Complete Notes On HTML

Instagram: @ Curious_.programmer

Telegram: @ curious_coder

Written by:

Damini Patil (CSE student)

Index

1. Introduction to HTML

HTML is an acronym which stands for Hyper Text Markup Language which is used for creating web pages and web applications

1.1 What is HTML ?

HTML is Hyper Text Markup language.

Hyper Text :

HyperText simply means "Text within Text". A Text has a link within it, is a hypertext. Whenever you click on a link which brings you to a new webpage, you have clicked on a hypertext. HyperText is a way to link two or more web pages (HTML documents) with each other.

Markup language :

A markup language is a computer language that is used to apply layout and formatting conventions to a text document. Markup language makes text more interactive and the dynamic. It can turn text into images, tables, links, etc.

Web pages :

A Web page is a document which is commonly written in HTML and translated by a web browser. A web page can be identified by entering an URL. A web page can be of the static or dynamic tags. With the help of HTML only, we can create static web pages. Hence, HTML is a markup language which is used for creating attractive web pages with the help of

of styling, and which looks in a nice format on a web browser. An HTML document is made of many HTML Tags and each HTML tag contains different content.

Example of HTML:

```
<! DOCTYPE>
<Html>
<head>
<title> web page title </title>
</head>
<body>
<h1> Write Your First Heading </h1>
<p> Write Your First Paragraph. </P>
</body>
</html>
```

Description of HTML Example:

<! DOCTYPE>: It defines the document type or it instruct the browser about the version of HTML.
<html> : This tag informs the browser that it is an HTML document. Text between html tag describes the web document. It is a container for all other elements of HTML except <! DOCTYPE>
<head> : It should be the first element inside <html> element, which contains the metadata. It must be closed before the body tag opens.
<title> : As its name suggested, It is used to add title of that HTML page which appears at the top of the browser window. It must be placed inside the head tag and should close.

<body>: Text between body tag describes the body content of the page that is visible to the end user. This tag contains the main content of the HTML document.
<h1>: Text between <h1> tag describes the first level heading of the webpage.
<p>: Text between <p> tag describes the paragraph of the webpages.

1.2 History and evolution of HTML.

In the late 1980's, a physicist, Tim Berner's-Lee who was a contractor at CERN, proposed a system for CERN researchers. In 1989, he wrote a memo proposing an internet based hypertext system. Tim Berner-Lee is known as the Father of HTML. The first available description of HTML was a document called "HTML Tags" proposed by Tim in late 1991. The latest version of HTML is HTML5

HTML Versions :

Since the time HTML was invented there are lots of HTML versions in market, the HTML versions are given below:-

HTML 1.0 :- The first version of HTML was 1.0, which was the barebones version of HTML language, and it was released in 1991.
HTML 2.0 : This was the next version which was released in 1995, and it was standard language version for website design.
HTML 3.2: HTML 3.2 version was published by W3C in early 1997.
HTML 4.01: HTML 4.01 version was released on December 1999,

and it is a very stable version of HTML language.

HTML5: HTML5 is the newest version of HyperText Markup Language.

1.3 structure of an HTML document

```
<!DOCTYPE>
<html>
<head>
<title> web page title </title>
</head>
<body>
 <h1> Write Your First Heading </h1>
  <p> Write Your first Paragraph </p>
  </body>
  </html>
```

The above is document structure of HTML.

1.4 Basic HTML Tags

There are many tags in HTML which faciliate the process of describing web Page to the server. These tags are arranged in specific manner to get the desired output. Some of the important tags of HTML are given below:

<!DOCTYPE> - It describes the type of document.
<html> - describes that the document is an html document
<title> - describes the title of the document.
<body> - describes the body of the document
<h1> to <h6> - describes the headings in html.

<p>: describes a paragraph.

: produces a line break of single line.
<hr>: describes a thematic content change.
<!-- -->: Describe a comment.

Tags in HTML are useful entities that allows a developer to manipulate the look and functionalities of a web page. Additionally, some of tags have closing tags that include a forward slash, indicating a closing tag. Examples of the closing tags are body, title and HTML. However, some tags do not have closing tags; examples of such tags
 and <img>. Such elements are often known as empty or void elements in HTML.

