

Q: Differences b/w HTML and XHTML.

## HTML

## XHTML

- |   |   |
|---|---|
| ① HTML stands for HyperText Markup lan <sup>m</sup> . | XHTML stands for Extensible HyperText Markup lan <sup>m</sup> . |
| ② It was developed in 1991.                           | It was released in 2000.  |
| ③ It is extended form of SGML.                        | It is extended form of XML and HTML.                            |
| ④ It is not case sensitive.                           | It is <del>case</del> case sensitive.                           |
| ⑤ Do not require a closing tag.                       | Require a closing tag.  |
| ⑥ Extension: .html or .htm                            | Extension: .xhtml or .xht or .xml                               |

Q: diff b/w inline and block element with example.

Q: What are empty elements in HTML briefly explain it with example.

$\Rightarrow$  Those elements of HTML that cannot have closing tag and cannot have some content like text or child element inside them.

Example  $\rightarrow <\text{img}>$ ,  $<\text{br}>$ ,  $<\text{input}>$ ,  $<\text{hr}>$

$<\text{meta}>$ ,  $<\text{link}>$

$<\text{p}>$  Hi I am Raahum  $<\text{br}>$  Kumar  $<\text{p}>$

Q In how many ways can you integrate CSS on a web page? discuss with the help of programming examples.

① **Inline CSS:** We can apply CSS directly within HTML element using the 'style' attribute.

Example :-

```
<p style="color: blue; font-size: 16px;">
This text is inline styled. </p>
```

② **Internal (or Embedded) :** We can include style within the '`<style>`' tag in the head document '`<head>`' section.

~~<!DOCTYPE~~

```
<!DOCTYPE html>
<html>
<head>
<style>
P { color: pink;
    font-size: 16px;
}
</style>
</head>
<body>
<p>This text is styled internally. </p>
</body>
</html>
```

③ External CSS: We can create a separate CSS file and link it to your HTML document using the `<link>` element.

`style.css:`

```
pre color: pink;
font-size: 20px;
```

3

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="style.css">
</head>
<body>
<p>This is externally styled </p>
</body>
</html>
```

Q Briefly explain the ordered and unordered lists in HTML with suitable example. Also explain how can you change the type of list and control the list.

Ans → In HTML ordered lists `<ol>` are used to create numbered lists, while unordered list (`<ul>`) create bulleted lists.

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

unordered list:

<ul>

<li> Apple </li>

<li> Banana </li>

<li> Orange </li>

</ul>

To change the type of the list item marker in unordered list, we can use 'type' attribute within the '<ul>' tag  
Ex → changing into square markers.

<ul type = "square">

<li> Item 1 </li>

<li> Item 2 </li>

<li> Item 3 </li>

</ul>

To change the type of list item marker in an ~~un~~ordered list, you can use the 'type' attribute within the '<ul>' tag

Ex → changing it to ~~square~~ counting to uppercase Roman numerals.

<ol type = "I">

<li> Item A </li>

<li> Item B </li>

<li> Item C </li>

</ol>

(8M) Create a feedback form in HTML and demonstrate the use of various form elements like text field, radio buttons, check boxes, text area and submit button. Also apply form validation on any two fields using javascript.

```

<!DOCTYPE html>
<html lang="en">
<head>
<title>Feedback Form</title>
<style> label { display: block; }
</style>
</head>
<body>
<h1>Feedback Form</h1>
<form>
<label for="name">Name:</label>
<input type="text" id="name" name="name" />
<label for="email">Email:</label>
<input type="email" id="email" name="email" />
<label> Rating:</label>
<label for="excellent">Excellent </label> value="excellent"
<input type="radio" id="excellent" name="rating" value="excellent" checked="" />
<label for="good">Good </label>
<input type="radio" id="good" name="rating" value="good" />
<label for="average">Average </label>
<input type="radio" id="average" name="rating" value="average" />
<label for="bad">Bad </label>
<input type="radio" id="bad" name="rating" value="bad" />

```

