

## **Unit I: INTRODUCTION TO WEB TECHNOLOGIES:**

History of the web, Understanding Web System Architecture, Understanding 3-tier Web Architecture, Overview of HTTP,Introducing HTML document structure, Creating Headings on a web page, Working with links,Creating a Paragraph, Working with images (Hot Spots), Working with tables, working with frames,Introduction to Forms and HTML controls. Inline, External, Internal, Style class, Multiple styles.

### **History of the web**

The [World Wide Web](#) ("WWW", "W3" or simply "the Web") is a global [information](#) medium that users can access via [computers](#) connected to the [Internet](#).

[Tim Berners-Lee](#) invented the World Wide Web while working at [CERN](#) in 1989. He proposed a "universal linked information system" using several concepts and technologies, the most fundamental of which was the connections that existed between information.

He developed the first [web server](#), the first [web browser](#), and a document formatting protocol, called [Hypertext Markup Language](#) (HTML).

After publishing the markup language in 1991, and releasing the browser source code for public use in 1993, many other web browsers were soon developed, with [Marc Andreessen's Mosaic](#) (later [Netscape Navigator](#)), being particularly easy to use and install, and often credited with sparking the Internet boom of the 1990s.

It was a graphical browser which ran on several popular office and home computers, bringing multimedia content to non-technical users by including images and text on the same page.

Websites for use by the general public began to emerge in 1993–94. This spurred competition in server and browser software, highlighted in the [Browser wars](#) which was initially dominated by [Netscape Navigator](#) and [Internet Explorer](#).

Following the complete removal of commercial restrictions on Internet use by 1995, commercialization of the Web amidst macroeconomic factors led to the [dot-com boom](#) and bust in the late 1990s and early 2000s.

The features of HTML evolved over time, leading to HTML version 2 in 1995, HTML3 and HTML4 in 1997, and [HTML5](#) in 2014.

The language was extended with advanced formatting in [Cascading Style Sheets \(CSS\)](#) and with [programming](#) capability by [JavaScript](#).

[AJAX programming](#) delivered dynamic content to users, which sparked a new era in [Web design](#), styled [Web 2.0](#).

The use of [social media](#), becoming common-place in the 2010s, allowed users to compose multimedia content without programming skills, making the Web ubiquitous in every-day life.

## **Understanding Web System Architecture**

*“For many decades we have been browsing our favorite websites on the internet and getting a quick response whenever we want... but do you ever try to know how each part of the application works together and how is the request being processed behind the scenes?”*

*When you’re building an application, you need to remember three principles in your mind...*

- *From the **customer’s point** of view, the application shouldn’t be complicated, it should be pleasing and it should address most of their problems.*
- *From a **business aspect**, your web application should stay aligned with its product/market fit*
- *From an **engineer’s perspective**, the application should be scalable, functional, and it should be able to bear high traffic loads.*

## **What is Web Application Architecture?**

**Do you know that there is a difference between websites and web applications?**

(you might have thought that both are the same).

The web application is a program that runs on a browser and it has mainly three formal characteristics.

- Addresses a particular problem, even if it's simply finding some information
- As interactive as a desktop application
- Works with Content Management System.

If we talk about the website then traditionally it's just a combination of static pages.

**Note:** A website becomes a web application when it consists of both static and dynamic pages

- *Web application architecture is a mechanism that gives us a clarification on how the connection is established between the client and the server.*
- *It determines how the components in an application communicate with each other*
- *In technical terms, when a user makes a request on a website, various components of the applications, user interfaces, middleware systems, databases, servers, and the browser interact with each other.*
- *Web Application Architecture is a framework that ties up this relation together and maintains the interaction between these components.*

### *How Does The Web Request Work?*

*Let's take an example if you want to visit Flipkart.com.*

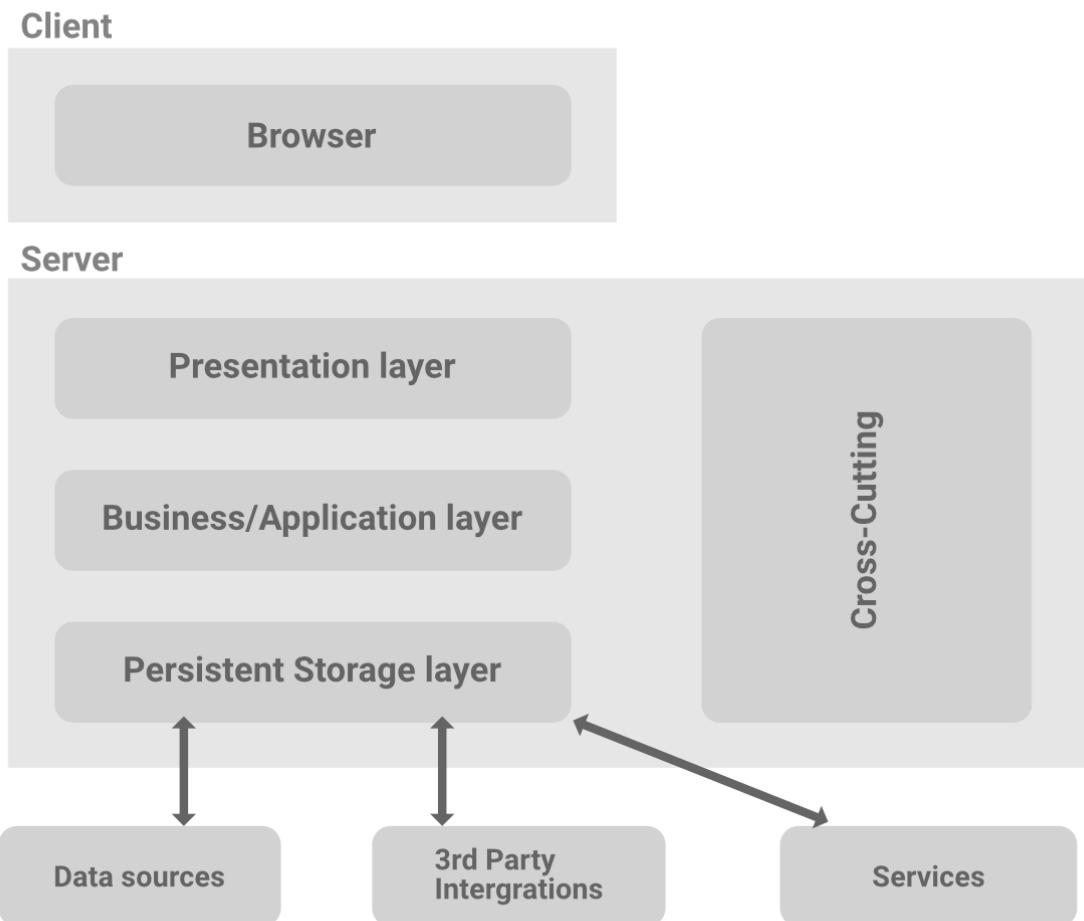
- **You enter flipkart.com in your browser:**
  - When you type the URL in your web browser and hit enter, the web browser needs to know the address of the server where the page is located.
  - So it sends the request to the **DNS** (Domain Name Server, which is a storehouse of domain names and their IP addresses ).

- After that browser sends the request to the found IP address using the *HTTPS protocol*.
- If you already have visited *Flipkart.com* earlier from the same browser, then it will pull the address from the *cache*.
- **The web server processes the request:**
  - In the next step, the web server sends the request to the storage area to locate the page and all data that follows with it.
  - Here the *Business logic* (also called *Domain Logic and Application Logic*) comes in the picture.
  - *BL* takes the responsibility of routing which means how each piece of data is accessed.
  - It regulates this workflow specifically for each application. As *BL* processes the request, it sends it to storage to locate the looked-for data.
- **You get the response back:**
  - When the response travels back, it displays in front of your screen.
  - The *web page (graphical interface)* for any website which is displayed on your screen is called the *front end of an application*.
  - Here you see all the *UI and UX components* to access the information.

## **Web Application Three Tier Architecture Layers**

Web application architectural patterns are separated into many different layers or tiers which is called Multi- or Three-Tier Architecture. You can easily replace and upgrade each layer independently.

## Web Application Architecture



### ***Presentation Layer:***

- This layer is accessible to the client via a browser and it includes user interface components and UI process components.
- UI components are built with HTML, CSS, and JavaScript (and its frameworks or library) where each of them plays a different role in building the user interface.

### ***Business Layer:***

- It is also referred to as a Business Logic or Domain Logic or Application Layer.

- *It accepts the user's request from the browser, processes it, and regulates the routes through which the data will be accessed.*
- *The whole workflow is encoded in this layer.*
- *You can take the example of **booking a hotel on a website**. A traveler will go through a sequence of events to book the hotel room and the whole workflow will be taken care of by the business logic.*

### **Persistence Layer:**

- *It is also referred to as a **storage or data access** layer.*
- *This layer collects all the data calls and provides access to the persistent storage of an application.*
- *The **business layer** is closely attached to the persistence layer, so the logic knows which database to talk to and the process of retrieving data becomes more optimized.*
- *A server and a database management system software exist in data storage infrastructure which is used to communicate with the database itself, applications, and user interfaces to retrieve data and parse it.*
- *You can store the data in hardware servers or in the cloud.*

*Some other parts of the web application which are separated from the main layers that exist in the architecture are...*

- **Cross-cutting code:** This part handles communications, operational management, and security. It affects all parts of the system but should never mix with them.
- **Third-party integrations:** Using third-party APIs we can integrate payment gateways, social logins, GDSs in travel websites, etc.

