
Chapter 1: Internet Basics & Web Terms

Introduction to the Internet

>Internet provides a global network, where interconnected nodes are accessible to organizations and individuals through communication devices and media. A **node** refers to any device, such as a computer, tablet, or smartphone that is part of a network. A network is formed when two or more computers are connected wired or wirelessly to share resources and information.

>Currently, **Internet of Things (IoT)** encompasses the growth of devices connecting to a network, ranging from televisions to household appliances. Data transfer lines in networks facilitate the transfer of data between different computers. The Internet is the backbone consists of high-speed data transfer connections that interconnect major computer systems worldwide. (Server, cloud...etc.)

Confidential | ©2024 EduSkills



Domain Name System (DNS)

- DNS or **Domain Name System**, serves as the Internet's equivalent of a phonebook. While humans access information online through easily memorable domain names like google.com or eduskillsfoundation.org, web browsers communicate through less user-friendly Internet Protocol (IP) addresses. DNS plays a crucial role in translating these human-readable domain names into machine-friendly IP addresses, enabling browsers to load the desired Internet resources.
- In the intricate process of DNS resolution, a hostname (e.g., www.example.com) is converted into a corresponding IP address (e.g., 192.168.1.1).
- Point to remember, An **Internet Service Provider (ISP)** is a company with a permanent connection to the Internet backbone, enabling it to provide access to the internet for users.
- Similar to how a home address guides us to a specific home, an IP address is essential for locating a particular device/server on the Internet. When users wish to access a webpage/website, there must be a translation between what they input into their web browser (e.g., example.com) and the machine-friendly address required to locate the desired webpage.

Confidential | ©2024 EduSkills



Four key DNS servers functions for loading a webpage:

1. DNS Recursor:

Analogy: A librarian tasked with finding a specific book in a vast library.

Function: The recursor receives queries from client machines, often through web browsers. It is responsible for making additional requests to fulfill the client's DNS query.

2.Root Nameserver:

Analogy: An index in a library pointing to various racks of books.

Function: The first step in translating human-readable host names into IP addresses. It serves as a reference to more specific locations.

3. TLD Nameserver (Top-Level Domain):

Analogy: A specific rack of books in a library.

Function: The next step in the search for a specific IP address, hosting the last portion of a hostname (e.g., in example.com, the TLD server is "com").

Confidential | ©2024 EduSkills



4. Authoritative Nameserver:

Analogy: A dictionary on a rack of books for translating specific names into definitions.

Function: The final stop in the nameserver query. If the authoritative nameserver has the requested record, it returns the IP address for the requested hostname to the DNS recursor that initiated the initial request.

- In short, **DNS ensures** a seamless translation process, allowing users to access websites using human-readable domain names while behind the scenes, a complex series of servers work together to convert these names into the necessary machine-friendly IP addresses.

Understanding Web Terminology

1. Static Web Page

- A static web page is a webpage that is sent to users exactly as it's stored, unlike dynamic web pages that are created on the spot by a web application.
- Here are some key points about static web pages:
- They show the same information to all users, regardless of their context.
- Modern web servers can adjust the content type or language based on user preferences.
- Static web pages are typically HTML documents saved as files in the file system and provided by the web server through HTTP.
- However, there are drawbacks to static web pages:
- Personalization and interactivity must be handled on the user's device, which can be limiting.
- Managing a large number of static pages as files can be challenging without automated tools.

2. Dynamic Web Page

- A dynamic web page is one that shows different content based on input from a user or a computer program.
- Here are some key points about dynamic web pages:
- They use both client-side scripting and server-side scripting for dynamic behavior.
- Client-side scripting changes how elements on a web page behave in response to user actions or specific timing events.
- Server-side scripting involves programs running on a web server, modifying content on various pages or adjusting the sequence of pages. Server responses depend on conditions like data in a form, parameters in the URL, the user's browser type, time passage, or the state of a database or server.

HTTP Protocol and Web Servers

- HTTP, which stands for **HyperText Transfer Protocol**, is a protocol designed for accessing data on the World Wide Web (www). It enables the transfer of various types of data, including plain text, hypertext, audio, and video. The efficiency of HTTP is evident in its suitability for hypertext environments, allowing seamless transitions between different documents.

Key Features of HTTP:

Connectionless Protocol: HTTP operates as a connectionless protocol, where the client initiates a request, awaits a server response, and then disconnects. The connection exists only during the request and response phase.

Media Independent: HTTP is media-independent, allowing data transmission as long as both the client and server understand how to handle the content. The content type must be specified by both parties in the MIME-type header.

Stateless: HTTP is a stateless protocol, meaning that during a request, both client and server are aware of each other, but this awareness is limited to the current transaction, and no information is retained between different web page requests.