```
<label for="comment">Comments </label>
<textarea id="comment" name="Comments" rows="4"
cols="50"></textarea>
<label for="subscribe">Subscribe to newsletter </label>
<input type="checkbox" id="subscribe" name="subscribe"
value="Yes">
<input type="submit" value="Submit">
</form>
</body>
</html>
```

### Q Difference b/w GET and POST methods in html

#### GET

- i Send data as part of the URL
- ii visible in the address bar.
- iii suitable for less sensitive information
- iv use → retrieving data

#### POST

- Send data in the request body
- Not visible in the URL.
- Suitable for large sensitive information.
- use → submitting data

### Q How tables are created in HTML? What are the various tags used during table?

Ans → To create tables in HTML we can use a combination of the `<table>`, `<tr>`, `<th>`, `<td>` tags

SAGAR  
Page No. \_\_\_\_\_  
Date \_\_\_\_\_

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>table</title>
</head>
<body>
  <table border="1">
    <tr>
      <th>Name </th>
      <th>URN </th>
      <th>CRN </th>
    </tr>
    <tr>
      <td>RAUSHAN </td>
      <td>9203751 </td>
      <td>9221139 </td>
    </tr>
    <tr>
      <td>[REDACTED] JAY </td>
      <td>9203752 </td>
      <td>9221139 </td>
    </tr>
  </table>
</body>
</html>
```

\* <table> : It represents the entire table.  
\* <tr> : It represents a row within table.  
\* <th> : It represents header cell of first row.  
\* <td> : It represents data cell within table.

Q diff b/w inline and block element with example  
|  
Inline element      |      Block element

i) It occupies only necessary width.

② Example: <span>, <a> <strong><em>	<div> <p> <h1>toch6> <ul> <li>
--	-----------------------------------

③ Does not start on a new line starts on a new line

④ only as wide as contents  
fill width of the parent container

## ⑤ Syntax:

`<span> This is inline ele </span> <div>This is a block ele </div>`

## \* Description list

400

$\langle \text{eff} \rangle_{\text{Term 1}} < \langle \text{eff} \rangle$

<dd>Description 1</dd>

$\Leftarrow$  Term 2  $\Leftarrow$

<dd> description 2 </dd>

$\angle 1 \text{ or } >$

### Difference between HTML and XHTML :

S.No.	HTML	XHTML
1.	HTML stands for Hypertext Markup Language.	XHTML stands for Extensible Hypertext Markup Language.
2.	It was developed by Tim Berners-Lee.	It was developed by W3C i.e World Wide Web Consortium.
3.	It was developed in 1991.	It was released in 2000.
4.	It is extended from SGML.	It is extended from XML and HTML.
5.	The format is a document file format.	The format is a markup language.
6.	All tags and attributes are not necessarily to be in lower or upper case.	In this, every tag and attribute should be in lower case.
7.	Doctype is not necessary to write at the top.	Doctype is very necessary to write at the top of the file.
8.	It is not necessary to close the tags in the order they are opened.	It is necessary to close the tags in the order they are opened.
9.	While using the attributes it is not necessary to mention quotes. For e.g. <Geeks>.	While using the attributes it is mandatory to mention quotes. For e.g. <Geeks="GFG">.
10.	Filename extension used are .html, .htm.	Filename extension are .xhtml, .xht, .xml.

### Difference between HTTP GET and HTTP POST

HTTP GET	HTTP POST
In GET method we can not send large amount of data rather limited data of some number of characters is sent because the request parameter is appended into the URL.	In POST method large amount of data can be sent because the request parameter is appended into the body.

<b>HTTP GET</b>	<b>HTTP POST</b>
GET request is comparatively better than Post so it is used more than the Post request.	POST request is comparatively less better than Get method, so it is used less than the Get request.
GET requests are only used to request data (not modify)	POST requests can be used to create and modify data.
GET request is comparatively less secure because the data is exposed in the URL bar.	POST request is comparatively more secure because the data is not exposed in the URL bar.
Request made through GET method are stored in Browser history.	Request made through POST method is not stored in Browser history.
GET method request can be saved as bookmark in browser.	POST method request can not be saved as bookmark in browser.
Request made through GET method are stored in cache memory of Browser.	Request made through POST method are not stored in cache memory of Browser.
Data passed through GET method can be easily stolen by attackers as the data is visible to everyone.GET requests should never be used when dealing with sensitive data	Data passed through POST method can not be easily stolen by attackers as the URL Data is not displayed in the URL
In GET method only ASCII characters are allowed.	In POST method all types of data is allowed.

