

UNIT-II

HTML

Hyper Text Markup Language

Unit – 02 HTML

UNIT 02 HTML

Basics of HTML, formatting and fonts, commenting code, colour, hyperlink, lists, tables, images, forms, XHTML, Meta tags, Character entities, frames and frame sets, Browser architecture and Web site structure. Overview and features of HTML5

Static Website

- ❖ Static website contains Web pages with fixed content.
- ❖ Each web page is coded in HTML and displays the same information for every visitors.



Dynamic Website

- ❖ Dynamic websites contain web pages that are generated in real-time.
- ❖ **Example-** login & signup pages, application & submission forms, inquiry and shopping cart pages.

What is HTML?

- ❖ HTML stands for Hyper Text Markup Language.
- ❖ HTML originally designed for sharing scientific documents.
- ❖ Adaptations to HTML over the years have made it suitable to describe a number of other types of documents that can be displayed as web pages on the Internet.
- ❖ HTML is the standard markup language for creating web pages and describes the structure of a web page.
- ❖ HTML elements tell the browser how to display the content.
- ❖ HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.
- ❖ "The Language of Internet".

History of HTML?

- ❖ In 1980, Lee discovered HTML in Geneva. Berners Lee Invented the world wide web in 1989.
- ❖ HTML has been in continuous evolution since it was introduced to the Internet in the early 1990s by CERN: [The European Organization for Nuclear Research](#) and the IETF: [Internet Engineering Task Force](#)
- ❖ At present , the task of developing HTML is within an Organization “WORLD WIDE WEB CONSORTIUM”. This organization now takes care of HTML.
- ❖ First HTML version was form of SGML(Standard Generalized Markup Language). Text could be structured by this. This version also called HTML 1.0
- ❖ Second version of HTML was named by IETF ([INTERNET ENGINEERING TASK FORCE](#)) . This version was called as HTML 2.0 which was published in 1995. Some new features were added in which “Image Tag” was most important.
- ❖ HTML 3.0 was never published.

History of HTML?

Contd..

- ❖ An organization had already emerged until the publication of HTML1.0 and the recommendation of HTML3.0, which is made to work for the HTML language.
- ❖ It is known as W3C.
- ❖ In 1997 the next version of HTML was published by the W3C.
- ❖ HTML4.0- It include style sheet, Accessibility, Frames, Advanced Tables etc.
- ❖ HTML5.0 version which is extension to HTML4.0, and this version was published in 2012.

Evolution of HTML

Year	HTML Version
1990	HTML 1.0
1995	HTML 2.0
1997	HTML 3.2
1997	HTML 4.01
2000	XHTML 1.0
2012	HTML 5.0
2016	HTML 5.1
2017	HTML 5.2

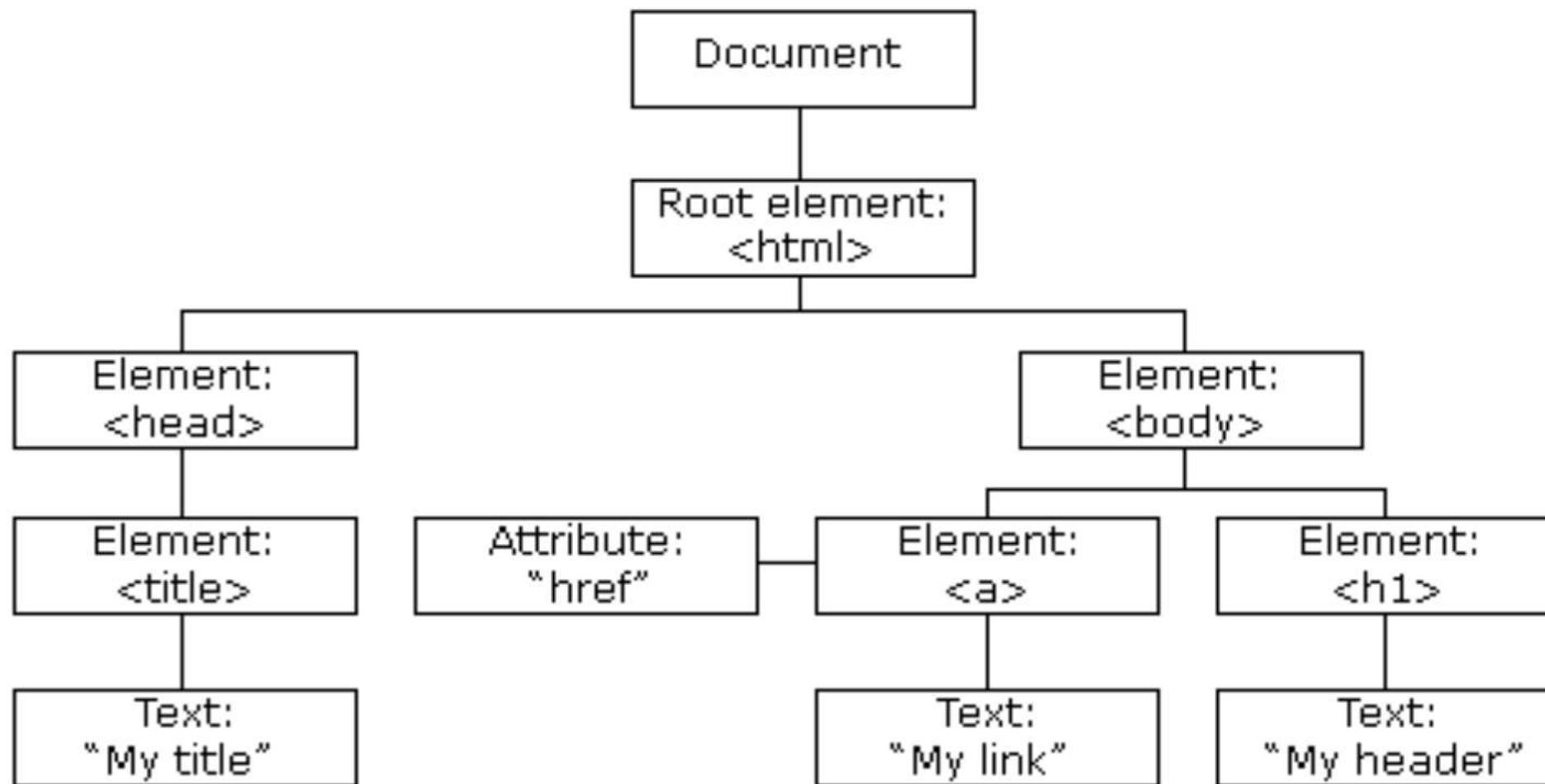
HTML5

- ❖ Define a single language which can be written in HTML syntax or XML syntax.
- ❖ New elements include video, nav, section, time, progress, canvas.
- ❖ The HTML5 specification defines a processing model that can interoperate with earlier HTML implementations.
- ❖ HTML5 improves the markup for documents.
- ❖ HTML5 includes markup and APIs for idioms, such as web storage, video, and audio content.

