTML Table Tag

Attribute	Description	Syntax
border	Used for setting the width of the border around the table.	**<table border=”1″>** creates a table with borders.
cellpadding	Sets the desired amount of space between the cell content and its border.	**<table cellpadding=”5″>** adds space inside cells.
cellspacing	Sets the amount of space between individual table cells.	**<table cellspacing=”5″>** adds space between cells.
width	Used for setting the width of the table.	**<table width=”100%”>** makes the table 100% wide.
height	Specifies the height of the table.	**<table height=”200px”>** sets the table height.
align	Aligns the table to the left, center, or right of the page.	**<table align=”center”>** centers the table on the page.

bgcolor	Sets the background color of the table.	**<table bgcolor="#f0f0f0″>** adds a light gray background to the table.
summary	Provides a brief description of the table's content for accessibility.	**<table summary="This table lists employee details">** helps screen readers describe the table content.

Example:

```
<table border="1">
   <caption>Table Caption</caption>
   <tr>
      <td>
         Row 1, col 1 item
      </td>
      <td>
         Row 1, col 2 item
      </td>
      <td>
         Row 1, col 3 item
      </td>
   </tr>
   <tr>
      <td>
         Row 2, col 1 item
      </td>
      <td>
         Row 2, col 2 item
      </td>
      <td>
         Row 2, col 3 item
      </td>
   </tr>
</table>
```

Q 17: Explain how to create hyperlinks in web pages. (OR) Explain the anchor tag in html.

- Use the <a> element to define a link.
- Use the href attribute to define the link address.
- Use the target attribute to define where to open the linked document.
- Use the <img> element (inside <a>) to use an image as a link.

Syntax:
<a href="*url*">*link text*</a>
Example:
<a href="www.bitm.edu.in">Visit our College!</a>
By default, links will appear as follows in all browsers:

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

The target attribute specifies where to open the linked document. The target attribute can have one of the following values:

_self - Default. Opens the document in the same window/tab as it was clicked
_blank - Opens the document in a new window or tab
_parent - Opens the document in the parent frame
_top - Opens the document in the full body of the window

Example

```
<h2>Absolute URLs</h2>
<p><a href="https://www.w3.org/">W3C</a></p>
<p><a href="https://www.google.com/">Google</a></p>

<h2>Relative URLs</h2>
<p><a href="html_images.asp">HTML Images</a></p>
<p><a href="/css/default.asp">CSS Tutorial</a></p>
```

Use mailto: inside the href attribute to create a link that opens the user's email program (to let them send a new email):

Example

```
<a href="mailto:someone@example.com">Send email</a>
```

Other examples:

- Link to a page located in the same folder as the current page:
  `<a href="default.asp">HTML tutorial</a>`
- Link to a page located in the html folder on the current web site:
  `<a href="/html/default.asp">HTML tutorial</a>`

Q 18: Explain the concept of frames in HTML.

Ans: HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document independently. A collection of frames in the browser window is known as a frameset. The window is divided into frames in a similar way the tables are organized: into rows and columns.

Example:

```
<frameset rows="50%,50%">
  <frame name="top" src="link/to/frame1" />
  <frame name="bottom" src="link/to/frame2" />
</frameset>
```

Where the rows attribute of frameset defines the division of the window into horizontal sections. In this case, the window is divided into two rows, each taking up 50% of the available height.

Creating Frames in HTML

To make frames on a page we use <frameset> tag instead of <body> tag. The **<frameset>** tag defines how to divide the window into frames. The **rows** attribute of **<frameset>** tag defines horizontal frames and **cols** attribute defines vertical frames. Each frame is indicated by **<frame>** tag and it defines which HTML document shall open into the frame.

Following is the example to create three horizontal frames. If your browser does not support frames, then body element is displayed.

```
<html>
<head>
  <title>HTML Frames</title>
</head>
```

```
<frameset rows="10%,80%,10%">
  <frame name="top" src="/html/top_frame.htm" />
  <frame name="main" src="/html/main_frame.htm" />
  <frame name="bottom" src="/html/bottom_frame.htm" />
  <noframes>
    <body>
      Your browser does not support frames.
    </body>
  </noframes>
</frameset>
</html>
```

Note: The <frame> tag is no longer recommended as it is not supported by HTML5. Instead of using this tag, we can use the <iframe> or <div> with CSS to achieve the similar effects.

Q 19. Explain meta tag in HTML.

The HTML <meta> tag defines metadata about an HTML document, including character set, description, keywords, author, and viewport settings. Placed within the <head> element, metadata aids browsers, search engines, and web services in interpreting and displaying content.

It helps in defining the page's title, encoding, author, and viewport settings, etc. These tags are not visible on the web page but play a vital role in structuring and categorizing content for browsers and search engines.