How table is created in HTML? What are the various tags used during table?

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <table border="black">
        <caption>Food Menu </caption>
        <thead>
            <tr>
                <th rowspan="2">Item</th>
                <th colspan="2">Price</th>
            </tr>
            <tr>
                <th>INR</th>
                <th>USD</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>Frooti</td>
                <td>10</td>
                <td>0.14</td>
            </tr>
            <tr>
                <td>Samosa</td>
                <td>20</td>
                <td>0.64</td>
            </tr>
            <tr>
                <td>Chips</td>
                <td>80</td>
                <td>0.94</td>
            </tr>
        </tbody>
    </table>
</body>
</html>
```

Food Menu		
Item	Price	
	INR	USD
Frooti	10	0.14
Samosa	20	0.64
Chips	80	0.94

## HTML Table Tags

Tag	Description
<a href="#"><u>&lt;table&gt;</u></a>	Defines a table
<a href="#"><u>&lt;th&gt;</u></a>	Defines a header cell in a table
<a href="#"><u>&lt;tr&gt;</u></a>	Defines a row in a table
<a href="#"><u>&lt;td&gt;</u></a>	Defines a cell in a table
<a href="#"><u>&lt;caption&gt;</u></a>	Defines a table caption
<a href="#"><u>&lt;colgroup&gt;</u></a>	Specifies a group of one or more columns in a table for formatting
<a href="#"><u>&lt;col&gt;</u></a>	Specifies column properties for each column within a <colgroup> element
<a href="#"><u>&lt;thead&gt;</u></a>	Groups the header content in a table
<a href="#"><u>&lt;tbody&gt;</u></a>	Groups the body content in a table
<a href="#"><u>&lt;tfoot&gt;</u></a>	Groups the footer content in a table

## Some of the most important applications of AJAX are as follows.

- Updating a webpage without reloading the page.
- Requesting data from the server after the page has been loaded.
- Receiving data from the server after the page has been loaded.
- Sending data to the server in the background without disturbing UI or other processes.
- View an XML CD catalog.
- Display XML data in an HTML table.
- Show XML data inside an HTML div element.
- Navigate through XML nodes.
- A simple CD catalog application.

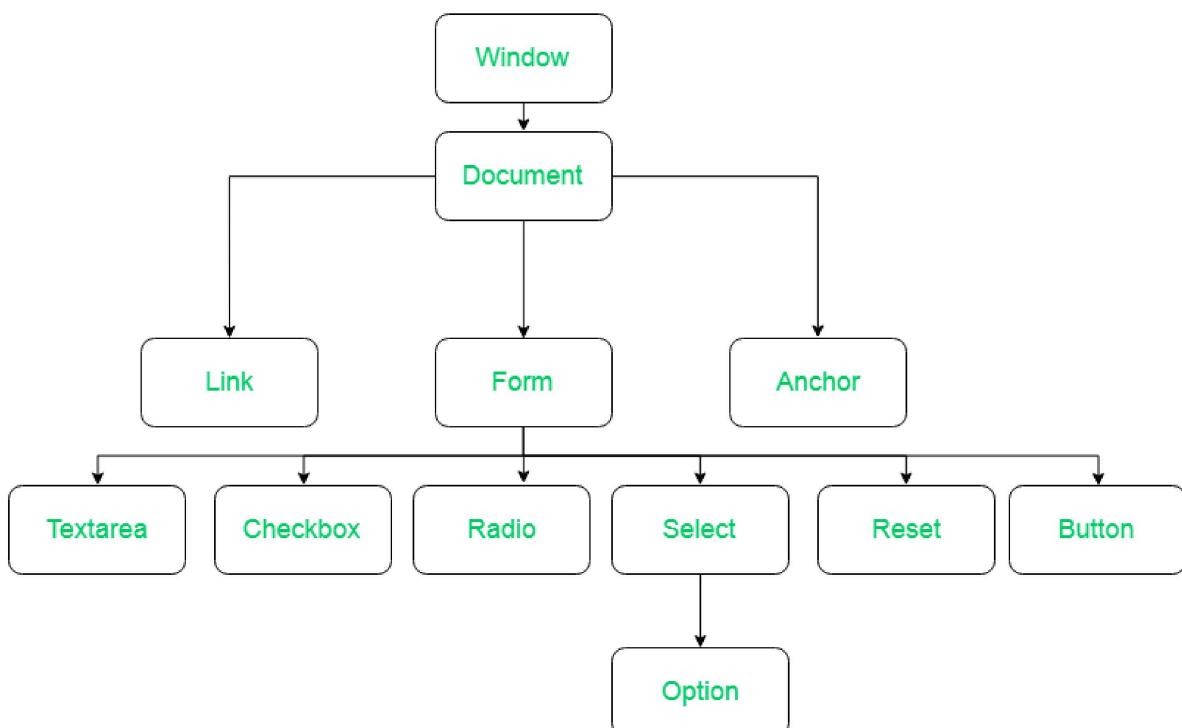
## What is the HTML DOM?

**The HTML DOM is a standard for how to get, change, add, or delete HTML elements.**

The Document Object Model (DOM) is a programming interface for HTML(HyperText Markup Language) and XML(Extensible markup language) documents. It defines the logical structure of documents and the way a document is accessed and manipulated.

Note: It is called a Logical structure because DOM doesn't specify any relationship between objects.

DOM is a way to represent the webpage in a structured hierarchical way so that it will become easier for programmers and users to glide through the document. With DOM, we can easily access and manipulate tags, IDs, classes, Attributes, or Elements of HTML using commands or methods provided by the Document object. Using DOM, the JavaScript gets access to HTML as well as CSS of the web page and can also add behavior to the HTML elements. so basically Document Object Model is an API that represents and interacts with HTML or XML documents.



