

Agenda

- Introduction to HTML
- Tags
 - Textual tags
 - Formatting tags
 - List tag
 - Table tag
 - Resource tags
 - Anchor tag
 - Form tag
- GET vs POST method
- ~~Header, Footer, section, article, aside, nav~~ Tags

History of the First Browser and HTML

Tim Berners-Lee's (Inventor of the World Wide Web) Vision

- In March 1989, Tim Berners-Lee wrote a proposal for a system to manage information at CERN, which he called the World Wide Web.
- By the end of 1990, he had developed the code for the first web server, web browser, and editor.
- WorldWideWeb (Nexus) was first web browser
- It served as both a browser and an editor, allowing users to view and create web content.
- The name was later changed to Nexus to differentiate it from the World Wide Web information space.

Browser History

- WorldWideWeb, later named Nexus, was the sole browser for the web until 1992 when Lynx was introduced.
- Lynx, introduced in 1992, was and is a text-based browser. Used primary for locating, reading, and downloading text-based articles, it continues to be used today for those purposes.
- Mosaic, an early version of Netscape Navigator, was introduced in 1993. It was the first web browser with a graphical user interface, making the web more accessible to the masses.
- Netscape Navigator, a renamed and improved version of Mosaic, was introduced in 1994.
- Microsoft's Internet Explorer 1.0 was introduced in 1995.

HTML

- HTML (HyperText Markup Language) is the standard language used to create and structure content on the web.
- It provides the foundation for web pages by defining the structure and elements of the content, such as headings, paragraphs, links, images, tables, and more.
- The latest standard is HTML5
- To add html 5 code, start the document with

```
<!DOCTYPE html>
```

- DOCTYPE: document type (tag used to start the html document)
- It is Case in-sensitive
- To add comment

```
<!-- comment -->
```

Key Features of HTML

1. Markup Language: HTML uses "tags" to mark up different parts of content, indicating their roles (e.g., heading, paragraph, list).
2. HyperText: Enables linking to other web pages or resources using hyperlinks (`<a>` tag).
3. Platform-Independent: HTML can be rendered on any device with a web browser.
4. Extensible: Can be combined with other technologies like CSS (for styling) and JavaScript (for interactivity).

HTML History

1. HTML 1.0 (1993)

- Tim Berners-Lee, the inventor of the World Wide Web.
- Purpose was to share research documents and enable hyperlinking between them.
- Key Features:
 - Basic text formatting (headings, paragraphs, lists).
 - Hyperlinks (`<a>` tag).

2. HTML 2.0 (1995)

- Standardize HTML for broader use.
- Key Features:
 - Support for tables and forms.
 - Basic image embedding (`` tag).
 - Standardized syntax rules.

3. HTML 3.2 (1997)

- Managed by the World Wide Web Consortium (W3C).
- Introduced features for richer content.
- Key Features:
 - Support for scripting languages like JavaScript.
 - Improved table support for better layouts.
 - Introduction of new elements like for styling.

4. HTML 4.01 (1999)

- W3C continued to refine HTML.
- Enhanced web functionality and accessibility.
- Key Features:
 - Separation of content and style via CSS.

- Introduction of attributes like id and class for better styling and scripting.
- Improved support for forms and multimedia.

5. XHTML 1.0 (2000)

- W3C attempted to merge HTML with XML.
- Introduced stricter syntax and better compatibility.
- Key Features:
 - All tags must be properly closed (e.g., `
`).
 - Case sensitivity for tags and attributes.
- Challenges:
 - Too strict for practical use; adoption was limited.
 - Developers found it difficult to migrate from HTML 4.

6. HTML5 (2014, Official Recommendation)

- Modernize HTML for web applications and multimedia.
- Key Features:
 - Semantic Elements: `<header>`, `<footer>`, `<article>`, `<nav>` for better content structure.
 - Multimedia Support: Native audio (`<audio>`) and video (`<video>`) elements.
 - Canvas and SVG: For drawing graphics and animations directly in the browser.
 - APIs: Built-in APIs for features like geolocation, drag-and-drop, and offline storage.
 - Backward Compatibility: Maintains support for older HTML versions.
- It provided Simplified coding practices.

Tags

- Word enclosed by `<` and `>` signs
- It is also called as an element
- All tags in HTML are pre-defined by W3C
- E.g. `<h1>`, `<p>`, `<table>`

Types of tags

1. Opening

- Used to open a data/information
- E.g. `<h1>`, `<p>`, `<html>`

2. Closing

- Used to close the data/information
- E.g. `</h1>`, `</p>`, `</html>`

3. Empty

- Tag having no data/content
- Two ways of representing it
 - `<tag></tag>`
 - `<tag />`
 - E.g. `
`, `<hr />`

4. Root

- Tag which starts and ends the document

- Is also called as Document Type or Document Tag or Document Element
- E.g. `<html>` is root tag for html document

Attribute

- Extra information about the tag
- Attribute always present in name=value format
- E.g. `<meta charset="utf-8">`
 - meta: tag
 - charset: attribute name
 - utf-8: attribute value
- A tag may have one or more attributes
- Every tag has following attributes
- name: used to create query string
- id: used to identify an element uniquely
- style: used to write inline css
- class: used for css