- The process of HTTP transactions between client and server involves the client initiating a transaction by sending a request message, to which the server responds with a corresponding response message.

Note: Currently, all website are using HyperText Transfer Protocol Secure (HTTPS) since it's secure than HTTP

HTTP Request Methods:

- **GET:** Requests the resource from the specified URL.
- **POST:** Requests the server to process and store the attached body information. Similar to a GET request, but includes additional data.
- **HEAD:** Requests only the header part of the information that a GET request would provide, without the body.
- **TRACE:** Requests the loopback of the sent message, useful for testing or diagnosing issues.
- **PUT:** Instructs to place the enclosed information (body) at the specified URL.
- **DELETE:** Instructs to remove the resource at the specified URL.
- **OPTIONS:** Asks for a list of the HTTP methods that the resource at the requested URL can support.

GET vs POST

- In a Get request, a limited amount of data is sent in the header, while in a Post request, a large amount of data can be sent in the body.
- Get requests are not secure because data is exposed in the URL bar, whereas Post requests are secure as data is not exposed in the URL bar.
- Get requests can be bookmarked, but Post requests cannot be bookmarked.
- Get requests are idempotent, meaning the second request will be ignored until the response of the first request is delivered. On the other hand, Post requests are non-idempotent.
- Get requests are more efficient and commonly used, while Post requests are less efficient and used less frequently than Get requests.

Uniform Resource Locator (URL):

For accessing documents on the internet, clients use Uniform Resource Locators (URLs).

These URLs consist of four parts:

Method: Specifies the protocol for retrieving the document (e.g., HTTP).

Host: Identifies the computer where information is stored, often starting with "www." This field is not mandatory.

Port: An **optional field indicating** the server's port number, placed between the host and path, separated by a colon. (e.g., Port: 8080)

Path: The pathname of the file storing the information, including slashes to separate directories and files. (e.g., webproject/homepage)

WEB SERVER

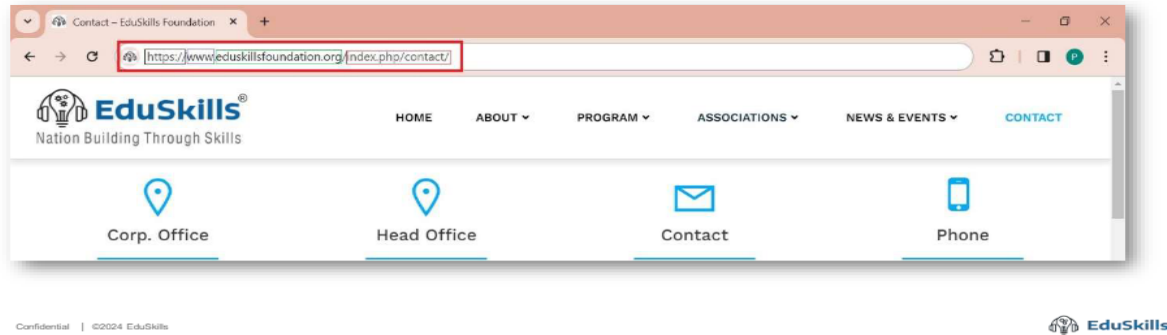
- The term **"web server"** encompasses both hardware and software components working collaboratively. On the hardware side, a web server refers to a computer that houses web server software along with a website's component files such as HTML documents, images, CSS stylesheets, and JavaScript files. This computer connects to the Internet and facilitates physical data exchange with other connected devices.
- On the software side, a web server comprises various elements that regulate how web users access hosted files. At its core, it includes an HTTP server, a software component that comprehends URLs (web addresses) and HTTP (the protocol used by browsers to view webpages). Accessible through the domain names of the stored websites, the HTTP server transmits the content of these websites to the end user's device.
- In essence, when a browser needs a file from a web server, it sends a request via HTTP. The HTTP server on the web server acknowledges the request, locates the requested document, and transmits it back to the browser, also through HTTP. If the server fails to find the requested document, it responds with a 404 error.
- To launch a website, a static or dynamic web server is required. A static web server comprises a computer (hardware) equipped with an HTTP server (software). It is termed "static" because it sends its hosted files unchanged to the user's browser.
- Conversely, a dynamic web server incorporates additional software, typically an application server and a database, in addition to the static web server.

Web Browsers and How the WWW Works

- A **web browser** serves as a software application that interprets and displays web pages, facilitating users in viewing and interacting with online content. Examples of web browsers include Microsoft Internet Explorer, Microsoft Edge, Mozilla Firefox, Google Chrome, Opera, and Apple Safari.



- A Uniform Resource Locator (URL) serves as the address for a document or file accessible on the Internet, such as <https://www.eduskillsfoundation.org/index.php/contact/>. Additionally, a domain refers to a specific area of the Internet managed by a particular organization or individual.