## ***Types of Web Application Architecture***

### ***1. Single Page Applications:***

*Today a lot of modern web applications are designed as single-page web applications that only include the most required elements and information to generate an intuitive and interactive user experience.*

*In a single page application, the content or information is updated on the current page rather than loading a new page from the server for each action performed by the user.*

- *The application requests only necessary content details and it prevents interruptions in the user experience.*
- *AJAX, Asynchronous JavaScript, and XML are mainly used for page communication.*
- *Users can continue the interaction with the page while the contents are updated on the page (faster interaction).*

### ***2. Microservices:***

- *These are small and lightweight services that execute specific, single functionality.*
- *The components in the applications are not dependent on each other, so there is no need to develop each component using the same programming language.*
- *This gives the flexibility to the developers to choose the language or technology stack of their own choice. It enhances the productivity of the developers and speeds up the development process.*

### **3. Serverless Architectures:**

- In this approach, developers outsource the server and infrastructure management from a third-party cloud infrastructure services provider.
- The advantage of this approach is that it allows the application to execute the required or custom logic without worrying about the infrastructure-related tasks.
- This approach is mainly preferred by the companies who don't want to manage or support the servers and the hardware they have developed the web application for.

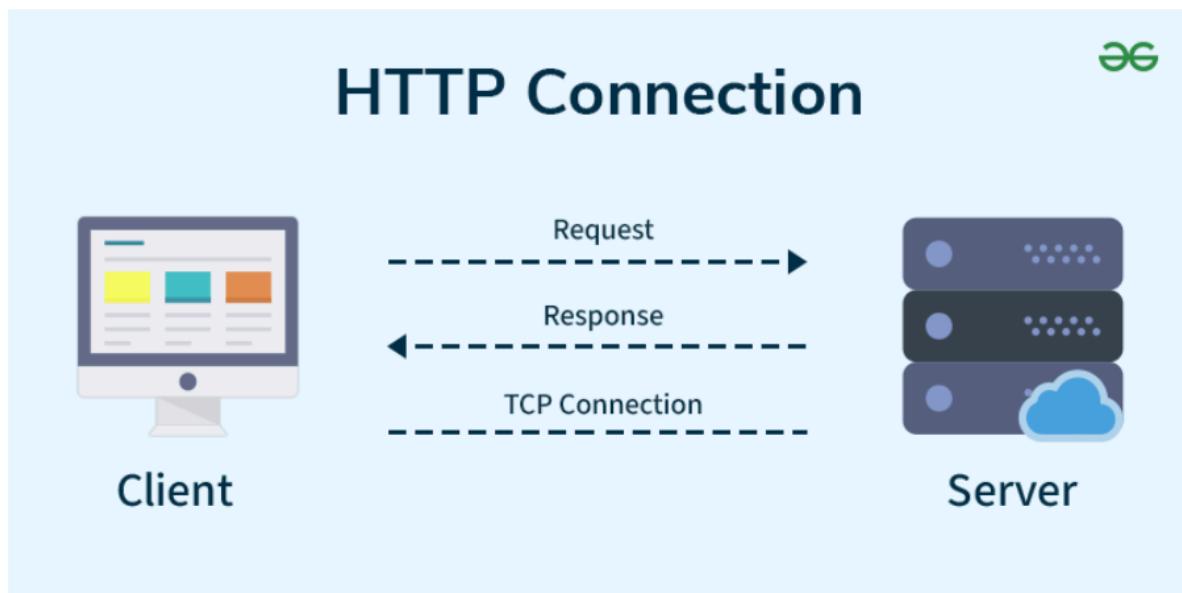
## **Overview of HTTP**

### **What is HyperText?**

- The protocol used to transfer hypertext between two computers is known as HyperText Transfer Protocol.
- HTTP provides a standard between a web browser and a web server to establish communication.
- It is a set of rules for transferring data from one computer to another. Data such as text, images, and other multimedia files are shared on the World Wide Web.
- Whenever a web user opens their web browser, the user indirectly uses HTTP.
- It is an application protocol that is used for distributed, collaborative, hypermedia information systems.

## Working of HTTP [HyperText Transfer Protocol]

- First of all, whenever we want to open any website we first open a web browser. After that we will type the URL of that website (e.g., [www.facebook.com](http://www.facebook.com) ).
- This URL is now sent to the [Domain Name Server \(DNS\)](#).
- Then DNS first checks records for this URL in their database, and then DNS will return the IP address to the web browser corresponding to this URL. Now the browser is able to send requests to the actual server.
- After the server sends data to the client, the connection will be closed.
- If we want something else from the server we should have to re-establish the connection between the client and the server.



### What is an HTTP Request?

HTTP request is simply termed as the information or data that is needed by Internet browsers for loading a website. This is simply known as HTTP Request.

There is some common information that is generally present in all HTTP requests. These are mentioned below.

- HTTP Version
- URL
- HTTP Method
- HTTP Request Headers
- HTTP Body

## **HTTP Request Headers**

HTTP Request Headers generally store information in the form of key-value and must be present in each HTTP Request. The use of this Request Header is to provide core information about the client's information, etc.

## **HTTP Request Body**

HTTP Request Body simply contains the information that has to be transferred. HTTP Request has the information or data to be sent to these browsers.

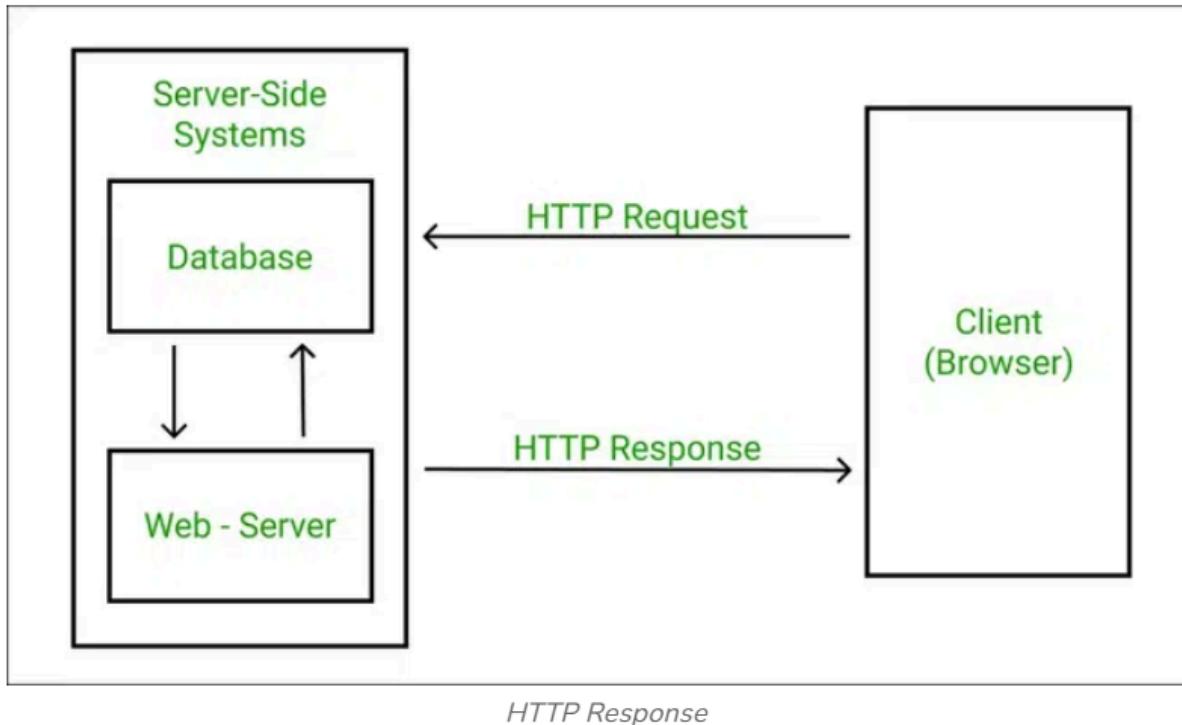
## **HTTP Method**

HTTP Methods are simply HTTP Verb. In spite of being present so many HTTP Methods, the most common HTTP Methods are **HTTP GET and HTTP POST**. These two are generally used in HTTP cases. In HTTP GET, the information is received in the form of a website.

## **What is HTTP Response?**

HTTP Response is simply the answer to what a Server gets when the request is raised. There are various things contained in HTTP Response, some of them are listed below.

- HTTP Status Code
- HTTP Headers
- HTTP Body



## HTTP Response Headers

HTTP Response headers are simply like an HTTP Request where it has that work to send some important files and data to the HTTP Response Body.

## HTTP Response Body

HTTP Responses are the responses that are received successfully upon the request. Generally, it comes under the requests generated by the web. In most cases, the request is to transfer the HTML data into a webpage.

## What is an HTTP Status Code?

HTTP Status Codes are the 3-digit codes that tell the message or simply tell us about the HTTP Request whether it has been completed or not. There are simply 5 types of status codes.