HTML Structure

- It consists of 2 parts

1. Head

- Contains extra information about the page
- tags that can be used inside the head are
 1. title: used to set the title for the tab
 2. script: used to add JS code in the page
 3. style: used to add CSS code
 4. meta: used to add more information about the page
 5. link: used to link external documents (files)
 6. base: used to set the base url used in the page

2. Body

- Contains actual design
- tags that can be used inside the body are
 1. Textual
 2. Resources
 3. List
 4. Table
 5. Linking(Anchor)
 6. Form
- The tags can be inline or block tags
- the inline tags keep the data on the same line however the block tags add the data on the next line

Textual tags

1. Header: used to add header in page

- There 6 levels
- Tags: h1 to h6
- H1 is the biggest while h6 is the smallest

2. Paragraph (

); used to add a para

3. Division(

); used to create groups of Tags and Textual contents

- All the above are block tags

Formatting Tags

- These all are inline tags

1. span - used during CSS
2. bold - `` or ``
3. italic - `<i></i>`
4. underline - `<u></u>`
5. strike - `<strike></strike>`
6. monospace - `<tt></tt>`
7. superscript - ``
8. subscript - ``
9. font - ``
10. formatted - `<pre></pre>`

HTML Character Entities

	non-breaking space	<code>&nbsp;</code>	<code>&#160;</code>
<code><</code>	less than	<code>&lt;</code>	<code>&#60;</code>
<code>></code>	greater than	<code>&gt;</code>	<code>&#62;</code>
<code>&</code>	ampersand	<code>&amp;</code>	<code>&#38;</code>
<code>"</code>	double quotation mark	<code>&quot;</code>	<code>&#34;</code>
<code>'</code>	single quotation mark	<code>&apos;</code>	<code>&#39;</code>
¢	cent	<code>&cent;</code>	<code>&#162;</code>
£	pound	<code>&pound;</code>	<code>&#163;</code>
¥	yen	<code>&yen;</code>	<code>&#165;</code>
€	euro	<code>&euro;</code>	<code>&#8364;</code>
©	copyright	<code>&copy;</code>	<code>&#169;</code>
®	registered trademark	<code>&reg;</code>	<code>&#174;</code>
™	trademark	<code>&trade;</code>	<code>&#8482;</code>

List tags

1. Unordered list

- Does not render the order

```
<ul>
  <li>list item1</li>
  <li>list item2</li>
</ul>
```

2. Ordered list

- Renders the list item order

```
<ol>
  <li>list item1</li>
  <li>list item2</li>
</ol>
```

3. Definition list

- Used to create list of definitions

```
<dl>
  <dt>term 1</dt>
  <dd>definition 1</dd>

  <dt>term 2</dt>
  <dd>definition 2</dd>
</dl>
```

Table tag

- Used to create tabular representation
- Divided into 3 sections

1. Header

- Use `<thead>` (table header)

2. Body

- Use `<tbody>` (table body)

3. Footer

- Use `<tfoot>` (table footer)

- To create row Use `<tr>`
- To create column In header Use `<th>` (table header column)
- In Body and Footer Use `<td>` (table data)
- Example

```
<table>
  <thead>
```

```
<tr>
  <th></th>
</tr>
</thead>
<tbody>
  <tr>
    <td></td>
  </tr>
</tbody>
<tfoot>
  <tr>
    <td></td>
  </tr>
</tfoot>
</table>
```

- Attributes:
 1. border: used to create border
 2. colspan: used to merge multiple columns horizontally
 3. rowspan: used to merge multiple columns vertically

Resource tag

1. Images

- img: used to add image
 - src: specify the source
 - alt: used to alternative text (rendered only in case of missing source)

2. Audio

- audio: used to play audio
 - src: to set the file to play
 - controls: to render controls
 - autoplay: to play the file automatically

3. Video

- video: used to play video
 - src: to set the file to play
 - controls: to render controls
 - autoplay: to play the file automatically

Linking Tag (Anchor)

- Used to link multiple pages
 - `text`
- multiple sections within same page
 - `text`
- multiple pages with multiple sections

- `text`

Form tag

- Used to get inputs from user
- Inside the form tag we can use below tags

1. input

- used to get input from the user
- type attribute is used to specify the input type.
- value to type attribute can be
 - text: textual value
 - number: number value
 - email: email value
 - tel: telephone
 - date: date input
 - time: time input
 - radio: create radio button
 - checkbox: multiple selection
 - password: masked characters
 - submit: to submit values
 - reset: to clear the form
 - file: used to upload file

2. textarea

- used to get multi-line input

3. select

- used to render drop-down

```
<select>
  <option>op1</option>
  <option>op2</option>
</select>
```

GET Method

- Values will be sent using Request Head
- Values will be appended on url (visible)
- Restriction on maximum size of data to be passed
- Only ascii values can be sent using GET
- URL can be bookmarked with the values
- Files can NOT sent using GET

POST Method

- Values will be sent using Request Body
- Values will be invisible
- There is no restriction on size of data
- Any value of any type (binary) can be sent by using POST
- URL can be bookmarked without the values
- Files can be using POST

SUNBEAM INFOTECH