Features of HTML5

- ❖ Categories web pages into different sections-
 - Tools for data management, drawing, video and audio.
- ❖ Cross-browser applications for web and portable devices.
- ❖ Greater flexibility -
 - Exciting and interactive websites.
- ❖ Help to create a more engaging user experience.
 - HTML5 pages provide an experience similar to desktop applications.

HTML DOM Tree



HTML Documents

- ❖ All HTML documents must start with a document type declaration: <!DOCTYPE html>.
- ❖ The HTML document itself begins with <html> and ends with </html>.
- ❖ The visible part of the HTML document is between <body> and </body>.

Example-

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

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

The <!DOCTYPE> Declaration

- ❖ The <!DOCTYPE> declaration represents the document type and helps browsers to display web pages correctly.
- ❖ It must only appear once, at the top of the page (before any HTML tags).
- ❖ The <!DOCTYPE> declaration is not case sensitive.
- ❖ The <!DOCTYPE> declaration for HTML5 is:

<!DOCTYPE html>

What is an HTML Element?

- ❖ An HTML element is defined by a start tag, some content, and an end tag:
 - `<tagname>Content goes here...</tagname>`
- ❖ The HTML **element** is everything from the start tag to the end tag:
 - `<h1>My First Heading</h1>`
 - `<p>My first paragraph.</p>`

HTML Headings

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

Examples-

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
```

HTML Headings

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

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

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<h5>This is heading 5</h5>
<h6>This is heading 6</h6>

</body>
</html>
```

This is heading 1

This is heading 2

This is heading 3

This is heading 4

This is heading 5

This is heading 6

HTML Page Structure

```
<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>
```

Container and Empty Tags in HTML

- ❖ HTML uses predefined tags that tell the browser how to display the content.
Tags are nothing but some instructions that are enclosed in angle braces(< >).
- ❖ Tags are used in many places of the webpage but many users are often confused about some tags whether it is a container or an empty tag. They get this confusion because they don't know for what tag there should be an ending tag along with the opening tag or not.
- ❖ There are two types of tags in HTML:
 1. Container
 2. Empty

Container tags

- ❖ **Syntax:** <tag_name></tag_name>

- ❖ Some commonly used container tags are:
 1. **Essential Tags:** Following tags are used to create the structure of the webpage:
 - ❖ <html>....</html>: This marks the beginning and ending of the webpage also it tells that the document is an HTML document. This contains all other tags in between these tags which are considered for making a webpage.

Cont..

- ❖ **<head>...</head>**: This tag is used to define the head part of the document which contains the information related to the webpage.
- ❖ **<title>...</title>**: This tag stores the description of the web page, whatever given in these tags appears on the tab name while opened by the browser. It is described in the head tag.
- ❖ **<body>....</body>**: This tag is used to display all the information or data, i.e, text, images, hyperlinks videos, etc., on the webpage to the user.

Cont....

2. Headings: Following tags are used for headings:

- ❖ **<h1>....</h1> to <h6>...</h6>**: It is used for including headings of different sizes ranging from 1 to 6.

3. Text formatters: Following tags are used for text formatting:

- ❖ **<p>....</p>**: When paragraphs are needed to be included, this tag is used
- ❖ **....**: Makes the contained text to bold.
- ❖ **<i>...</i>**: Makes the contained text to italic.

4. HyperLinks: Following tag is used to define a hyperlink in the webpage:

- ❖ **...**: When we link some other web pages we add the hyper links to other webpages using this **<a ...>...** tag.

Cont..

5. Button tag: Following tag is used to create a click button:

- ❖ **<button>...</button>:** This is used in many ways but mainly used to manipulate dom by adding events and many more.

6. Division tag: Following tag is used to create a division:

- ❖ **<div>....</div>:** This defines a section in a document. The webpage can be divided to different sections using the **<div>....</div>** tag.

7. Iframe tag: Following tag is used for inline framing:

- ❖ **<iframe src="link.com> </iframe>:** When some other document is to be embedded like some video or image into HTML we use this tag.

8. Navigation tag: Following tag is used to set a navigation link:

- ❖ **<nav>...</nav>:** Defines a navigation bar that contains a set of menu or a menu of hyperlinks.

Cont..

9. Script tag: Following tag is used to add JavaScript code to the webpage:

- ❖ **<script>...</script>** : This contains the Javascript code that adds interactivity to the webpage.

10. Lists: Following tags are used to write data in the form of ordered and unordered lists:

...: This tag is used to create ordered lists.

- ❖ **...**: This tag is used to create unordered lists.

- ❖ **...**: This tag is used to add list items.

Empty Tags

Some commonly used empty tags are:

- ❖ **
**: Inserts a line break in a webpage wherever needed.
- ❖ **<hr>**: Inserts a horizontal line wherever needed in the webpage.
- ❖ ****: This tag is used to display the images on the webpage which were given in the src attribute of the tag.
- ❖ **<input>**: This is mainly used with forms to take the input from the user and we can also define the type of the input.
- ❖ **<link>**: When we store our CSS in an external file this can be used to link external files and documents to the webpage and it is mainly used to link CSS files.
- ❖ **<meta>**: Contains all metadata of the webpage. Metadata is the data about data and is described in the head tag.
- ❖ **<source>**: When an external media source is needed to be included in the webpage. source tag is used to insert any media source like audio, video etc... in our webpage.