- Informational responses (100–199)
- Successful responses (200–299)
- Redirects (300–399)

- Client errors (400–499)
- Server errors (500–599)

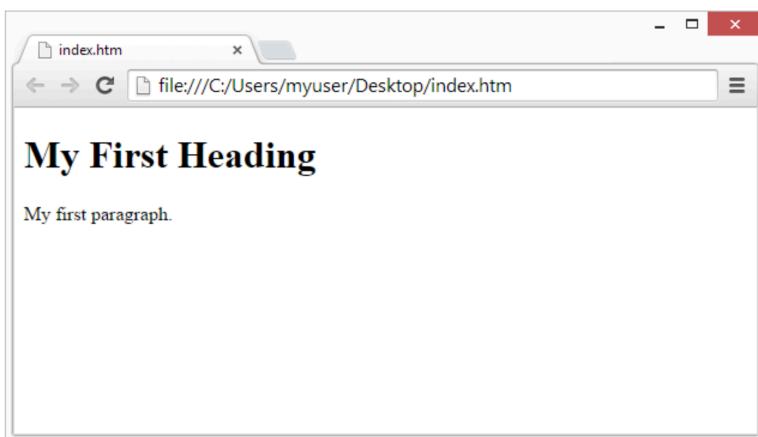
## Introducing HTML document structure

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</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.
- The <h1> element defines a large heading
- The <p> element defines a paragraph



```
<html>

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

    <body>
        <h1>This is a heading</h1>
        <p>This is a paragraph.</p>
        <p>This is another paragraph.</p>
    </body>
</html>
```

## Creating Headings on a web page

**HTML headings are titles or subtitles that you want to display on a webpage.**

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

    <h1>Heading 1</h1>
    <h2>Heading 2</h2>
    <h3>Heading 3</h3>
    <h4>Heading 4</h4>
    <h5>Heading 5</h5>
    <h6>Heading 6</h6>

</body>
</html>
```

# Heading 1

## Heading 2

### Heading 3

#### Heading 4

##### Heading 5

###### Heading 6

- HTML headings are defined with the `<h1>` to `<h6>` tags.
- `<h1>` defines the most important heading. `<h6>` defines the least important heading.

## Bigger Headings

```
<h1 style="font-size:60px;">Heading 1</h1>
```

## HTML - The Head Element

The HTML `<head>` element is a container for the following elements: `<title>`, `<style>`, `<meta>`, `<link>`, `<script>`, and `<base>`.

**title:**

```
<head>
  <title>A Meaningful Page Title</title>
</head>
```

**style:**

```
<style>
  body {background-color: powderblue;}
  h1 {color: red;}
  p {color: blue;}
</style>
```

**Link:**

```
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
```

**meta:**

```
<head>
<meta charset="UTF-8">
<meta name="description" content="Free Web tutorials">
<meta name="keywords" content="HTML, CSS, JavaScript">
<meta name="author" content="John Doe">
</head>
```

**Define the character set used:**

```
<meta charset="UTF-8">
```

**Define keywords for search engines:**

```
<meta name="keywords" content="HTML, CSS, JavaScript">
```

**Define a description of your web page:**

```
<meta name="description" content="Free Web tutorials">
```

**Define the author of a page:**

```
<meta name="author" content="John Doe">
```

**Refresh document every 30 seconds:**

```
<meta http-equiv="refresh" content="30">
```

**Setting the viewport to make your website look good on all devices:**

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

### script:

The **<script>** element is used to define client-side JavaScripts.

```
<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Hello
JavaScript!";
}
</script>
```

### Base:

The **<base>** element specifies the base URL and/or target for all relative URLs in a page.

The **<base>** tag must have either an href or a target attribute present, or both.

There can only be one single **<base>** element in a document

### Example:

```
<head>
<base href="https://www.w3schools.com/" target="_blank">
</head>

<body>
![Stickman](images/stickman.gif)

```

## Working with HTML links

- HTML links are hyperlinks.
- You can click on a link and jump to another document.
- When you move the mouse over a link, the mouse arrow will turn into a little hand

### HTML Links - Syntax

The HTML **<a>** tag defines a hyperlink. It has the following syntax:

```
<a href="url">link text</a>
```

## Example

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

<h1>HTML Links</h1>
<p><a href="https://www.w3schools.com/">Visit W3Schools.com!</a></p>

</body>
</html>
```

# HTML Links

[Visit W3Schools.com!](https://www.w3schools.com/)

## HTML Links - The target Attribute

By default, the linked page will be displayed in the current browser window. To change this, you must specify another target for the link.

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:

```
<a href="https://www.w3schools.com/" target="_blank">Visit W3Schools!</a>
```

## Absolute URLs vs. Relative URLs

Both examples above are using an absolute URL (a full web address) in the **href** attribute.

A local link (a link to a page within the same website) is specified with a relative URL (without the "https://www" part):

## <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>
```

### **HTML Links - Use an Image as a Link**

```
<a href="default.asp">  
  
</a>
```

### **Link to an Email Address**

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

### **Button as a Link**

```
<button onclick="document.location='default.asp'">HTML Tutorial</button>
```

### **Link Titles**

```
<a href="https://www.w3schools.com/html/" title="Go to W3Schools HTML  
section">Visit our HTML Tutorial</a>
```

### **Creating a Paragraph**

The HTML `<p>` element defines a paragraph.

**A paragraph always starts on a new line**, and browsers automatically add some white space (a margin) before and after a paragraph.

#### **Example:**

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

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

**Output:**

This is a paragraph.

This is a paragraph.

This is a paragraph.

**HTML Display**

- You cannot be sure how HTML will be displayed.
- Large or small screens, and resized windows will create different results.
- With HTML, you cannot change the display by adding extra spaces or extra lines in your HTML code.
- The browser will automatically remove any extra spaces and lines when the page is displayed:

**Example:**

```
<p>  
This paragraph  
contains a lot of spaces  
in the source code,  
but the browser  
ignores it.  
</p>
```

## Working with images (Hot Spots) - Image Maps

- With HTML image maps, you can create clickable areas on an image.

## Image Maps

The HTML **<map>** tag defines an image map. An image map is an image with clickable areas. The areas are defined with one or more **<area>** tags.

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

<h2>Image Maps</h2>

<p>Click on the computer, the phone, or the cup of coffee to go to a new page and read more about the topic:</p>



<map name="workmap">
  <area shape="rect" coords="34,44,270,350" alt="Computer"
 href="computer.htm">

  <area shape="rect" coords="290,172,333,250" alt="Phone"
 href="phone.htm">

  <area shape="circle" coords="337,300,44" alt="Cup of coffee"
 href="coffee.htm">

</map>

</body>
</html>
```

## How Does it Work?

The idea behind an image map is that you should be able to perform different actions depending on where in the image you click.

To create an image map you need an image, and some HTML code that describes the clickable areas.

## The Image

The image is inserted using the `<img>` tag. The only difference from other images is that you must add a `usemap` attribute:

```

```

The `usemap` value starts with a hash tag # followed by the name of the image map, and is used to create a relationship between the image and the image map.

## Create Image Map

Then, add a `<map>` element.

The `<map>` element is used to create an image map, and is linked to the image by using the required `name` attribute:

```
<map name="workmap">
```

The `name` attribute must have the same value as the `<img>`'s `usemap` attribute.

## The Areas

Then, add the clickable areas.

A clickable area is defined using an `<area>` element.

## Shape

You must define the shape of the clickable area, and you can choose one of these values:

- **rect** - defines a rectangular region
- **circle** - defines a circular region
- **poly** - defines a polygonal region
- **default** - defines the entire region

You must also define some coordinates to be able to place the clickable area onto the image.



### Shape="rect"

The coordinates for `shape="rect"` come in pairs, one for the x-axis and one for the y-axis.

So, the coordinates `34, 44` is located 34 pixels from the left margin and 44 pixels from the top:



The coordinates `270, 350` is located 270 pixels from the left margin and 350 pixels from the top:



### Working with tables

## What is HTML Tables?

An **HTML Table** is an arrangement of data in **rows** and **columns** in **tabular format**. Tables are useful for various tasks, such as presenting text information and numerical data. A table is a useful tool for quickly and easily finding connections between different types of data. Tables are also used to create databases.

## Tags used in HTML Tables

HTML Tags	Descriptions
<u>&lt;table&gt;</u>	Defines the structure for organizing data in rows and columns within a web page.
<u>&lt;tr&gt;</u>	Represents a <b>row</b> within an HTML table, containing individual cells.
<u>&lt;th&gt;</u>	Shows a table <b>header</b> cell that typically holds titles or headings.
<u>&lt;td&gt;</u>	Represents a standard <b>data</b> cell, holding content or data.

<u>&lt;caption&gt;</u>	Provides a title or description for the entire table.
<u>&lt;thead&gt;</u>	Defines the header section of a table, often containing column labels.
<u>&lt;tbody&gt;</u>	Represents the main content area of a table, separating it from the header or footer.
<u>&lt;tfoot&gt;</u>	Specifies the footer section of a table, typically holding summaries or totals.
<u>&lt;col&gt;</u>	Defines attributes for table <b>columns</b> that can be applied to multiple columns at once.

## Defining Tables in HTML

An HTML table is defined with the “table” tag. Each table row is defined with the “tr” tag. A table header is defined with the “th” tag. By default, table headings are bold and centered. A table data/cell is defined with the “td” tag.