- **Link:** <https://www.eduskillsfoundation.org/index.php/contact/>.

Understanding the given URL link :

Link Portions	Explanation
https://	Protocol (HTTPS or HTTP)
www.	Sub-domain
eduskillsfoundation.org	Server or domain name
/index.php	Webpage location
contact/	Webpage file name

***Note:** Observe Port number is not mentioned, since its optional field.

- The internet is the infrastructure comprising physical networks of computers. The **World Wide Web (WWW)**, commonly known as the web, functions as the service that grants access to information stored on web servers. The web is composed of interconnected files referred to as web pages, and a website is a collection of related web pages created and maintained by an individual, company, educational institution, or organization.
- When users access a website, the first document they encounter is typically the home page. Furthermore, a hyperlink, commonly known as a link, serves as an element connecting one webpage to another on the same server or to any web server worldwide.

Chapter 2: HTML Fundamentals

Grasping the Anatomy of an HTML Page

What is HTML?

- HTML, which stands for Hyper Text Markup Language, is the basic code used to create web pages. It defines the layout and structure of a webpage through a set of elements. These elements instruct the browser on how to present content, such as headings, paragraphs, links, and more. In simpler terms, HTML is like the blueprint that guides the appearance and organization of information on a website.

Let's understand the evolution of HTML in next page.....

HTML Evolution:

HTML 1.0 (1991):

This is where it all began. The first version of HTML was introduced by Tim Berners-Lee in 1991.

It was a simple markup language used to create basic documents with headings, paragraphs, and links.

Think of it as the foundational stage, where the web was just starting to take shape.

HTML 2.0 (1995):

The second version brought improvements and added new features.

It introduced attributes like "align" for better control over page elements.

Tables were introduced, allowing for better organization of content.

HTML 3.2 (1997):

HTML 3.2 expanded further on the capabilities of the language.

This version introduced support for scripting languages like JavaScript.

It also added new elements like the table caption and improved support for forms.

HTML 4.01 (1999):

HTML 4.01 was a significant step forward in standardizing the language.

It introduced style sheets, allowing for better control of page layout and design.

The concept of frames for dividing the browser window into multiple sections was also introduced.

XHTML (2000):

XHTML (eXtensible HyperText Markup Language) was introduced to make HTML more compatible with XML (eXtensible Markup Language).

It followed stricter syntax rules and aimed for a cleaner, more organized code structure.

HTML5 (2014): (currently in use)

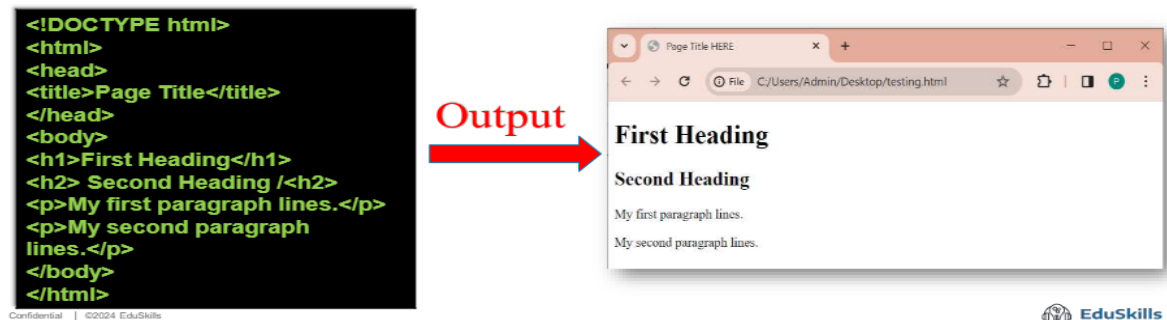
HTML5 marked a major milestone in the evolution of HTML.

It introduced a plethora of new features, including native support for audio and video, new semantic elements (like header, nav, and article), and enhanced forms.

HTML5 also focused on improving the language's compatibility with different devices and browsers.

HTML Elements

- HTML is the primary language for crafting web documents, and these documents typically bear the file extensions .html or .htm. In HTML, tags are strings enclosed by the symbols "<" and ">." For instance, an **opening tag** is represented as <html>, and its corresponding **closing tag** is </html>.
- Now let's try to understand the below basic HTML code:



From the previous page code:

- The **<!DOCTYPE html>** declaration indicates that the document follows the HTML5 standard.
- The **<html>** element serves as the root element for an HTML page.
- The **<head>** element contains meta information related to the HTML page.
- The **<title>** element specifies the title displayed in the browser's title bar or tab.
- The **<body>** element encapsulates all visible content, including headings, paragraphs, images, links, tables, lists, etc.
- The **<h1>** element denotes a large heading.
- The **<h2>** element denotes a medium heading
- The **<p>** element is used to define a paragraph.