## Why DOM is required?

HTML is used to structure the web pages and Javascript is used to add behavior to our web pages. When an HTML file is loaded into the browser, the javascript can not understand the HTML document directly. So it interprets and interacts with the Document Object Model (DOM), which is created by the browser based on the HTML document. DOM is basically the representation of the same HTML document but in a tree-like structure composed of objects. Javascript interprets DOM easily, using it as a bridge to access and manipulate the elements i.e javascript can not understand the tags(<h1>H</h1>) in HTML document but can understand object h1 in DOM. Now, Javascript can access each of the objects (h1, p, etc) by using different functions.

### Discuss various Sectors in JQuery with example.

SR No.	Selector	Example	Description
1.	<u>*</u>	<code>\$("")</code>	All elements are selected
2.	<u>#id</u>	<code>\$("#roll_no")</code>	The element with id="roll_no" is selected.
3.	<u>.class</u>	<code>\$(".name")</code>	All elements with class "name" are selected
4.	<u>.class, .class</u>	<code>\$(".name, .surname")</code>	It will select all elements with the class "name" or "surname"
5.	<u>element</u>	<code> \$("p")</code>	It will select all p elements.
6.	<u>:first</u>	<code> \$("p:first")</code>	The first p element is selected.
7.	<u>:last</u>	<code> \$("p:last")</code>	The last p element is selected.
8.	<u>:first-child</u>	<code> \$("p:first-child")</code>	All p elements that are the first child of their parent are selected.
9.	<u>:last-child</u>	<code> \$("p:last-child")</code>	All p elements that are the last child of their parent are selected.

10.	<a href="#"><u>only-child</u></a>	<code>\$("p:only-child")</code>	All p elements that are the only child of their parent are selected
11.	<a href="#"><u>:header</u></a>	<code>\$(":header")</code>	All header elements get selected.
12.	<a href="#"><u>:hidden</u></a>	<code>\$("table:hidden")</code>	All hidden p elements are selected.
13.	<a href="#"><u>:animated</u></a>	<code>\$(":animated")</code>	All animated elements are selected.
14.	<a href="#"><u>:root</u></a>	<code>\$(":root")</code>	The document's root element will be selected
15.	<a href="#"><u>:focus</u></a>	<code>\$(":focus")</code>	The element that currently has focus is selected.

## What is HTML?

- HTML stands for Hyper Text Markup Language.
- HTML is the standard markup language for creating Web pages.
- HTML describes the structure of a Web page.
- HTML elements tell the browser how to display the content

## What does the standard markup language means?

A standard markup language uses tags or codes to format text and convey information about its structure. HTML is a well-known example widely used on the web.

## HTML Page 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 it defines the whole HTML document.
- The `<head>` element contains meta information about the HTML page
- { The `<head>` element in HTML contains metadata or information about the HTML page, such as title, character set, linked stylesheets, and scripts.}

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial
  scale=1.0">
  <title>My Webpage</title>
  <link rel="stylesheet" href="styles.css">
  <script src="script.js" defer></script>
</head>
```

- 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.