### Table Cells

Table Cell are the building blocks for defining the Table. It is denoted with `<td>` as a start tag & `</td>` as a end tag.

#### Syntax

```
</td> Content...</td>
```

### Table Rows

The rows can be formed with the help of combination of Table Cells. It is denoted by `<tr>` and `</tr>` tag as a start & end tags.

## Syntax

```
</tr> Content...</tr>
```

## Table Headers

The Headers are generally used to provide the Heading. The Table Headers can also be used to add the heading to the Table. This contains the `<th>` & `</th>` tags.

## Syntax

```
</th> Content...</th>
```

**Example 1:** Creating a simple table in HTML using a table tag.

```
<!-- index.html -->
<!DOCTYPE html>
<html>

<body>
    <table>
        <tr>
            <th>Book Name</th>
            <th>Author Name</th>
            <th>Genre</th>
        </tr>
        <tr>
            <td>The Book Thief</td>
            <td>Markus Zusak</td>
            <td>Historical Fiction</td>
        </tr>
        <tr>
            <td>The Cruel Prince</td>
            <td>Holly Black</td>
            <td>Fantasy</td>
        </tr>
        <tr>
            <td>The Silent Patient</td>
            <td> Alex Michaelides</td>
            <td>Psychological Fiction</td>
        </tr>
    </table>
</body>

</html>
```

Output:

Book Name	Author Name	Genre
The Book Thief	Markus Zusak	Historical Fiction
The Cruel Prince	Holly Black	Fantasy
The Silent Patient	Alex Michaelides	Psychological Fiction

## Adding a border to an HTML Table

A border is set using the CSS border property. If you do not specify a border for the table, it will be displayed without borders.

### Syntax

```
table, th, td {  
    border: 1px solid black;  
}
```

Example: Addition of the border to the HTML Table.

```
<!-- index.html -->  
<!DOCTYPE html>  
<html>  
  
<head>  
    <style>  
        table,  
        th,  
        td {  
            border: 1px solid black;  
        }  
    </style>  
</head>
```

```

<body>
    <table style="width:100%">
        <tr>
            <th>Firstname</th>
            <th>Lastname</th>
            <th>Age</th>
        </tr>
        <tr>
            <td>Priya</td>
            <td>Sharma</td>
            <td>24</td>
        </tr>
        <tr>
            <td>Arun</td>
            <td>Singh</td>
            <td>32</td>
        </tr>
        <tr>
            <td>Sam</td>
            <td>Watson</td>
            <td>41</td>
        </tr>
    </table>
</body>

</html>

```

**Output:**

Firstname	Lastname	Age
Priya	Sharma	24
Arun	Singh	32
Sam	Watson	41

## Adding Collapsed Borders in an HTML Table

For borders to collapse into one border, add the CSS border-collapse property.

### Syntax

```
table, th, td {  
    border: 1px solid black;  
    border-collapse: collapse;  
}
```

**Example:** Addition of Collapsed Borders in HTML.

```
<!-- index.html -->  
<!DOCTYPE html>  
<html>  
  
<head>  
    <style>  
        table,  
        th,  
        td {  
            border: 1px solid black;  
            border-collapse: collapse;  
        }  
    </style>  
</head>
```

```

<body>
    <table style="width:100%">
        <tr>
            <th>Firstname</th>
            <th>Lastname</th>
            <th>Age</th>
        </tr>
        <tr>
            <td>Priya</td>
            <td>Sharma</td>
            <td>24</td>
        </tr>
        <tr>
            <td>Arun</td>
            <td>Singh</td>
            <td>32</td>
        </tr>
        <tr>
            <td>Sam</td>
            <td>Watson</td>
            <td>41</td>
        </tr>
    </table>
</body>

</html>

```

**Output:**

Firstname	Lastname	Age
Priya	Sharma	24
Arun	Singh	32
Sam	Watson	41

*HTML Table with Colored Borders*

## Adding Cell Padding in an HTML Table

Cell padding specifies the space between the cell content and its borders. If we do not specify a padding, the table cells will be displayed without padding.

### Syntax

```
th, td {  
    padding: 20px;  
}
```

**Example:** Addition of Table cell padding in HTML.

```
<html>  
  
<head>  
    <style>  
        table,  
        th,  
        td {  
            border: 1px solid black;  
            border-collapse: collapse;  
        }  
  
        th,  
        td {  
            padding: 20px;  
        }  
    </style>  
</head>
```

```
<body>
    <table style="width:100%">
        <tr>
            <th>Firstname</th>
            <th>Lastname</th>
            <th>Age</th>
        </tr>
        <tr>
            <td>Priya</td>
            <td>Sharma</td>
            <td>24</td>
        </tr>
        <tr>
            <td>Arun</td>
            <td>Singh</td>
            <td>32</td>
        </tr>
        <tr>
            <td>Sam</td>
            <td>Watson</td>
            <td>41</td>
        </tr>
    </table>
</body>
```

### Output:

Firstname	Lastname	Age
Priya	Sharma	24
Arun	Singh	32
Sam	Watson	41

Addition Table with padding

## Adding Border Spacing in an HTML Table

Border spacing specifies the space between the cells. To set the border-spacing for a table, we must use the CSS border-spacing property.

### Syntax

```
table {
    border-spacing: 5px;
}
```

**Example:** Explains the border space property to make the space between the Table cells.

```
<!-- index.html -->
<!DOCTYPE html>
<html>

<head>
    <style>
        table,
        th,
        td {
            border: 1px solid black;
        }

        table {
            border-spacing: 5px;
        }
    </style>
</head>
```

```
<body>
  <table style="width:100%">
    <tr>
      <th>Firstname</th>
      <th>Lastname</th>
      <th>Age</th>
    </tr>
    <tr>
      <td>Priya</td>
      <td>Sharma</td>
      <td>24</td>
    </tr>
    <tr>
      <td>Arun</td>
      <td>Singh</td>
      <td>32</td>
    </tr>
    <tr>
      <td>Sam</td>
      <td>Watson</td>
      <td>41</td>
    </tr>
  </table>
</body>

</html>
```

**Output:**

Firstname	Lastname	Age
Priya	Sharma	24
Arun	Singh	32
Sam	Watson	41

## Adding Cells that Span Many Columns in HTML Tables

To make a cell span more than one column, we must use the colspan attribute.

**Example:** Use of [colspan attribute](#) in HTML.

```
<!-- index.html -->
<!DOCTYPE html>
<html>

<head>
    <style>
        table,
        th,
        td {
            border: 1px solid black;
            border-collapse: collapse;
        }

        th,
        td {
            padding: 5px;
            text-align: left;
        }
    </style>
</head>
```

```

<body>
    <h2>Cell that spans two columns:</h2>
    <table style="width:100%">
        <tr>
            <th>Name</th>
            <th colspan="2">Telephone</th>
        </tr>
        <tr>
            <td>Vikas Rawat</td>
            <td>9125577854</td>
            <td>8565557785</td>
        </tr>
    </table>
</body>

</html>

```

Output:

### Cell that spans two columns:

Name	Telephone	
Vikas Rawat	9125577854	8565557785

*colspan attribute*

### Adding Cells that span many rows in HTML Tables

To make a cell span more than one row, we must use the rowspan attribute.

**Example:** Use of the [rowspan attribute](#) in HTML.

```
<!-- index.html -->
<!DOCTYPE html>
<html>

<head>
    <style>
        table,
        th,
        td {
            border: 1px solid black;
            border-collapse: collapse;
        }

        th,
        td {
            padding: 5px;
            text-align: left;
        }
    </style>
</head>
```

```

<body>
    <h2>Cell that spans two rows:</h2>
    <table style="width:100%">
        <tr>
            <th>Name:</th>
            <td>Vikas Rawat</td>
        </tr>
        <tr>
            <th rowspan="2">Telephone:</th>
            <td>9125577854</td>
        </tr>
        <tr>
            <td>8565557785</td>
        </tr>
    </table>
</body>

</html>

```

Output:

### **Cell that spans two rows:**

Name:	Vikas Rawat
Telephone:	9125577854
	8565557785

*Use of rowspan attribute*

### **Adding a Caption in an HTML Table**

To add a caption to a table, we must use the “caption” tag.

### **Syntax**

```
<table style="width:100%">  
<caption>DETAILS</caption>
```

```
<!-- index.html -->  
<!DOCTYPE html>  
<html>  
  
<head>  
    <style>  
        table,  
        th,  
        td {  
            border: 1px solid black;  
            border-collapse: collapse;  
        }  
  
        th,  
        td {  
            padding: 20px;  
        }  
  
        th {  
            text-align: left;  
        }  
    </style>  
</head>
```

```
<body>
  <table style="width:100%">
    <caption>DETAILS</caption>
    <tr>
      <th>Firstname</th>
      <th>Lastname</th>
      <th>Age</th>
    </tr>
    <tr>
      <td>Priya</td>
      <td>Sharma</td>
      <td>24</td>
    </tr>
    <tr>
      <td>Arun</td>
      <td>Singh</td>
      <td>32</td>
    </tr>
    <tr>
      <td>Sam</td>
      <td>Watson</td>
      <td>41</td>
    </tr>
  </table>
</body>

</html>
```

### Output:

DETAILS		
Firstname	Lastname	Age
Priya	Sharma	24
Arun	Singh	32
Sam	Watson	41

## Adding a Background Colour to the Table

A color can be added as a background in an HTML table using the “background-color” option.