Web Browsers

- ❖ The purpose of a web browser (Chrome, Edge, Firefox, Safari) is to read HTML documents and display them correctly.
- ❖ A browser does not display the HTML tags, but uses them to determine how to display the document:



HTML Paragraphs

- ❖ HTML paragraphs are defined with the `<p>` tag:

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

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

</body>
</html>
```

This is a paragraph.

This is another paragraph.

HTML Link tag

- ❖ Links are found in nearly all web pages. Links allow users to click their way from page to page.
- ❖ HTML links are hyperlinks.
- ❖ We can click on a link and jump to another document.
- ❖ When we move the mouse over a link, the mouse arrow will turn into a little hand
- ❖ A link does not have to be text.
- ❖ A link can be an image or any other HTML element

Cont..

Example

- ❖ 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

HTML Images tag

- ❖ HTML images are defined with the tag.

Following attributes are-

- ❖ **src:** The src stands for source. Every image has an src attribute which tells the browser where to find the image you want to display. The URL of the image provided points to the location where the image is stored.
- ❖ **alt:** If the image cannot be displayed then the alt attribute acts as an alternative description for the image. The value of the alt attribute is a user-defined text.
- ❖ **Setting width and height of Image:** The width and height attributes are used to specify the height and width of an image. The attribute values are specified in pixels by default.

HTML Link tag

- ❖ HTML links are defined with the `<a>` tag:
- ❖ The link's destination is specified in the `href` attribute.
- ❖ Attributes are used to provide additional information about HTML elements.

```
<!DOCTYPE html>

<html>
<body>
<h1>HTML Links</h1>
<p><a href=""> </a></p>
</body>
</html>
```

Example

```
<!DOCTYPE html>
<html>
<head>
  <title>Src ,Alt, width & height Attribute Example</title>
</head>
<body>
  <p>inserted image using <img> tag:</p>
  
</body>
</html>
```

Cont...

- ❖ **Adding titles to Image:** Along with the images, titles can also be added to images to provide further information related to the inserted image. For inserting a title, the title attribute is used.
- ❖ **Setting a border to Image:** By default, every picture has a border around it. By using the border attribute, the thickness of the border can be changed. A thickness of “0” means that there will be no border around the picture.
- ❖ **Aligning an Image:** By default, an image is aligned on the left side of the page, but it can be aligned to the centre or right using the align attribute.
- ❖ **Adding Image as a Link:** An image can work as a link with a URL embedded in it. It can be done by using the “img” tag inside an “a” tag.

Example

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

Example

```
<!DOCTYPE html>
<html>
<body>
<p>inserted image using <img> tag:</p>
  <a href="https://www.google.com">
     </a>
</body>
</html>
```

HTML Styles

- ❖ The <style> tag in HTML helps us to modify our text, viewed in the page.
- ❖ This modification includes changing font size, font family, font color etc.
- ❖ Now let's look at various attributes of style and what else the tag supports.

Syntax: <tagname style="property: value;">

1. HTML Font Family: The font family changes the font style of a text and can be used in any text writing tag like <p> or heading tag. These font families include all the names that you find in Microsoft Office or any other writing-based software.

Example

```
<html>
<head>
    <title>Font Family</title>
</head>
<body>
    <h1 style="font-family:cooper black;"> HTML Style Tag</h1>
    <h2 style="font-family:Cambria;"> HTML Style Tag</h2>
    <h3 style="font-family:algerian;"> Font Family </h3>
    <p style="font-family:Castellar;"> Font Family </p>
</body>
</html>
```

Cont..

2.HTML Font Size: The font size changes the size of a text and this can also be used in any text writing tag like <p> or heading tag. The units can be given in “%” or pixels or other units can also be included.

Example:

```
<html>
<head>    <title>Font Size</title>        </head>
<body>
    <h1 style="font-size:80%;">HTML Font Size1</h1>
    <h2 style="font-size:20px;"> HTML Font Size2</h2>
    <p style="font-size:30px;"> HTML Font Size3</p>
</body>
</html>
```

Cont..

3.HTML Font Color: The font color tag changes the color of a text and can be used in any text writing tag like <p> or heading tag. We can use both name of the colors or also the color codes that is mainly used in Photoshop. For various color codes- <http://htmlcolorcodes.com/>

Example:

```
<html>
<head>  <title>Font Color</title>  </head>
<body>
  <h1 style="color:red;">HTML Font Color</h1>
  <h2 style="color:#8CCEF9;"> HTML Font Color</h2>
  <h3 style="color:green;"> HTML Font Color</h3>
  <p style="color:#810CA6;"> HTML Font Color</p>
</body>
</html>
```

Cont..

4.HTML Text Align: The text alignment tag is used to change the alignment of a text including centre, left or right alignment. For example:

```
<html>
<head>
  <title>Text Align</title>
</head>
<body>
  <h1 style="text-align:left;">HTML Text Align</h1>
  <h2 style="text-align:center;"> HTML Text Align</h2>
  <p style="text-align:right;"> HTML Text Align</h2>
</body>
</html>
```

Implementing style in HTML

- ❖ Styles in HTML are basically rules that describe how a document will be presented in a browser. Style information can be either attached as a separate document or embedded in the HTML document.

There are 3 ways of implementing style in HTML :

- **Inline Style** : In this method, the style attribute is used inside the HTML start tag.
- **Embedded Style** : In this method, the style element is used inside the <head> element of the document.
- **External Style Sheet** : In this method the <link> element is used to point to an external CSS file.

Example of Inline Style

```
<!DOCTYPE html>
<html>
<body>
  <h1 style="color:Blue;font-size:25px;">
    Example of Inline Style
  </h1>
  <p style="color:red;">First paragraph </p>
  <p style="color:green;font-size:40px;">
    Second paragraph </p>
  <hr style="border-color:orange;">
</body>
</html>
```

Example of Embedded Style

```
<!DOCTYPE html>
<html lang="en">
<head>
<style type="text/css">
  body { background-color: powderblue; }

  h1 {
    color: black;
    font-family: arial;      }

  p {
    color: yellow;
    font-family: verdana;    }
</style> </head>
<body>  <h1>Example of Embedded Style</h1>
<p>First paragraph.</p>
</body> </html>
```

Example of Linking External Style Sheets

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" type="text/css"
        href="externalstyle.css">
</head>
<body>
  <h3>Example of Linking External Style Sheet</h3>
  <p>First paragraph.</p>
</body> </html>
```

Background Color

- ❖ The **background-color property** in CSS is used to specify the background color of an element. The background covers the total size of the element with padding and border but excluding margin. It makes the text so easy to read for the user.