Understanding more HTML tags:

- **
** tag defines a line break, and is an empty element without a closing tag.
- **<meta>** tag provides metadata about the HTML document such as descriptions and keywords for search engine.
- **<table>** tags are used to create tables.
- **** tags are used to insert images.
- **<!--...-->** tag is used to write comments.

□ Logical Character Effects

- Heading Styles: **<hn>Heading Here</hn>**, where n = 1, 2, 3, 4, 5, 6
- **<code>** tag is used displays any code part in the web page.
- **<var>** tag is used displays any variable on the web page.
- **<kbd>** tag is used displays computer commands and arguments.
- **** tag is used highlights specific areas of text enclosed within it.
- **<q>** tag defines the short quotation.
- **** tag is used for defining important text.
- **<dfn>** tag is used to designed specifically for words and phrases defined in text.
- **<address>** tag specifies information about the creator e.g. Address, mailing address.
- **<ins>** tag defines inserted text.

1. Table Borders:

- 1. To add Basics border we will be using the CSS border property on **<table>**, **<th>**, and **<td>** elements.

Syntax:

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


Example: 1 code Explanation:

`<!DOCTYPE html>`: This declaration defines the document type and version of HTML being used, in this case, HTML5.

`<html>`: The root element of the HTML document.

`<head>`: Contains meta-information about the HTML document, such as the title and links to external stylesheets or scripts.

`<style>`: This tag is used to embed the CSS (Cascading Style Sheets) code directly within the HTML file.

`table, th, td`: This CSS rule selects all `<table>`, `<th>` (table header cells), and `<td>` (table data cells) elements.

`border: 1px solid black;`: This CSS rule sets a border for the selected elements. It's a 1-pixel wide solid black border.

`</style>`: Closes the style tag.

`</head>`: Closes the head section of the HTML document.

`<body>`: Contains the content of the HTML document.

`<h2>Table With Border</h2>`: This is a level 2 heading displaying the text "Table With Border."

`<p>Use the CSS border property to add a border to the table.</p>`: A paragraph providing a brief explanation of how the CSS border property is used in this example.

`<table style="width:100%">`: Defines the start of an HTML table with a width set to 100% of the containing element (in this case, the body).

`<tr>`: Represents a table row.

`<th>Firstname</th>`, `<th>Lastname</th>`, `<th>Age</th>`: These are table header cells containing column headings for "Firstname," "Lastname," and "Age."

`<td>Jill</td>`, `<td>Smith</td>`, `<td>50</td>`: These are table data cells containing the information for the first row of the table, representing the first person's first name, last name, and age.

Similar rows (`<tr>` and `<td>`) follow for the other two individuals (Eve Jackson and John Doe).

`</table>`: Closes the table.

`</body>`: Closes the body section.

`</html>`: Closes the HTML document.

2. Collapsed Table Borders:

To prevent the occurrence of duplicated borders as seen in basic border sample code, configure the CSS property border-collapse to have a value of "collapse."

Syntax:

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

Example: 2 Code: [part-1 and part-2 are together one code]

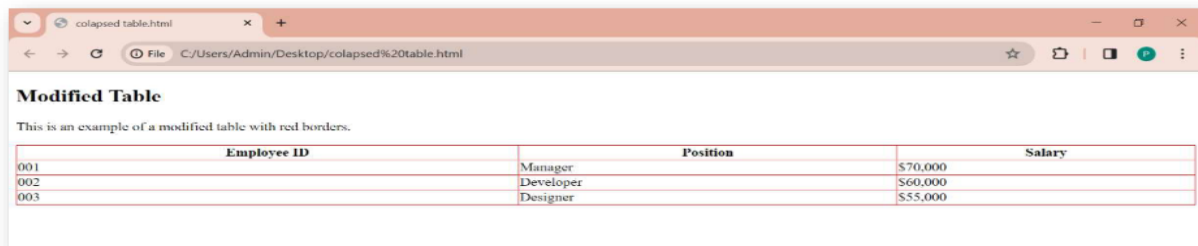
```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
table, th, td {  
  border: 1px solid red; /* Changed border color  
to red */  
  border-collapse: collapse;  
}  
</style>  
</head>  
<body>  
  
<h2>Modified Table</h2>  
<p>This is an example of a modified table with  
red borders.</p>  
<table style="width:100%">  
  <tr>  
    <th>Employee ID</th>  
    <th>Position</th>
```

Part-
1

```
    <th>Salary</th>  
  </tr>  
  <tr>  
    <td>001</td>  
    <td>Manager</td>  
    <td>$70,000</td>  
  </tr>  
  <tr>  
    <td>002</td>  
    <td>Developer</td>  
    <td>$60,000</td>  
  </tr>  
  <tr>  
    <td>003</td>  
    <td>Designer</td>  
    <td>$55,000</td>  
  </tr>  
</table>  
</body>  
</html>
```

Part-
2

Example: 2 Code Output:



Employee ID	Position	Salary
001	Manager	\$70,000
002	Developer	\$60,000
003	Designer	\$55,000

Example: 2 Code Explanation:

<!DOCTYPE html>: This declaration defines the document type and version of HTML being used, which is HTML5 in this case.