### Syntax

```
table#t01 {
    width: 100%;
    background-color: #f2f2d1;
}
```

**Example:** Addition of the Table background color in HTML.

```
<!-- index.html -->
<!DOCTYPE html>
<html>

<head>
    <style>
        table,
        th,
        td {
            border: 1px solid black;
            border-collapse: collapse;
        }

        th,
        td {
            padding: 5px;
            text-align: left;
        }

        table#t01 {
            width: 100%;
            background-color: #f2f2d1;
        }
    </style>
</head>
```

```
<body>
  <table style="width:100%">
    <tr>
      <th>Firstname</th>
      <th>Lastname</th>
      <th>Age</th>
    </tr>
    <tr>
      <td>Priya</td>
      <td>Sharma</td>
      <td>24</td>
    </tr>
    <tr>
      <td>Arun</td>
      <td>Singh</td>
      <td>32</td>
    </tr>
    <tr>
      <td>Sam</td>
      <td>Watson</td>
      <td>41</td>
    </tr>
  </table>
  <br />
  <br />
```

```
<table id="t01">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Priya</td>
    <td>Sharma</td>
    <td>24</td>
  </tr>
  <tr>
    <td>Arun</td>
    <td>Singh</td>
    <td>32</td>
  </tr>
  <tr>
    <td>Sam</td>
    <td>Watson</td>
    <td>41</td>
  </tr>
</table>
</body>

</html>
```

### Output:

Firstname	Lastname	Age
Priya	Sharma	24
Arun	Singh	32
Sam	Watson	41

Firstname	Lastname	Age
Priya	Sharma	24
Arun	Singh	32
Sam	Watson	41

*Adding Table Background color using CSS properties*

## Creating Nested Tables

Nesting tables simply means making a Table inside another Table. Nesting tables can lead to complex tables layouts, which are visually interesting and have the potential of introducing errors.

**Example:** Nesting of HTML Table.

```
<!-- index.html -->
<!DOCTYPE html>
<html>

<body>
    <table border=5 bordercolor=black>
        <tr>
            <td> First Column of Outer Table </td>
            <td>
                <table border=5 bordercolor=grey>
                    <tr>
                        <td> First row of Inner Table </td>
                    </tr>
                    <tr>
                        <td> Second row of Inner Table </td>
                    </tr>
                </table>
            </td>
        </tr>
    </table>
</body>

</html>
```

## Output:

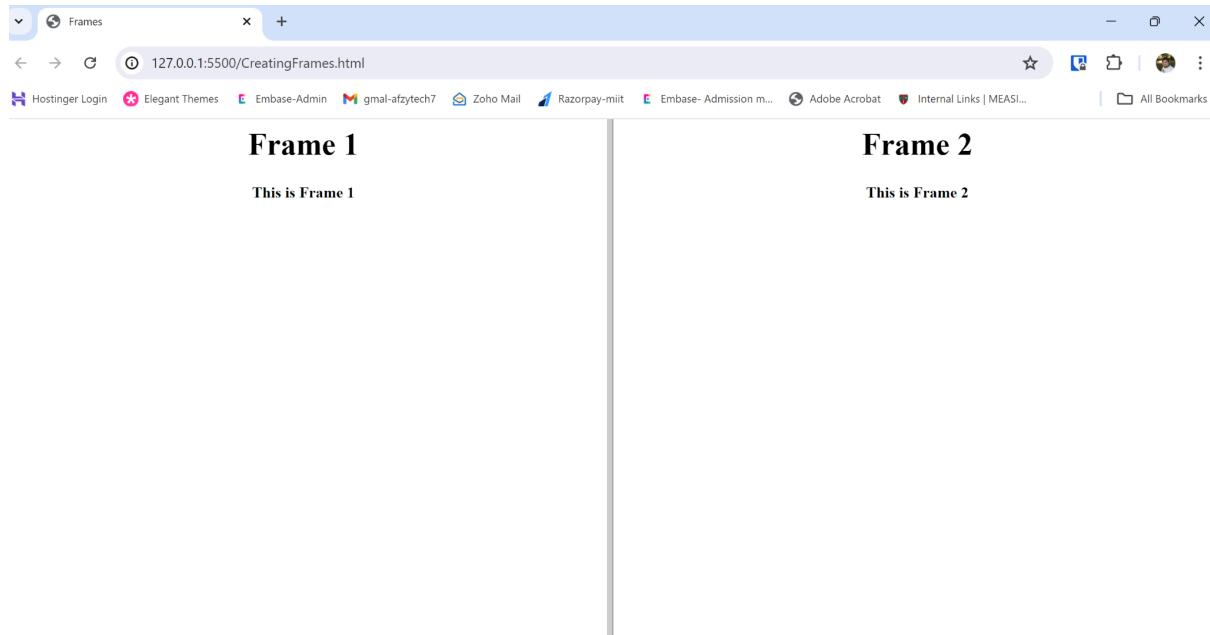
Fisrt Column of Outer Table	First row of Inner Table
	Second row of Inner Table

# HTML - Frames

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.

## **creatingframes.html**

```
<html>
<head>
<title>Frames</title>
</head>
<frameset cols="50% ,50%">
<frame src="Frame1.html">
<frame src="Frame2.html">
</frameset>
<noframes></noframes>
</html>
```



### **frame1.html**

```
<html>
<head>
<title>Frames</title>
</head>
<body>
<h1 align="center">Frame 1</h1>
<h4 align="center">This is Frame 1</h4>
</body>
</html>
```

### **frame2.html**

```
<html>
<head>
<title>Frames</title>
</head>
<body>
<h1 align="center">Frame 2</h1>
<h4 align="center">This is Frame 2</h4>
</body>
</html>
```

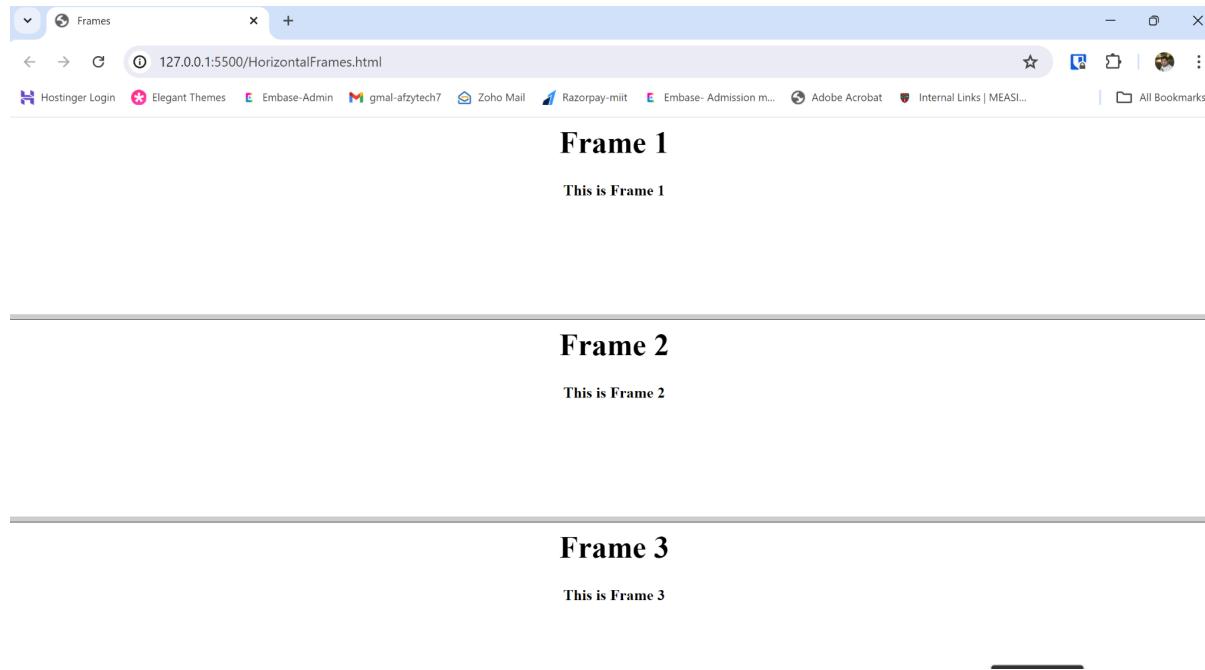
### **frame3.html**

```
<html>
<head>
<title>Frames</title>
</head>
<body>
<h1 align="center">Frame 3</h1>
<h4 align="center">This is Frame 3</h4>
</body>
</html>
```