The following are the attributes of meta tag:

Attribute	**Value**	**Description**
charset	*character_set*	Specifies the character encoding for the HTML document
content	*text*	Specifies the value associated with the http-equiv or name attribute
http-equiv	content-security-policy content-type default-style refresh	Provides an HTTP header for the information/value of the content attribute
name	application-name author description generator keywords viewport	Specifies a name for the metadata

Q 20: Explain span and div tags in HTML.

<span> tag:

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

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

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

```
<span>content goes here</span>
```

In the above syntax, we have-

Opening Tag: The <span> is the opening tag.

Content: Between the opening and closing tags, you can place the content that you want to style or manipulate. This could be text, other HTML elements, or even a combination of both.

Closing Tag: The </span> is the closing tag, which indicates the end of the span element.

Example:

```
<html>
<head>
  <title>Styling Text Example</title>
</head>
<body>
  <p>
    Welcome to <span style="color: blue;">BITM</span>,
    where learning <span style="color: green;">never stops</span>.
  </p>
</body>
</html>
```

<div> tag:

The <div> tag in HTML is a fundamental and versatile element used for grouping and structuring content on a web page. It stands for "division" or "divider." The <div> tag itself doesn't provide any visual formatting or styling. Instead, it is a container for other HTML elements and content. One of the primary purposes of <div> is to enable styling and layout control through CSS (Cascading Style Sheets). Web developers use <div> elements to create layout structures and apply CSS styles to control the appearance and positioning of content on a webpage.

Syntax of div Tag in HTML

```
<div>
  <!-- Your content goes here -->
</div>
```

In the syntax above, we have-

<div> : This is the opening tag of the <div> element.
</div> : This is the closing tag of the <div> element.

<!-- Content goes here -->: This is a comment indicating where you would place your content. You can place any HTML content, including text, images, links, other HTML elements or even nested <div> elements between the opening and closing <div> tags.

```
<html>
  <head>
    <title>Example of div Tag</title>
    <style>
         .mystyles {font-family:Lucida handwriting; color:blue}
    </style>
  </head>

  <body>
    <h1>Example of div Tag</h1>
    <div class="mystyles">
      <h3>Heading inside div tag.</h3>
      <p>Paragraph inside div tag.</p>
      <p>Another paragraph inside div tag.</p>
    </div>
    </body>
 </html>
```

Q 22: Explain any 8 text formatting tags in HTML

1. <i> – Italicizes text
Use the <i> tag to display text in italics without implying emphasis.
Example: <i>This is italic text.</i>

2. <small> – Reduces the font size of the text
The <small> tag renders text in a smaller font than the surrounding text.
Example:
<small>This text is smaller than the rest.</small>

3. <sub> – Displays subscript text
Use the <sub> tag for subscripted text, often used in chemical formulas or footnotes.
Example: H<sub>2</sub>O

4. <b> – Makes text bold
The <b> tag visually makes the text bold but does not imply any special significance.
Example: <b>This is bold text.</b>

5. <mark> – Highlights text with a background color
The <mark> tag highlights text with a background color, similar to using a highlighter on paper.
Example:
<mark>This text is highlighted.</mark>

6. <del> – Strikes through text
The <del> tag is used to show that text has been deleted or is no longer relevant.
Example:
<del>This text is crossed out.</del>

7. <sup> – Displays superscript text
Use the <sup> tag to show superscripted text, commonly used in exponents or footnotes.

Example:
x<sup>2</sup>+y<sup>2</sup>

8. <u>- Displays underlined text
Use this tag to underline the text, commonly used for headings or sub headings
Example:
<u>This text is underlined</u>

Q 23: Write an HTML program to display the following table.

Student details table

Roll No.	**Name**	**Phone Numbers**	
A1234	Hari Prasad G	9898989899	9090909090
A1235	Ishitha M	7878787878	8888889898
A1236	Zeeshan	8898989898	9990909090

```
<html>
  <head>
    <title>Student details</title>
  </head>
  <body>
        <table border='1'>
        <caption>Students details table</caption>
                <tr>
                        <th>Roll No.</th>
                        <th>Name</th>
                        <th colspan='2'>Phone Nos.</th>
                </tr>
                <tr>
                        <td>A1234</td>
                        <td>Hari Prasad G</td>
                        <td>9898989899</td>
                        <td>9090909090</td>
                </tr>
                <tr>
                        <td>A1235</td>
                        <td>Ishitha M</td>
                        <td>7878787878</td>
                        <td>8888889898</td>
                </tr>
                <tr>
                        <td>A1236</td>
                        <td>Zeeshan</td>
                        <td>8898989898</td>
                        <td>9990909090</td>
                </tr>
        </table>
        </body>
</html>
```