<html>: This tag marks the start of the HTML document.

<head>: This section contains metadata about the HTML document, such as the title and linked stylesheets. In this case, it includes a <style> block.

<style>: This block contains CSS (Cascading Style Sheets) rules that define the styling for the HTML elements.

table, th, td: This selector targets the <table>, <th> (table header), and <td> (table data) elements.

border: 1px solid red;: This rule sets the border of the selected elements to be 1 pixel wide and solid, with a red color.

Confidential | ©2024 EduSkills



border-collapse: collapse;: This rule specifies that the borders of the table cells should be collapsed into a single border. This can create a cleaner appearance for tables.

</style>: Marks the end of the <style> block.

</head>: Marks the end of the <head> section.

<body>: This section contains the actual content of the HTML document.

<h2>Modified Table</h2>: This is a level 2 heading that indicates the title of the modified table.

<p>This is an example of a modified table with red borders.</p>: This is a paragraph providing a description or explanation of the table.

<table style="width:100%">: This defines the start of the table. The style attribute is used to set the width of the table to 100% of the container.

<tr>: This tag represents a table row.

<th>: This tag represents a table header cell. The content inside these cells represents column headers.

<td>: This tag represents a table data cell. The content inside these cells represents the actual data.

</table>: Marks the end of the table.

</body>: Marks the end of the <body> section.

</html>: Marks the end of the HTML document.

Confidential | ©2024 EduSkills



3. Nested Tables: We can integrate a table inside another table.

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
</style>
</head>
<body>
<table>
<tr> <th>Region</th> <th> South Zone
office List </th> </tr>
<tr> <td>South</td>
<td><table>
```

Part-1

```
<tr> <th>State</th> <th>Capital City</th> </tr>
<tr> <td>Karnataka</td> <td>Bangalore</td>
</tr>
<tr> <td>Tamil Nadu</td> <td>Chennai</td> </tr>
<tr> <td>Andhra Pradesh</td>
<td>Hyderabad</td>
</tr>
</table></td></tr>
</table>
```

</body>
</html>

Part-2

[Note: Part-1 and part-2 are together one code]

Confidential | ©2024 EduSkills



Example: 3 Code Output:

NestedTable.html

NestedTable.html

File

C:/Users/Admin/Desktop/NestedTable.html

Region	South Zone office List	
South	State	Capital City
	Karnataka	Bangalore
	Tamil Nadu	Chennai
	Andhra Pradesh	Hyderabad

Example: 3 code Explanation:

`<!DOCTYPE html>`: Declares the document type and version of HTML.

`<html>`: The root element of an HTML document.

`<head>`: Contains meta-information about the HTML document, such as the title and linked stylesheets.

`<style>`: Contains CSS (Cascading Style Sheets) code to define the style of the HTML elements.

`table`, `th`, `td`: Selects the table, table header (`th`), and table data (`td`) elements.

`border: 1px solid black;`: Sets a 1-pixel wide solid black border for the selected elements.

`border-collapse: collapse;`: Specifies that the borders of the selected elements should collapse into a single border.

`</style>`: Closes the style block.

`</head>`: Closes the head section.

`<body>`: Contains the content of the HTML document.

`<p>If you want the borders to collapse into one border, add the CSS border-collapse property.</p>`: Provides a paragraph explaining the purpose of the CSS property border-collapse.

`<table>`: Defines the outer table.

`<tr> <th>Region</th> <th> South Zone office List </th> </tr>`: Creates a table row with two table header cells, specifying the column headers.

`<tr> <td>South</td> <td>...</td></tr>`: Creates a table row with two table data cells. The second cell contains another nested table.

`<table>` (nested table): Defines an inner table within the second cell.

`<tr> <th>State</th> <th>Capital City</th> </tr>`: Creates a table row with two table header cells for the inner table.

`<tr> <td>Karnataka</td> <td>Bangalore</td> </tr>`: Creates a table row with two table data cells containing information about the first state.

`<tr> <td>Tamil Nadu</td> <td>Chennai</td> </tr>`: Creates a table row with information about the second state.

`<tr> <td>Andhra Pradesh</td> <td>Hyderabad</td></tr>`: Creates a table row with information about the third state.

`</table>` (inner table): Closes the inner table.

`</td></tr>`: Closes the row and the outer table.

`</table>` (outer table): Closes the outer table.

`</body>`: Closes the body section.

`</html>`: Closes the HTML document.

4. Table Cell Spanning: Table cells have the ability to extend across multiple columns or rows.

Cell spanning includes:

Spanning across rows

Spanning across columns

Syntax:

```
<table>
<tr>
<th rowspan=m>Multiple Column Header</th>
<th colspan=n>Multiple Row Header</th>
</tr>
</table>
```

Note: **m & n** are integers specifying number of rows and columns respectively.

Example: 4 Code: [part-1 and part-2 are together one code]

```
<!DOCTYPE html>
<html>
<head>
</head>
<table border=1>
<tr>
<th>S.No</th><th>Full
Name</th><th>Rating</th>
</tr>
<tr>
<td rowspan="2">1</td>
<td>Ram</td>
<td>3.4</td>
</tr>
<tr>
<td>Krishna</td>
<td>2.4</td>
</tr>
</table>
```

Part-1

```
<td rowspan="3">2</td>
<td>Laksman</td>
<td>4.5</td>
</tr>
<tr>
<td>Sneha</td>
<td>3.2</td>
</tr>
<tr>
<td>Divya</td>
<td>4.2</td>
</tr>
<tr style="background-color:skyblue">
<td colspan="2">Average Rating</td>
<td>3.54</td>
</tr>
</table>
</body>
</html>
```

Part-2

Example: 4 Code Output:



S.No	Full Name	Rating
1	Ram	3.4
	Krishna	2.4
2	Laksman	4.5
	Sneha	3.2
	Divya	4.2
Average Rating		3.54

Confidential | ©2024 EduSkills



Example: 4 Code Explanation:

`<table border=1>`: This opening tag starts the definition of an HTML table. The border attribute with a value of 1 specifies that the table should have a border with a width of 1 pixel.

`<tr>`: This opening tag signifies the beginning of a table row.

`<th>`: This opening tag defines a table header cell. In this case, it is used to create the header row of the table.

`<th>S.No</th>`: This cell contains the text "S.No," representing Serial Number.

`<th>Full Name</th>`: This cell contains the text "Full Name," representing the names of individuals.

`<th>Rating</th>`: This cell contains the text "Rating," representing the ratings of individuals.

`</tr>`: This closing tag marks the end of the header row.

`<tr>`: Another opening tag for a table row, representing the first data row.

`<td rowspan="2">1</td>`: This cell contains the text "1" and has a rowspan attribute set to 2. This means that this cell spans two rows vertically.

`<td>Ram</td>`: This cell contains the text "Ram," representing the full name of the first individual.

`<td>3.4</td>`: This cell contains the text "3.4," representing the rating of Ram.

Confidential | ©2024 EduSkills



`</tr>`: Closing tag for the first data row.

`<tr>`: Another opening tag for a table row, representing the second data row.

`<td>Krishna</td>`: This cell contains the text "Krishna," representing the full name of the second individual.

`<td>2.4</td>`: This cell contains the text "2.4," representing the rating of Krishna.

`</tr>`: Closing tag for the second data row.

`<tr>`: Another opening tag for a table row, representing the third data row.

`<td rowspan="3">2</td>`: This cell contains the text "2" and has a rowspan attribute set to 3. This means that this cell spans three rows vertically.

`<td>Laksman</td>`: This cell contains the text "Laksman," representing the full name of the third individual.

`<td>4.5</td>`: This cell contains the text "4.5," representing the rating of Laksman.

`</tr>`: Closing tag for the third data row.

Confidential | ©2024 EduSkills



Similar structure is repeated for the next two individuals (Sneha and Divya).

`<tr style="background-color:skyblue">`: This opening tag represents a table row with a style attribute, changing the background color to sky blue.

`<td colspan="2">Average Rating</td>`: This cell spans two columns horizontally and contains the text "Average Rating," representing the label for the average rating row.

`<td>3.54</td>`: This cell contains the text "3.54," representing the calculated average rating.

`</tr>`: Closing tag for the row with the average rating.

`</table>`: Closing tag for the entire table.

Cell Spacing & Table Border:

In HTML, you can create tables to organize and display data. Two important aspects related to tables are cell spacing and table borders.

Cell Spacing:

Cell spacing refers to the space between cells in a table. You can control this spacing using the `cellspacing` attribute in the `<table>` tag. This attribute specifies the amount of space between the border of each cell.

Example: 5 Code: [part-1 and part-2 are together one code]

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Cell Spacing Example</title>
  <style>
    table {
      border-collapse: collapse;
    }
    td, th {
      border: 1px solid black;
      padding: 8px;
    }
  </style>
</head>
<body>
```

Part-1

```
<h2>Cell Spacing Example</h2>

<table cellspacing="10">
  <tr>
    <td>Cell 1</td>
    <td>Cell 2</td>
  </tr>
  <tr>
    <td>Cell 3</td>
    <td>Cell 4</td>
  </tr>
</table>
</body>
```

Part-2

Note: In this example, the `cellspacing` attribute is set to 10, which adds space between the cells.