### **horizontalframes.html**

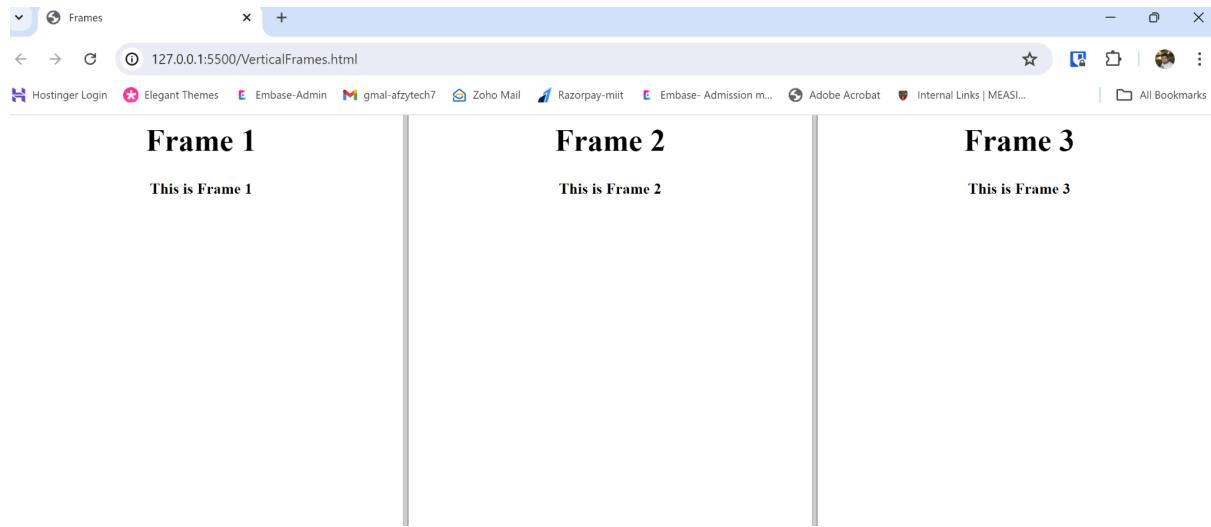
```
<html>
<head>
<title>Frames</title>
</head>
<frameset rows="33%,33%,33%">
<frame src="Frame1.html">
```

```
<frame src="Frame2.html">
<frame src="Frame3.html">
</frameset>
<noframes></noframes>
</html>
```



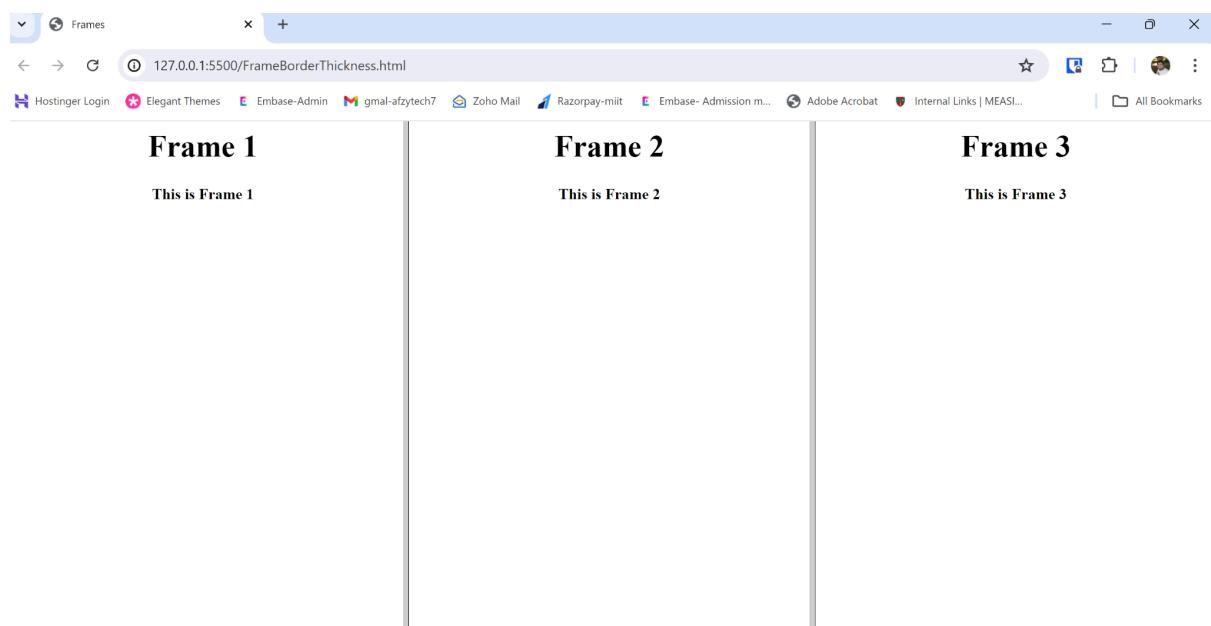
## **verticalframes.html**

```
<html>
<head>
<title>Frames</title>
</head>
<frameset cols="33%,33%,33%">
<frame src="Frame1.html">
<frame src="Frame2.html">
<frame src="Frame3.html">
</frameset>
<noframes></noframes>
</html>
```



## frame\_border\_thickness.html

```
<html>
<head>
<title>Frames</title>
</head>
<frameset cols="33%,33%,33%" frameborder="1" framespacing="10">
<frame src="Frame1.html">
<frame src="Frame2.html">
<frame src="Frame3.html">
</frameset>
<noframes></noframes>
</html>
```



## Introduction to Forms and HTML controls

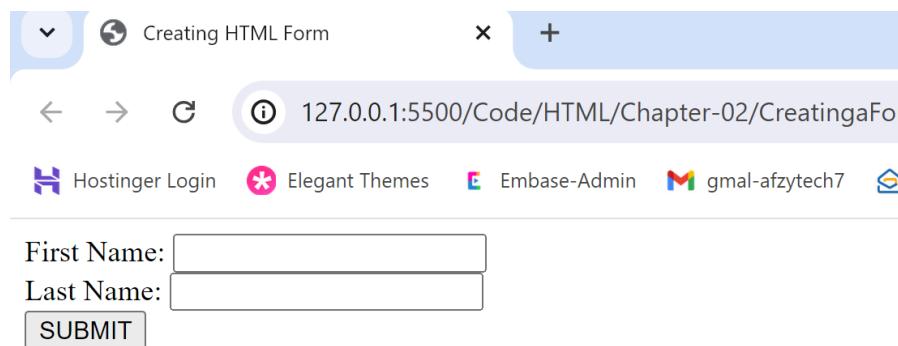
The basic purpose of an html form is to allow user to enter the data on one end and then send the data on other end through webserver.

### Creating html forms

```

<html>
<head>
<title>
Creating HTML Form
</title>
</head>
<body>
<form>
First Name:
<input type="text" name="firstname">
<br>
Last Name:
<input type="text" name="lastname">
<br>
<input type="submit" value="SUBMIT">
</form>
</body>
</html>

```



## Form with action url:

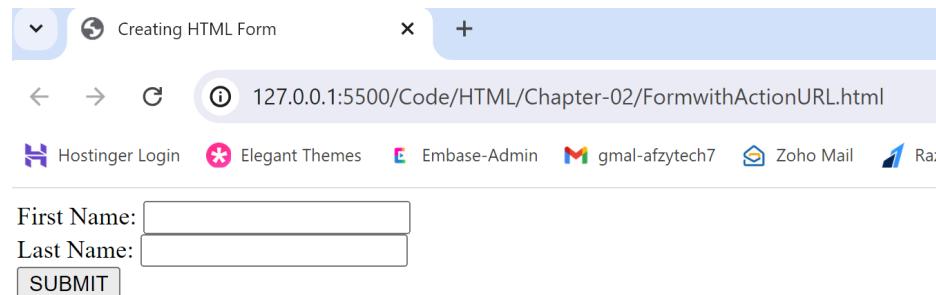
Method Attribute	
Table 2.8: Values of the Method Attribute	
Value	Description
method="get"	Indicates that the form data has to be encoded by the Web browser into a URL. This is the default method.
method="post"	Indicates that the form data appears within the message body.

Let's create a Web page, named `FormwithActionURL.html` to learn how to specify the action URL.

```

<html>
<head>
<title>
Creating HTML Form
</title>
</head>
<body>
<form action="Example.html" method="post">
First Name:
<input type="text" name="firstname">
<br>
Last Name:
<input type="text" name="lastname">
<br>
<input type="submit" value="SUBMIT">
</form>
<p>
If you click the <b><u>"SUBMIT"</u></b> button you will be redirected
to the
<b><u>Example.html</u></b> page.
</p>
</body>
</html>

```



Creating HTML Form

127.0.0.1:5500/Code/HTML/Chapter-02/FormwithActionURL.html

Hostinger Login    Elegant Themes    Embase-Admin    gmal-afzytech7    Zoho Mail    Raz

First Name:

Last Name:

If you click the "SUBMIT" button you will be redirected to the Example.html page.



## USING HTML CONTROLS:

### USING INPUT TAGS TO ADD CONTROLS

Table 2.9: Attributes of the <input> Tag

Attribute	Value	Description
align	left right top texttop middle absmiddle	Defines the alignment of the text following the image. It is used only with type="image".

Attribute	Value	Description
	baseline bottom absbottom	
alt	text /	Defines an alternate text for the image. It is used only with type="image".
checked	checked	Indicates that the input element should be checked when it loads on the Web server. It is used only with type="checkbox" and type="radio".
disabled	disabled	Disables the input element when it loads on the Web server, so that the user cannot write text in it or select it. It cannot be used with type="hidden".
maxlength	number	Specifies the maximum number of characters allowed in a text field. It can be used only with type="text".
name	field_name	Defines a unique name for the input element. This attribute is required with type="button", type="checkbox", type="field", type="hidden", type="image", type="password", type="text", and type="radio".
readonly	readonly	Indicates that the value of this field cannot be modified. It can be used only with type="text".
src	URL	Defines the URL of the image to display. It can be used only with type="image".
type	button checkbox file hidden image password radio reset submit text	Indicates the type of the input element. The default value is "text".