## 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>**

**Note:** Some HTML elements have no content (like the `<br>` element). These elements are called empty elements. Empty elements do not have an end tag!

Year	Version
1989	Tim Berners-Lee invented www
1991	Tim Berners-Lee invented 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:

## HTML Links

HTML links are defined with the `<a>` tag:

```
<a href="https://www.w3schools.com">This is a link</a>
```

## HTML Images

HTML images are defined with the `<img>` tag.

The source file (`src`), alternative text (`alt`), `width`, and `height` are provided as attributes:

```

```

**NOTE:** HTML is Not Case Sensitive.

## HTML Attributes

HTML attributes provide additional information about HTML elements.

Attributes usually come in name/value pairs like: **name="value"**

### Example href,src.

There are two ways to specify the URL in the **src** attribute:

1. **Absolute URL** - Links to an external image that is hosted on another website.
2. **Relative URL** - Links to an image that is hosted within the website.

## The alt Attribute

The required **alt** attribute for the **<img>** tag specifies an alternate text for an image, if the image for some reason cannot be displayed. This can be due to a slow connection



## The style Attribute

The **style** attribute is used to add styles to an element, such as color, font, size, and more. Like **<p style="color:red;">This is a red paragraph.</p>**

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

<h2 title="I'm a header">The title
Attribute</h2>

<p title="I'm a tooltip">Mouse over this
paragraph, to display the title attribute
as a tooltip.</p>

</body>
</html>
```

## The title Attribute

I'm a header  
Mouse over this paragraph, to display the title attribute as a tooltip.

## The title Attribute

The **title** attribute defines some extra information about an element.

The value of the title attribute will be displayed as a tooltip when you mouse over the element:

## Chapter Summary

- All HTML elements can have **attributes**

- The `href` attribute of `<a>` specifies the URL of the page the link goes to
- The `src` attribute of `<img>` specifies the path to the image to be displayed
- The `width` and `height` attributes of `<img>` provide size information for images
- The `alt` attribute of `<img>` provides an alternate text for an image
- The `style` attribute is used to add styles to an element, such as color, font, size, and more
- The `lang` attribute of the `<html>` tag declares the language of the Web page
- The `title` attribute defines some extra information about an element
- you can specify the size for any heading with the `style` attribute, using the CSS `font-size` property: EXAMPLE `<h1 style="font-size:60px;">Heading 1</h1>`

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

“ h4 heading tag is equivalent to paragraph in html wrt size”

- A paragraph always starts on a new line, and is usually a block of text.
- The `<hr>` tag is an empty tag, which means that it has no end tag.
- The `<hr>` element is used to separate content (or define a change) in an HTML page.
- The `<br>` tag is an empty tag, which means that it has no end tag.

`<pre>` tag do not eliminate spaces and next line like `<p>` tag.

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

<p>The pre tag preserves both spaces and line breaks:</p>

<pre>
My Bonnie lies over the ocean.
My Bonnie lies over the sea.
My Bonnie lies over the ocean.
Oh, bring back my Bonnie to me.
</pre>

</body>
</html>
```

The pre tag preserves both spaces and line breaks:

```
My Bonnie lies over the ocean.
My Bonnie lies over the sea.
My Bonnie lies over the ocean.
Oh, bring back my Bonnie to me.
```

NOTE:

Tag	Description
<code>&lt;p&gt;</code>	Defines a paragraph
<code>&lt;hr&gt;</code>	Defines a thematic change in the content
<code>&lt;br&gt;</code>	Inserts a single line break
<code>&lt;pre&gt;</code>	Defines pre-formatted text

The HTML Style Attribute syntax:

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

- Use the `style` attribute for styling HTML elements
- Use `background-color` for background color
- Use `color` for text colors
- Use `font-family` for text fonts
- Use `font-size` for text sizes
- Use `text-align` for text alignment

codingcomb