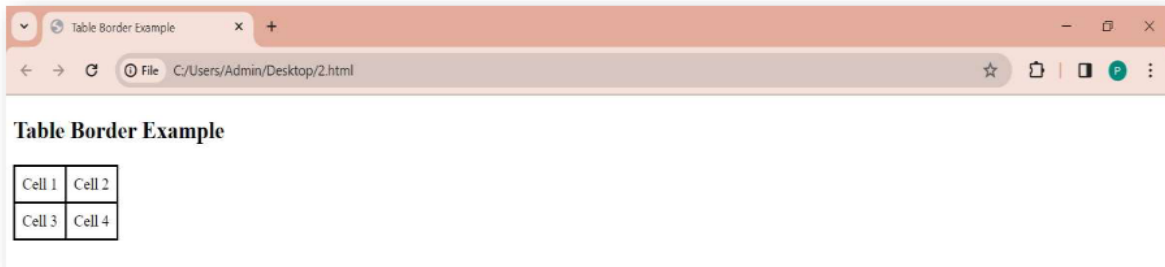
Example: 5 Output:



Table Border:

The border attribute in the <table> tag is used to control the width of the table border. You can set it to 0 for no border or any positive value for a border of that width.

Table Border Example :



Confidential | ©2024 EduSkills



Table Border Example Code: [part-1 and part-2 are together one code]

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title> Table Border
  Example</title>
  <style>
    table {
      border-collapse: collapse;
    }
    td, th {
      border: 2px solid black;
      padding: 8px;
    }
  </style>
</head>
<body>
```

Part-
1

```
<h2>Table Border Example</h2>

<table border="1">
  <tr>
    <td>Cell 1</td>
    <td>Cell 2</td>
  </tr>
  <tr>
    <td>Cell 3</td>
    <td>Cell 4</td>
  </tr>
</table>

</body>
</html>
```

Part-
2



In Table Border Example, the border attribute is set to 1, which adds a border around the entire table.

Note: The border-collapse: collapse; style is used to ensure that adjacent table cell borders are collapsed into a single border. This helps in achieving a cleaner look for the table.

Confidential | ©2024 EduSkills



Chapter 4: Lists

❑ **Numbered List:**

A list that can be either numbered or ordered is referred to as a numbered/ordered list.

Tags utilized to construct an ordered list include:

-
- tag

Defines an individual list item.

Should be employed as a subordinate tag of either the or tag.

Confidential | ©2024 EduSkills



 tag defines an ordered list.

- The ordered list can feature numerical or alphabetical sequencing.
- Attributes within the tag include:
 - start
 - type

```
<!DOCTYPE html>
<html>
<body>
<h2>My Playlist</h2>
<ol>
<li>Song Name 1</li>
<li>Song name 2</li>
</ol>
</body>
</html>
```



Output

My Playlist

1. Song Name 1
2. Song name 2

Confidential | ©2024 EduSkills



Type Attribute:

Indicates the type of marker to utilize in the list.

Acceptable values for the Type Attribute include:

1. A: Uppercase letters
2. a: Lowercase letters
3. I: Uppercase Roman letters
4. i: Lowercase Roman letters
5. 1: Standard numbers (default)

Sample Code:

```
<!DOCTYPE html>
<html>
<body>
<h2>My Playlist</h2>
<ol start="2" type="a">
<li>Song 1
<li>Song 2
<li>Song 3
<li>Song 4
<li>Song 5
</ol>
</body>
</html>
```



Output

My Playlist

- b. Song 1
- c. Song 2
- d. Song 3
- e. Song 4
- f. Song 5

Confidential | ©2024 EduSkills



❑ **Bulleted List:** List of items prefixed with bullets is called Bulleted /Unordered List

 tag defines an unordered list. In short, an unordered list items will be marked with bullets.

Example:

```
<!DOCTYPE html>
<html>
<body>
<h2>My Playlist</h2>
<ul>
<li>Song Name 1</li>
<li>Song name 2</li>
</ul>
</body>
</html>
```

Confidential | ©2024 EduSkills

Output

My Playlist

- Song Name 1
- Song name 2

EduSkills

Style Attribute:

Employed within the tag, the style attribute defines the appearance of the marker.

Example: Implementing <UL style="list-style-type:circle"> will result in an unordered list of items prefixed with circles.

Style Attribute Value Descriptions:

list-style-type:disc - Marks list items with bullets (default).

list-style-type:circle - Marks list items with circles.

list-style-type:square - Marks list items with squares.

list-style-type:none - Prevents the marking of list items.

list-style-image: url('logo.gif') - Marks list items with a specified image ('logo.gif').

Confidential | ©2024 EduSkills

EduSkills

Chapter 5: Frames

Recognizing the Importance of Frames in Web Pages

Frames enable the nesting of multiple HTML document contents within a single webpage. The utilization of frames in a webpage is driven by several factors:

Enhanced Reusability: Frames allow for the efficient reuse of HTML content, contributing to a more modular and easily maintainable structure.

Persistent Static Navigation: By incorporating frames, static navigation elements can remain visible at all times, providing a consistent and accessible user interface.

Resolution of Design Issues: Frames offer a means to address design challenges effectively, ensuring a cohesive and aesthetically pleasing layout for the webpage.

Advertisement Display: Utilizing frames facilitates the display of advertisements, thereby promoting the website and potentially increasing its visibility and engagement.

The Inline Frame Element (<iframe>) serves the purpose of embedding another HTML page within the current page seamlessly.

The <iframe> can be seamlessly incorporated within the body of an HTML document, much like the usage of an tag.

Confidential | ©2024 EduSkills

EduSkills

The syntax for implementing <iframe> is as follows:

Various types of documents can be embedded using <iframe>, including images (GIF, JPEG, etc.), PDF files, audio files, video files, HTML documents, and more.

Attribute



Attribute Name	Attribute Value	Description
name	Name of the frame	Specifies the name of the iframe.
height	pixels	Indicates the height dimension of the iframe.
width	pixels	Specifies the width dimension of the iframe.
src	URL of the file	Specifies the address or URL of the content to be embedded within the iframe.
sandbox	"" , allow-forms, allow-same-origin, allow-scripts, allow-top-navigation	Imposes restrictions on the contents within the iframe.
seamless	seamless	Indicates that the iframe should seamlessly integrate with the containing document.
srcdoc	HTML code	Represents the HTML content of a page to be displayed within the iframe.

Chapter 6: Embed Multimedia

Embed Multimedia:

HTML5 provides several elements to embed multimedia, such as <audio> and <video>. Use these elements to embed audio and video content.

```
<audio controls>
  <source src="audio.mp3" type="audio/mp3">
  Your browser does not support the audio tag.
</audio>

<video width="320" height="240" controls>
  <source src="video.mp4" type="video/mp4">
  Your browser does not support the video tag.
</video>
```

Use of Multimedia Attributes:

Multimedia elements in HTML5 come with various attributes for customization. Common attributes include controls, autoplay, loop, width, height, etc.

```
<video width="320" height="240" controls autoplay
loop>
  <source src="video.mp4" type="video/mp4">
  Your browser does not support the video tag.
</video>
```

Understanding Differences in Media Formats:

Different browsers support different media formats. Provide multiple sources with different formats to ensure compatibility.

```
<video width="320" height="240" controls>
  <source src="video.mp4" type="video/mp4">
  <source src="video.webm" type="video/webm">
  Your browser does not support the video tag.
</video>
```

Preloading, Autoplaying, and Looping:

Utilize attributes like preload, autoplay, and loop to control the loading, automatic playing, and looping of multimedia.

```
<video width="320" height="240" preload="auto" autoplay loop>
  <source src="video.mp4" type="video/mp4">
  Your browser does not support the video tag.
</video>
```

Displaying Subtitles and Captions:

Use the <track> element to add subtitles or captions to your multimedia content.

```
<video width="320" height="240" controls>
  <source src="video.mp4" type="video/mp4">
  <track kind="subtitles" src="subtitles_en.vtt" srclang="en"
  label="English">
  Your browser does not support the video tag.
</video>
```

Generating Embed Code with an Embed Code Builder:

Many online tools provide embed code builders. Input your multimedia details, and they generate the HTML code for you.

Creating a Load Progress Bar:

JavaScript and CSS can be used to create a load progress bar for multimedia elements.

Example:

```
<style>
  #progress-bar {
    width: 100%;
    height: 10px;
    background-color: #ddd;
  }

  #progress {
    width: 0;
    height: 100%;
    background-color: #4caf50;
  }
</style>

<div id="progress-bar">
  <div id="progress"></div>
</div>
```

```
<script>
  var video =
document.querySelector('video');
  var progressBar =
document.getElementById('progress');

  video.addEventListener('progress',
function() {
    var value = (video.buffered.end(0) /
video.duration) * 100;
    progressBar.style.width = value + '%';
  });
</script>
```

THE END of Module 1 HTML