Syntax: element { background-color property }

Property Values:

- ❖ **Color** : It defines the background color value or color codes. For example: A color name can be given as: “green” or HEX value as “#5570f0” or RGB value as “rgb(25, 255, 2)”.

Syntax : element { background-color: color_name; }

Example-1

```
<!DOCTYPE html>
<html>
<!--This line changes the color of background-->
<body style="background-color:pink">
  <h1 style="color:green;text-align:center;">
    IWT- HTML
  </h1>
  <h3 style="text-align:center;">
    How to change color of Background?
  </h3>
</body>
</html>
```

Example-2

```
<!DOCTYPE html>
<html>
  <head>
    <title>background-color property</title>
    <style>
      body { text-align:center; background-color:green; }
      h1 { color:white; background-color:blue; }
      h2 { color:white; background-color:black; }
    </style>
  </head>
  <body>
    <h1> IWT Background-Color</h1>
    <h2>background-color: color_name;</h2>
  </body> </html>
```

HTML Formatting Elements

❖ **Formatting elements were designed to display special types of text:**

- - Bold text
- - Important text
- <i> - Italic text
- - Emphasized text
- <mark> - Marked text
- <small> - Smaller text
- - Deleted text
- <ins> - Inserted text
- <sub> - Subscript text
- <sup> - Superscript text

Example

```
<!DOCTYPE html>
<html>
<body >
  <p> <b> This Text is Bold </b></p>
  <p><strong> This text is important! </strong></p>
  <p> <i> This Text is Italic </i></p>
  <p>You <em>have</em> to hurry up!</p>
  <p>Do not forget to practice <mark>HTML program</mark> today.</p>
  <p> This is <sub> Subscript</sub> and <sup> Superscript</sup></p>
</body> </html>
```

HTML Tables

- ❖ Tables represent tabular data.
- ❖ A table consists of one or several rows
- ❖ Each row has one or more columns
- ❖ Tables comprised of several core tags: <table></table>: begin / end the table
 - <tr></tr>: create a table row
 - <td></td>: create tabular data (cell)
- ❖ Tables should not be used for layout. Use CSS floats and positioning styles instead.

HTML Tables

- ❖ A table is an arrangement of data in rows and columns, or possibly in a more complex structure. Tables are widely used in communication, research, and data analysis.
- ❖ Tables are useful for various tasks such as presenting text information and numerical data.
- ❖ Tables can be used to compare two or more items in tabular form layout.
- ❖ Tables are used to create databases.

HTML Tables

- ❖ Start and end of a table

```
<table> ... </table>
```

- ❖ Start and end of a row

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

- ❖ Start and end of a cell in a row

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

HTML Tables – Example

```
<!DOCTYPE html>
<html> <body>
<table style="width:100%">
    <tr><td>Firstname</td>
        <td>Lastname</td>
        <td>Grade</th>      </tr>
    <tr><td>Priya</td>
        <td>Sharma</td>
        <td>9.4</td>       </tr>
    <tr><td>Arun</td>
        <td>Singh</td>
        <td>8.6</td>      </tr>
    </table>
</body> </html>
```

Adding a border to a 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.

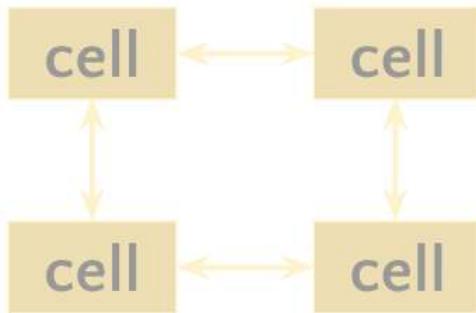
```
<head>
  <style>
    table, th, td {
      border: 1px solid black;    }
  </style> </head>

<body>
  <table style="width:100%">
    <tr>
      <th>Firstname</th>
      <th>Lastname</th>
      <th>Grade</th>
    </tr> </table>
```

Cell Spacing and Padding

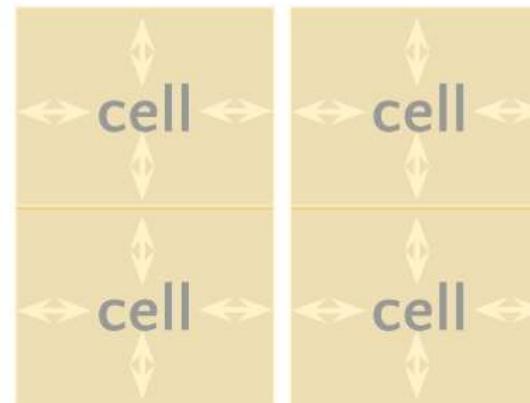
- ❖ Tables have two important attributes:

- ◆ cellspacing



- ◆ Defines the empty space between cells

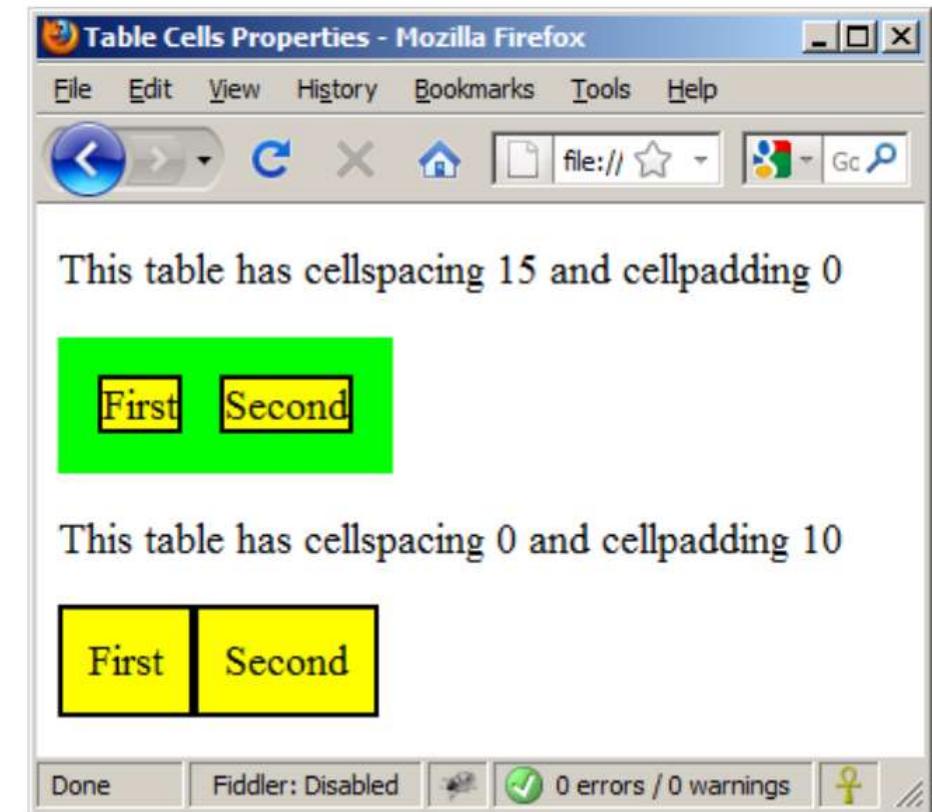
- ◆ cellpadding



- ◆ Defines the empty space around the cell content

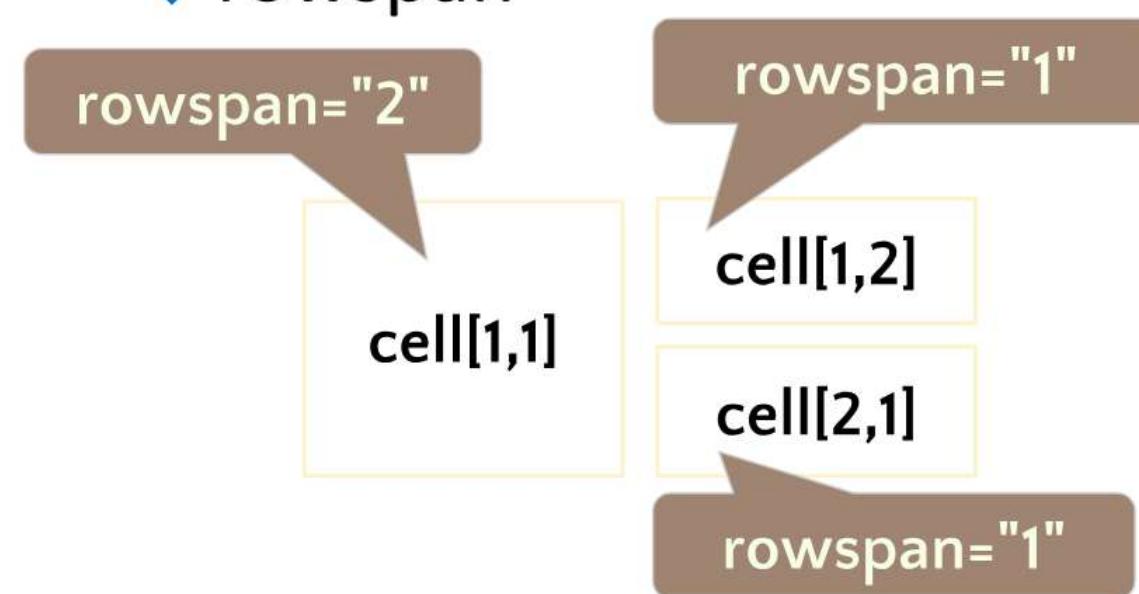
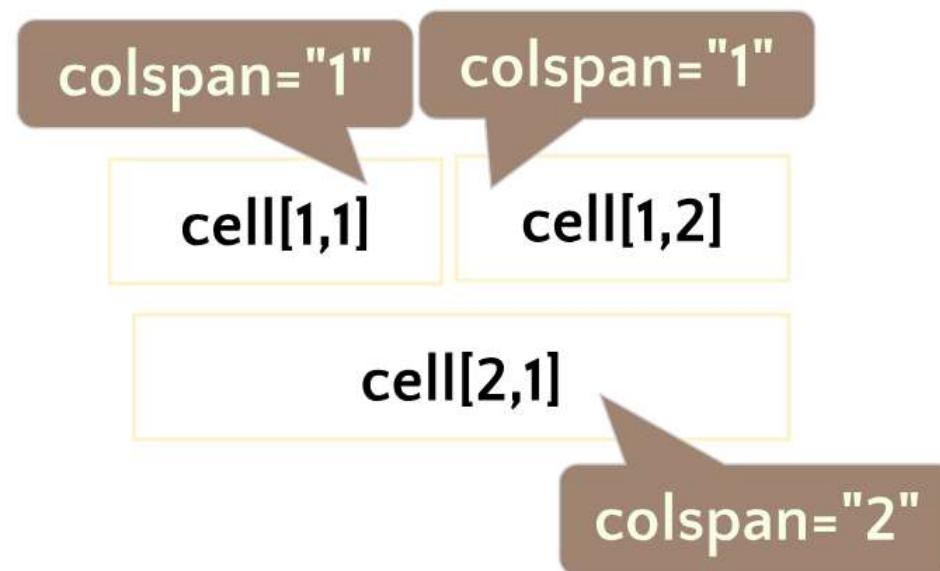
Cell Spacing and Padding – Example

```
<html>
  <head><title>Table Cells</title></head>
  <body>
    <table cellspacing="15" cellpadding="0">
      <tr><td>First</td>
      <td>Second</td></tr>
    </table>
    <br/>
    <table cellspacing="0" cellpadding="10">
      <tr><td>First</td><td>Second</td></tr>
    </table>
```



Column and Row Span

- ❖ Table cells have two important attributes:
 - ❖ colspan
 - ❖ rowspan



HTML Form Tag

- ❖ Forms are required to take input from the user who visits the website. This form is used basically for the registration process, logging into your profile on a website or to create your profile on a website, etc.
- ❖ Now the form will take input from the form and post that data in backend applications (like PHP). So the backend application will process the data which is received by them. There are various form elements that we can use like text fields, text area, drop-down list, select, checkboxes, radio, etc.