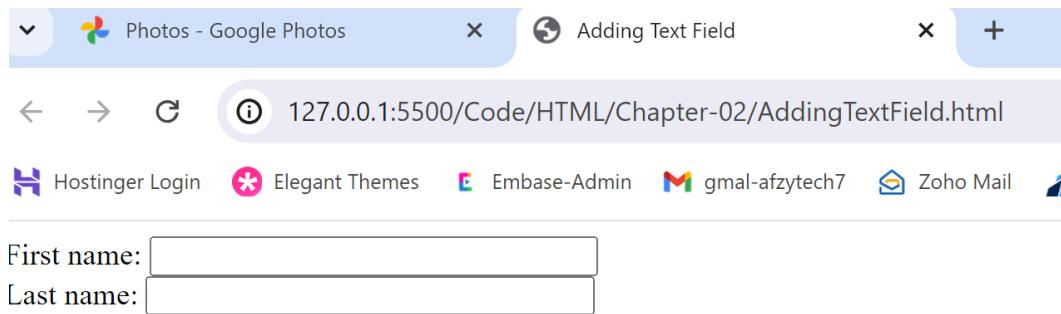
### ADDING TEXT FIELD TO THE FORM:

```
<html>
<head>
<title>
Adding Text Field
```

```

</title>
</head>
<body>
<form action="">
First name:
<input type="text" name="firstname" size="30" />
<br />
Last name:
<input type="text" name="lastname" size="30" />
</form>
</body>
</html>

```

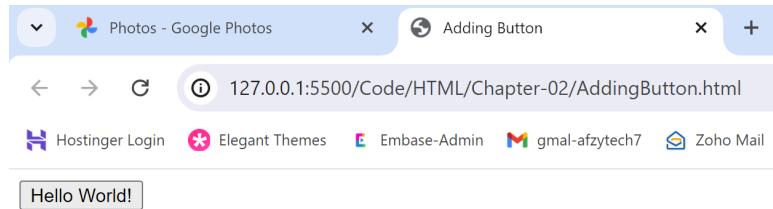


### ADDING BUTTON:

```

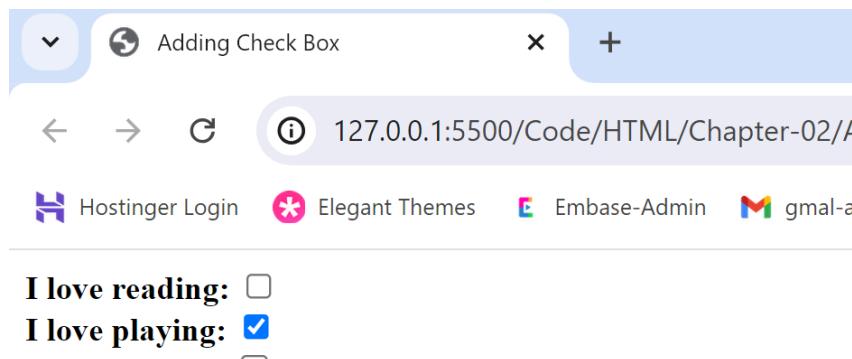
<html>
<head>
<title>
Adding Button
</title>
</head>
<body>
<form action="">
<input type="button" value="Hello World!">
</form>
</body>
</html>

```



### ADDING CHECKBOX:

```
<html>
<head>
<title>
Adding Check Box
</title>
</head>
<body>
<form action="">
<b>I love reading:</b>
<input type="checkbox" name="hobby" value="books">
<br>
<b>I love playing:</b>
<input type="checkbox" name="hobby" value="play" checked>
<br>
<b>I love driving:</b>
<input type="checkbox" name="hobby" value="drive">
</form>
</body>
</html>
```



### ADDING RADIO BUTTON:

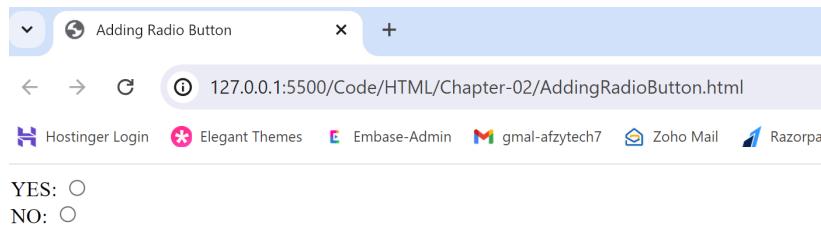
```
<html>
<head>
<title>
Adding Radio Button
</title>
</head>
<body>
<form action="">
```

YES:

```
<input type="radio" name="poll" value="yes"/>
</br>
```

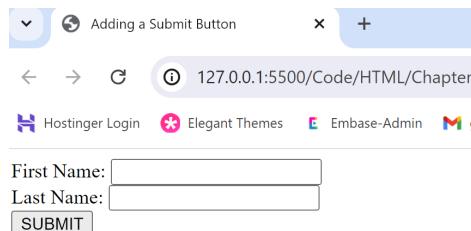
NO:

```
<input type="radio" name="poll" value="no"/>
</form>
</body>
</html>
```



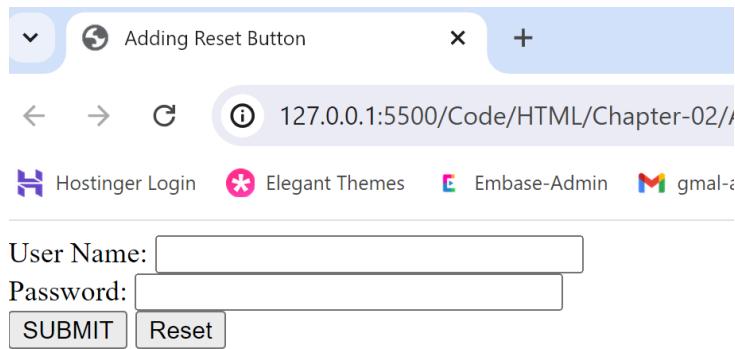
### ADDING SUBMIT BUTTON:

```
<html>
<head>
<title>
Adding a Submit Button
</title>
</head>
<body>
<form action="Example.html" method="post">
First Name:
<input type="text" name="Enter your name">
<br>
Last Name:
<input type="password" name="Password">
<br>
<input type="submit" value="SUBMIT">
</form>
</body>
</html>
```



## ADDING RESET BUTTON

```
<html>
<head>
<title>
Adding Reset Button
</title>
</head>
<body>
<form action="Example.html" method="post">
User Name:
<input type="text" size="30"/>
<br>
Password:
<input type="password" size="30"/>
<br>
<input type="submit" Value="SUBMIT">
<input type="reset">
</form>
</body>
</html>
```



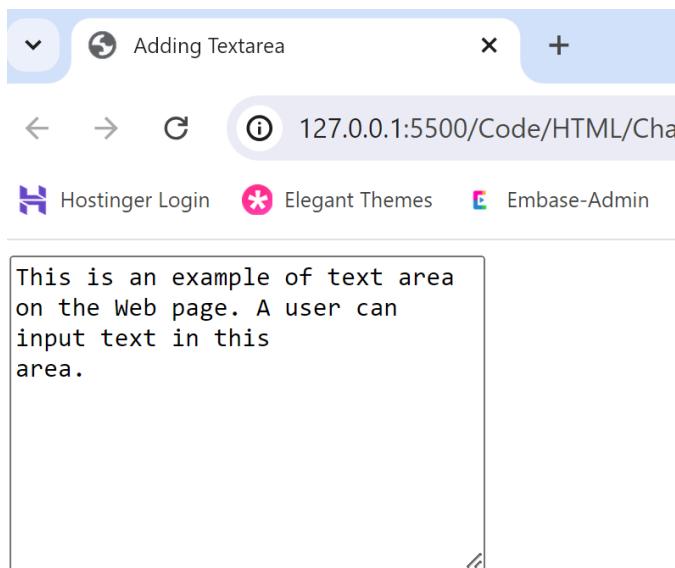
## ADDING A TEXT AREA:

```
<html>
<head>
<title>
Adding Textarea
</title>
</head>
<body>
```

```

<form>
<textarea rows="10" cols="30">
This is an example of text area on the Web page. A user can input text
in this
area.
</textarea>
</form>
</body>
</html>

```



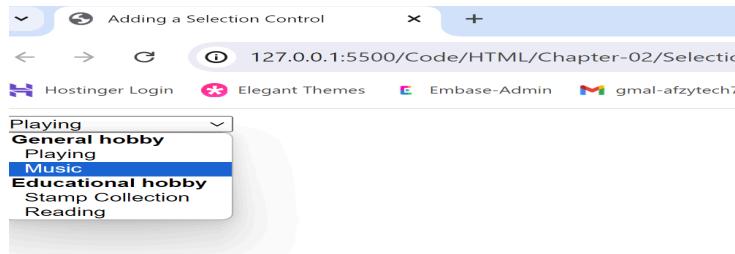
### ADDING SELECTION CONTROL:

```

<html>
<head>
<title>
Adding a Selection Control
</title>
</head>
<body>
<form>
<select>
<optgroup label="General hobby">
<option value ="playing">Playing</option>
<option value ="music">Music</option>
</optgroup>
<optgroup label="Educational hobby">
<option value ="stamp collection">Stamp Collection</option>
<option value ="reading">Reading</option>

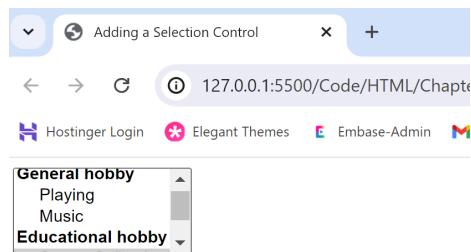
```

```
</optgroup>
</select>
</form>
</body>
</html>
```



## ADDING MULTIPLE SELECTION CONTROL

```
<html>
<head>
<title>
    Adding a Selection Control
</title>
</head>
<body>
<form>
<select MULTIPLE>
<optgroup label="General hobby">
<option value ="playing">Playing</option>
<option value ="music">Music</option>
</optgroup>
<optgroup label="Educational hobby">
<option value ="stamp collection">Stamp Collection</option>
<option value ="reading">Reading</option>
</optgroup>
</select>
</form>
</body>
</html>
```





## Inline, External, Internal, Style class, Multiple styles

### Inline Style Sheets:

```
<html>
<head>
<title>Inline Styles</title>
</head>
<body>
<table border="1">
<caption><h2>Student Details</h2></caption>
<th style="background-color: #800000">Name</th>
<th style="background-color: #800000">Date of Birth</th>
<th style="background-color: #800000">Address</th>
<tr>
<td style="background-color: #808080">Sumit Saxena</td>
<td style="background-color: #808080">15-03- 1983</td>
<td style="background-color: #808080">Flat No.303, Shipra
Suncity,
Ghaziabad</td>
</tr>
<tr>
<td style="background-color: #008000">Amitabh Kumar</td>
<td style="background-color: #008000">22-02- 1984</td>
<td style="background-color: #008000">H.No- 125, Kalkaji,
Delhi</td>
</tr>
<tr>
<td style="background-color: #00ff00">Rohit Jandial</td>
<td style="background-color: #00ff00">05-07- 1983</td>
<td style="background-color: #00ff00">Flat No- 324, South Ext,
Delhi</td>
</tr>
<tr>
<td style="background-color: #808000">Avantika Srivastava</td>
<td style="background-color: #808000">10-12- 1984</td>
<td style="background-color: #808000">H.No-541, Vikas Puri,
Delhi</td>
</tr>
```

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

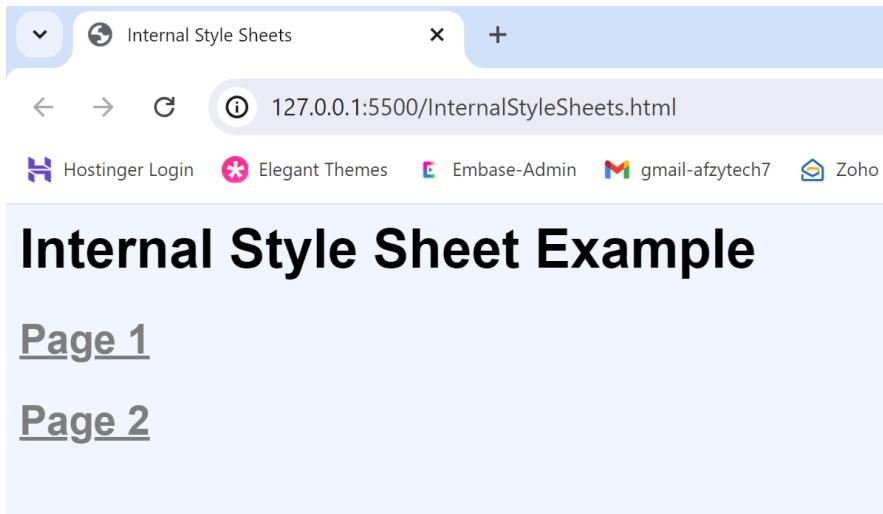
**Output:**

## Student Details

Name	Date of Birth	Address
Sumit Saxena	15-03- 1983	Flat No.303, Shipra Suncity, Ghaziabad
Amitabh Kumar	22-02- 1984	H.No- 125, Kalkaji, Delhi
Rohit Jandial	05-07- 1983	Flat No- 324, South Ext, Delhi
Avantika Srivastava	10-12- 1984	H.No-541, Vikas Puri, Delhi

### Internal Style Sheets:

```
<html>
<head>
<title>Internal Style Sheets</title>
<style type="text/css">
body
{background-color:#f0f8ff; font-family:Arial}
a:link {color: #808080}
a:visited {color: #ffff00}
a:hover {color: #00ff00}
a:active {color: #ff0000}
</style>
</head>
<body>
<h1>Internal Style Sheet Example</h1>
<a href= Page1.html target="_blank">
<h2>Page 1</h2>
<a href= Page2.html target="_blank">
<h2>Page 2</h2>
</body>
</html>
```



# Internal Style Sheet Example

[Page 1](#)

[Page 2](#)

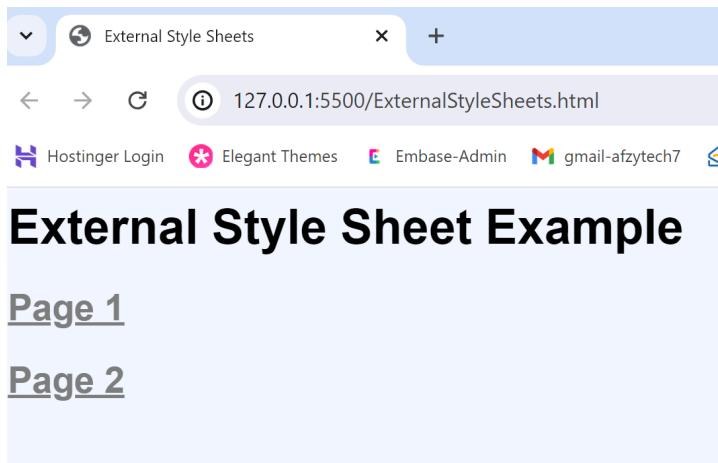
### External Style Sheet:

`external_style.css`

```
<html>
<head>
<title>External Style Sheets</title>
<link rel="stylesheet" href="Style.css">
</head>
<body>
<h1>External Style Sheet Example</h1>
<a href= "Page1.html" target="_blank">
<h2>Page 1</h2>
<a href= "Page2.html" target="_blank">
<h2>Page 2</h2>
</body>
</html>
```

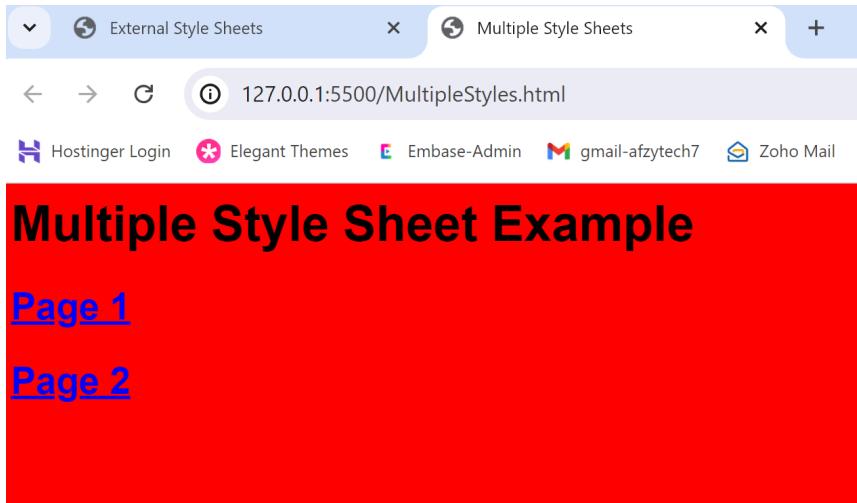
`style.css`

```
body{background-color:#f0f8ff; font-family: Arial}
a:link {color: #808080}
a:visited {color: #0000ff}
a:hover {color: #00ff00}
a:active {color: #ff0000}
```



## Multiple Style Sheets:

```
<html>
<head>
<title>Multiple Style Sheets</title>
<link rel="stylesheet" type="text/css" href="Style.css">
<link rel="stylesheet" type="text/css" href="style1.css">
<link rel="stylesheet" type="text/css" href="style2.css">
</head>
<body>
<h1>Multiple Style Sheet Example</h1>
<a href= Page1.html target="_blank">
<h2>Page 1</h2>
<a href= Page2.html target="_blank">
<h2>Page 2</h2>
</body>
</html>
```



A screenshot of a web browser window. At the top, there are two tabs: "External Style Sheets" and "Multiple Style Sheets". The "Multiple Style Sheets" tab is active, showing the URL "127.0.0.1:5500/MultipleStyles.html". Below the tabs is a toolbar with icons for back, forward, search, and refresh. To the right of the toolbar, there are several links: "Hostinger Login", "Elegant Themes", "Embase-Admin", "gmail-afzytech7", and "Zoho Mail". The main content area has a red background and contains the text "Multiple Style Sheet Example" in large black font. Below it, there are two blue hyperlinks: "Page 1" and "Page 2".