HTML Form Tag

- ❖ Forms are the primary method for gathering data from site visitors
- ❖ Create a form block with

```
<form> ..... </form>
```

- ❖ **Example:**

```
<form action = "Script URL" method = "GET|POST">  
    form elements like input, textarea etc. </form>
```

Attributes

There are many attributes that are associated with the <form> tag. Some of them are listed below:

- ❖ **Action Attribute**: -This is used to send the data to the server after the submission of the form.
- ❖ **Method**: -
 1. Get Method: -It has a limited length of characters of URL. -we should not use get to send some sensitive data. -This method is better for non-secure data.
 2. Post Method: -1. It has no size limitations 2. The submission of the form with the method post, can not be bookmarked.

Form Element

- ❖ There are various form elements available like
 - text fields,
 - textarea fields,
 - drop-down menus,
 - radio buttons,
 - Checkboxes
 - Fieldset

'Input' Element

```
<form method="post" action="http://server1">
    Enter your name:
    <input type="text" name="fname">
    <br/>
    Enter your age:
    <input type="text" name="age">
    <br/>
    <input type="submit" value="Submit">
</form>
```

'Select' Element

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title> HTML Form Select Attribute </title>  </head>
<body>
<form action="action-page.php">
<select name="Branch">
<option value="CSE"> CSE</option>
<option value="CSIT"> CSIT </option>
<option value="IT"> IT </option>
</select>
<input type="submit">
</form> </body>
```

'Textarea' Element

```
<!DOCTYPE html>
<html lang="en">
<head> <meta charset="UTF-8">
    <title> HTML Form Textarea Attribute </title> </head>
<body>
<h2>Textarea</h2>
<p>The textarea element defines a multi-line input field.</p>
<form action="action-page.php">
<textarea name="message" rows="5" cols="60"> This is a simple Example of
Textarea. </textarea>
<br>
<input type="submit">
</form> </body> </html>
```

Fieldsets

- ❖ Fieldsets are used to enclose a group of related form fields:

```
<form method="post" action="form.php">
  <fieldset>
    <legend>Client Details</legend>
    <input type="text" id="Name" />
    <input type="text" id="Phone" />
  </fieldset>
  <fieldset>
```

Form Input Controls

- ❖ Checkboxes:

```
<input type="checkbox" name="fruit" value="apple" />
```

- ❖ Radio buttons:

```
<input type="radio" name="title" value="Mr." />
```

- ❖ Radio buttons can be grouped, allowing only one to be selected from a group:

```
<input type="radio" name="city" value="Lom" />
<input type="radio" name="city" value="Ruse" />
```

Other Form Controls

❖ Dropdown menus:

```
<select name="gender">
  <option value="Value 1"
    selected="selected">Male</option>
  <option value="Value 2">Female</option>
  <option value="Value 3">Other</option>
</select>
```

❖ Submit button:

```
<input type="submit" name="submitBtn" value="Apply Now" />
```

Other Form Controls

- ❖ Reset button – brings the form to its initial state

```
<input type="reset" name="resetBtn" value="Reset the form" />
```

- ❖ Image button – acts like submit but image is displayed and click coordinates are sent

```
<input type="image" src="submit.gif" name="submitBtn" alt="Submit" />
```

- ❖ Ordinary button – used for Javascript, no default action

```
<input type="button" value="click me" />
```

Other Form Controls

- ❖ Password input – a text field which masks the entered text with * signs

```
<input type="password" name="pass" />
```

- ❖ Multiple select field – displays the list of items in multiple lines, instead of one

```
<select name="products" multiple="multiple">
  <option value="Value 1"
    selected="selected">keyboard</option>
  <option value="Value 2">mouse</option>
  <option value="Value 3">speakers</option>
</select>
```

Other Form Controls

- ❖ File input – a field used for uploading files

```
<input type="file" name="photo" />
```

- ❖ When used, it requires the form element to have a specific attribute:

```
<form enctype="multipart/form-data">
...
<input type="file" name="photo" />
...
</form>
```

Labels

- ❖ Form labels are used to associate an explanatory text to a form field using the field's ID.

```
<label for="fn">First Name</label>  
<input type="text" id="fn" />
```

- ❖ Clicking on a label focuses its associated field (checkboxes are toggled, radio buttons are checked)
- ❖ Labels are both a usability and accessibility feature and are required in order to pass accessibility validation.

Iframe Tag

- ❖ The iframe in HTML stands for **Inline Frame**.
- ❖ The “ iframe ” tag defines a rectangular region within the document in which the browser can display a separate document, including scrollbars and borders. An inline frame is used to embed another document within the current HTML document.
- ❖ Iframe is basically used to show a webpage inside the current web page.
- ❖ The ‘ **src** ’attribute is used to specify the URL of the document that occupies the iframe.

Syntax : '`<iframe src="URL"></iframe>`'

Attributes and their Examples

1. Height and Width:

The height and width attributes are used to specify the size of the iframe. The attribute values are specified in pixels by default, but they can also be specified in percentages like " 80% ".

Example

```
<!DOCTYPE html>
<html>
<body>
  <iframe src="www.google.com"
    height="300" width="400">
  </iframe>
</body>
</html>
```

HTML Block and Inline Elements

- ❖ Every HTML element has a default display value, depending on what type of element it is.
- ❖ There are two display values: block and inline.

1. Block-level Elements:

- ❖ A block-level element always starts on a new line.
- ❖ A block-level element always takes up the full width available (stretches out to the left and right as far as it can).
- ❖ A block level element has a top and a bottom margin, whereas an inline element does not.
- ❖ The <div> element is a block-level element.

Block-level elements in HTML

<address>

<article>

<aside>

<blockquote>

<canvas>

<dd>

<div>

<dl>

<dt>

<fieldset>

<figcaption>

<figure>

<footer>

<form>

<h1>-<h6>

<header>

<hr>

<main>

<nav>

<noscript>

<p>

<pre>

<section>

<table>

<tfoot>

<video>

Inline Elements

- ❖ An inline element does not start on a new line.
- ❖ An inline element only takes up as much width as necessary.
- ❖ This is a element inside a paragraph.
- ❖ An inline element cannot contain a block-level element!

inline elements in HTML:

| | |
|------------------------|-----------------------|
| <u><a></u> | <u><input></u> |
| <u><abbr></u> | <u><kbd></u> |
| <u><acronym></u> | <u><label></u> |
| <u></u> | <u><map></u> |
| <u><i></u> | <u><object></u> |
| <u><big></u> | <u><output></u> |
| <u>
</u> | <u><q></u> |
| <u><button></u> | <u><samp></u> |
| <u><cite></u> | <u><script></u> |
| <u><code></u> | <u><select></u> |
| <u></u> | <u><small></u> |
| | <u></u> |
| | <u></u> |

Div Tag (Division tag)

- ❖ The div tag is used in HTML to make divisions of content in the web page like (text, images, header, footer, navigation bar, etc). Div tag has both open(<div>) and closing (</div>) tag and it is mandatory to close the tag.
- ❖ The Div is the most usable tag in web development because it helps us to separate out data in the web page and we can create a particular section for particular data or function in the web pages.
- ❖ Div tag is Block level tag
- ❖ It is a generic container tag

Example

```
<html>
<head> <title>Division tag example</title>
<style type=text/css>
P{ background-color:gray;
    margin: 10px; }
div
{ color: white;
background-color: Blue;
margin: 2px;
font-size: 25px; }
</style> </head>
<body>    <div > division1</div>    <div > division2</div>
<div > division3</div>
</body></html>
```

What is HTML Canvas?

- ❖ The <canvas> tag in HTML is used to draw graphics on a web page using JavaScript. By default, it does not contain borders and text.
- ❖ The <canvas> element is only a container for graphics. You must use JavaScript to actually draw the graphics.
- ❖ Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

Syntax:

```
<canvas id = "script"> Contents... </canvas>
```

Cont...

- ❖ **Attributes:** The tag accepts two attributes as mentioned above and described below.
- ❖ **Height:** This attribute is used to set the height of the canvas.
- ❖ **Width:** This attribute is used to set the width of the canvas.

```
<!DOCTYPE html>
<html>  <body>
    <canvas id = "HTML Canvas Tag" width = "200"
            height = "100" style = "border:1px solid black">
    </canvas>
</body> </html>
```

Example

```
<!DOCTYPE html> <html> <head>
<meta charset="utf-8"/>
<script type="application/javascript">
function draw()
{ var canvas = document.getElementById('canvas');
  if (canvas.getContext)
  { var ctx = canvas.getContext('2d');
    ctx.fillStyle = 'rgb(200, 0, 0)';
    ctx.fillRect(10, 10, 50, 50);
    ctx.fillStyle = 'rgba(0, 0, 200, 0.5)';
    ctx.fillRect(30, 30, 50, 50);
  } } </script> </head>
<body onload="draw();">
<canvas id="canvas" width="150" height="150">
</canvas> </body> </html>
```

Audio Tag

- ❖ To embed audio in HTML, we use the <audio> tag. Before HTML5, audio can not be added to web pages in the Internet Explorer era.
- ❖ To play audio, we used web plugins like Flash.
- ❖ After the release of HTML5, it is possible. This tag supports Chrome, Firefox, Safari, Opera, and Edge in three audio formats – MP3, WAV, OGG. Only Safari browser doesn't support OGG audio format.

Syntax:

```
<audio>
    <source src="file_name" type="audio_file_type">
</audio>
```

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<h2>Click play button to play audio</h2>
<audio src="test.mp3" controls></audio>
</body>
</html>
```

Video Tag

- ❖ The <video> tag is used to embed video content in a document, such as a movie clip or other video streams.
- ❖ The <video> tag contains one or more <source> tags with different video sources. The browser will choose the first source it supports.
- ❖ The text between the <video> and </video> tags will only be displayed in browsers that do not support the <video> element.

Example

```
<!DOCTYPE html>
<html>
<body>
<p>Adding Video Tag</p>
<video width="400" height="350" controls>
<source src="a.mp4" type="video/mp4">
<source src="b.ogg" type="video/ogg">
</video>
</body>
</html>
```

Nav Tag

- ❖ The <nav> tag is used for declaring the navigational section in HTML documents. Websites typically have sections dedicated to navigational links, which enables users to navigate the site.
- ❖ These links can be placed inside a nav tag. In other words, the nav element represents a section of the page whose purpose is to provide navigational links, either in the current document or to another document.
- ❖ The links in the nav element may point to other webpages or to different sections of the same webpage. It is a semantic element. Common examples of the nav elements are menus, tables, contents, and indexes.

Example

```
<!DOCTYPE html>
<html>
<body>
    <h2> HTML nav Tag</h2>
    <!-- nav tag starts -->
    <nav>
        <a href="#">Home</a> |
        <a href="www.google.com">Google</a> |
        <a href="#">Languages</a> |
        <a href="#">Data Structure</a> |
        <a href="#">Algorithm</a>
    </nav>
    <!-- nav tag ends -->
</body> </html>
```

XHTML

- XHTML stands for Extensible Hyper Text Markup Language
- XHTML is a stricter, more XML-based version of HTML
- XHTML is HTML defined as an XML application
- XHTML is supported by all major browsers

The Most Important Differences from HTML

The Most Important Differences from HTML

- <!DOCTYPE> is mandatory
- The xmlns attribute in <html> is mandatory
- <html>, <head>, <title>, and <body> are mandatory
- Elements must always be properly nested
- Elements must always be closed
- Elements must always be in lowercase
- Attribute names must always be in lowercase
- Attribute values must always be quoted

Example

Here is an XHTML document with a minimum of required tags:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"  
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
    <title>Title of document</title>  
</head>  
<body>  
some content here...  
</body>  
</html>
```

XHTML Elements Must Always be Closed

In XHTML, elements must always be closed, like this:

Correct:

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

Wrong:

```
<p>This is a paragraph
<p>This is another paragraph
```

XHTML Elements Must be in Lowercase

In XHTML, element names must always be in lowercase, like this:

Correct:

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

Wrong:

```
<BODY>
<P>This is a paragraph</P>
</BODY>
```

HTML <meta> Tag

Example

Describe metadata within an HTML document:

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

More Examples

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 for HTML and CSS">
```

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">
```

HTML Character Entities

Reserved characters in HTML must be replaced with entities:

- < (less than) = **<**;
- > (greater than) = **>**;

To display a less than sign (<) we must write: **<** or **<**

Entity names are easier to remember than entity numbers.

Some Useful HTML Character Entities

Result	Description	Name	Number
	non-breaking space	 	
<	less than	<	<
>	greater than	>	>
&	ampersand	&	&
"	double quotation mark	"	"
'	single quotation mark	'	'
¢	cent	¢	¢
£	pound	£	£
¥	yen	¥	¥
€	euro	€	€
©	copyright	©	©
®	trademark	®	®

HTML <frameset> Tag

Not Supported in HTML5.

The `<frameset>` tag was used in HTML 4 to define a frameset.

What to Use Instead?

Example

Use the `<iframe>` tag to embed another document within the current HTML document:

```
<iframe src="https://www.w3schools.com"></iframe>
```

Web Browsers

- A web browser or Internet browser is a **software application** for **retrieving, presenting, and traversing** information resources on the **World Wide Web**.
- Example :



Mozilla Firefox



Google Chrome



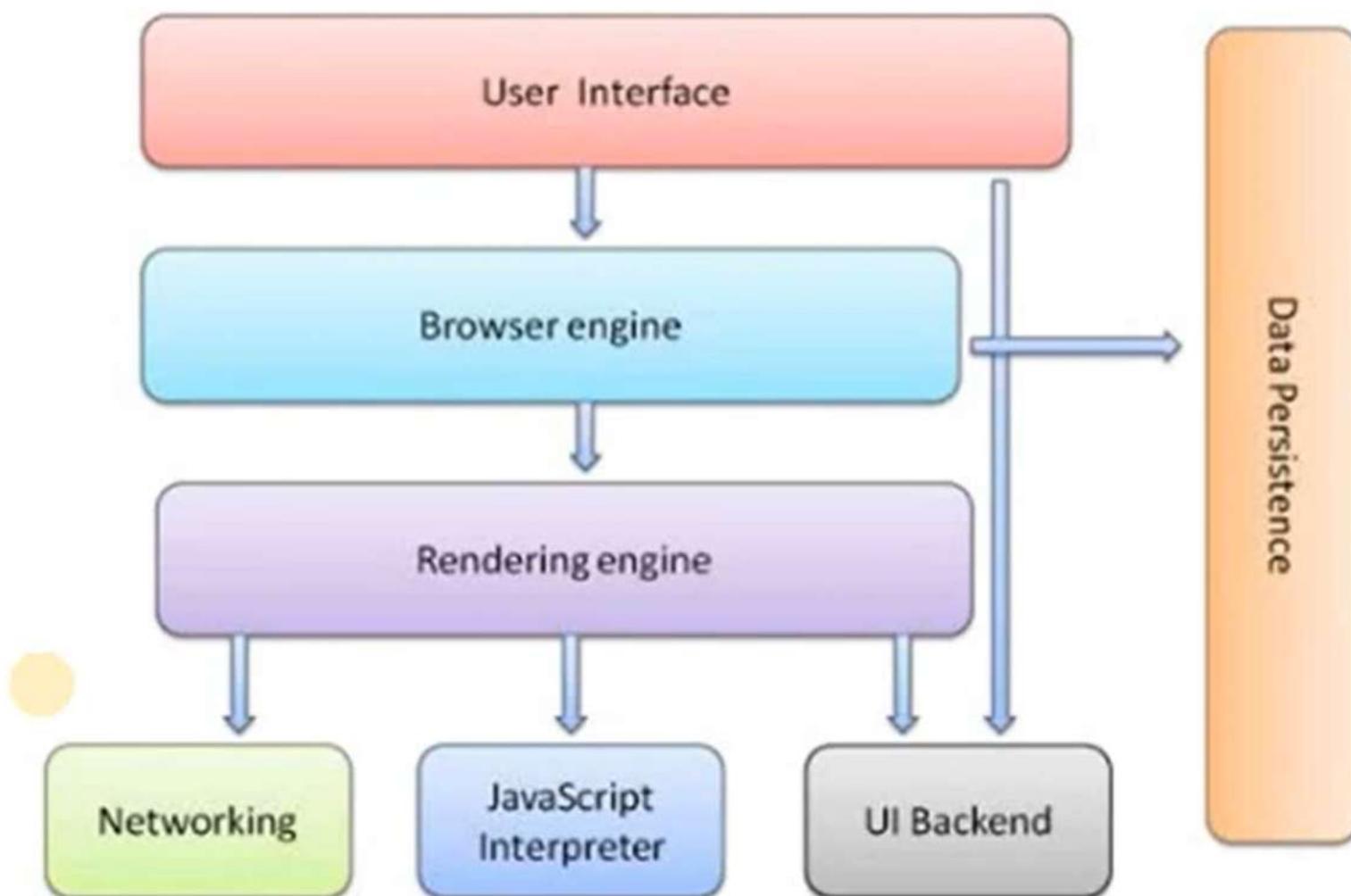
Opera



Internet Explorer

- An *information resource* is identified by a Uniform Resource Identifier (**URI**) and may be a web page, image, video, or other **piece** of content.
- Hyperlinks present in resources enable users to easily navigate their browsers to related resources

Architecture of Web Browser



Component of Web Browser

The User Interface:

- Its space where User interacts with the browser.
- It includes the address bar, back and next buttons, home button, refresh and stop, bookmark option, etc.

The Browser Engine:

- works as a bridge between the User interface and the rendering engine.
- According to the inputs from various user interfaces, it queries and manipulates the rendering engine.

Component of Web Browser

The Rendering Engine:

- It's responsible for rendering the requested web page on the browser screen.
- It interprets the HTML, XML documents and images that are formatted using CSS and generates the layout that is displayed in the User Interface.
- Different browsers use different rendering engines.

Networking:

- retrieves the URLs using the common internet protocols of HTTP or FTP.
- Handles all aspects of Internet communication and security.

Component of Web Browser

JavaScript Interpreter:

- interprets and executes the javascript code embedded in a website.
- The interpreted results are sent to the rendering engine for display.

UI Backend:

- UI backend is used for drawing basic widgets like combo boxes and windows.
- It underneath uses operating system user interface methods.

Component of Web Browser

Data Persistence/Storage:

- Browsers support storage mechanisms such as localStorage, IndexedDB, WebSQL and FileSystem.
- It is a small database created on the local drive of the computer where the browser is installed.
- It manages user data such as cache, cookies, bookmarks and preferences.

An important thing to note here is that in web browsers such as Google Chrome each tab runs in a separate process(multiple instances of rendering engine).

THANKS