Other useful tags in HTML:

<b> - It is used to bold the written text.
<i> - It is used to make the written text italic.
<u> - It is used to underline the content written in the tag.
 - It is used to form a list in the content.
 - It is used to form an ordered list in the content
 - It is used to form an unordered list in the content
<marquee> - Adds scrolling effect to the text or image in the content.

2. HTML Document Structure

2.1 HTML DOCTYPE declaration

On the HTML document you have often seen that there is a <!DOCTYPE> declaration before the <html> tag. HTML <!DOCTYPE> tag is used to inform the browser about the version of HTML used in the document. It is called as the document type declaration (DTD).

Technically <!DOCTYPE> is not a tag/element, it just instruction to the browser about the document type. It is a null element which does not contain the closing tag, and must not include any content within it.

Actually, there are many type of HTML e.g HTML 4.01 strict, HTML 4.01 Transitional, HTML 4.01 Frameset, XHTML 1.0 strict, XHTML 1.0 Transitional, XHTML 1.0 frameset, XHTML 1.1 etc.

The <!DOCTYPE> declaration refers Document Type Declaration (DTD) in HTML 4.01; because HTML 4.01 was based on SGML. But HTML5 is not SGML based language.

DTD defines the rules for the markup languages so that the browsers recognize the content correctly. The Doctype declaration differs between HTML versions. The HTML5 Doctype declaration is given below:

Syntax:

```
<!DocTYPE html>
```

Following are some specifications about the HTML <!DOCTYPE>

Display	None
start tag/ End tag	start tag only
Usage	structural

Example:

```
<!DOCTYPE html>
<html>
<head>
<title> This is the title </title>
</head>
<body>
This is the content of the document
</body>
</html>
```

Supporting Browsers:

chrome
Internet explorer
Firefox
Opera
Safari

DOCTYPE declaration is not case sensitive.

2.2 HTML Head Section :

The HTML <head> element is used as a container for metadata (data about data). It is used between <html> tag and <body> tag.
The head of an HTML document is a part whose content is not displayed in the browser on page loading. It just contains meta data about the HTML document which specifies data about the HTML document.
An HTML head can contain lots of metadata information or can have very less or no information, it depends on our environment. But head part has a crucial role an HTML document while creating a website.

Metadata defines the document title, character set, styles, links, scripts, and other meta information.
Following is a list of tags used in metadata:

<title>
<style>
<meta>
<link>
<script>
<base>

2.3 HTML body Section:

HTML <body> tag defines the main content of an HTML document which displays on the browser. It can contain text content, paragraphs, headings, images, tables, links, videos, etc.

The <body> must be the second element after the <head> tag or it should be placed between </head> and

</html> tags. This tag is required for every HTML document and should only use in the whole HTML document.

Syntax:

<body> Place your Content Here.... </body>

Following are some specifications about the <body> tag:

Display	Inline
start tag / End tag	Both start and End tag
Usage	structural

Example:

```
<! DOCTYPE html>
<html>
<head>
<title> Body Tag </title>
</head>
<body>
<h2> Example of body Tag </h2>
<p> This paragraph is written between the body tag </p>
</body>
</html>
```

Output:

Example of body tag

This paragraph is written between the body tag.

Supporting Browsers:

chrome
Internet Explorer
FireFox
Opera
Safari

2.4 HTML Meta Tag:

HTML <meta> tag is used to represent the meta-data about the HTML document. It specifies page description, keywords, copyright, language, author of the documents, etc.
The metadata does not display on the webpage, but it is used by search engines, browsers and other web services which scan the site or webpage to know about the web page. With the help of meta tag, you can experiment and preview that how your webpage will render on the browser. The <meta> tag is placed within the <head> tag, and it can be used more than one times in a document.

Syntax:

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

Following are some specifications about the HTML <meta> tag.

Display	None
start tag/End tag	Empty tag (only start tag)
Usage	Document structural

Following are some specific syntaxes of meta tag which shows the different uses of meta Tag.

1. <meta charset = "utf-8">

It defines the charster encoding. The value of charset is "utf-8" which means it will support to display any language.

2. <meta name = "keywords" content = "HTML, CSS, JavaScript, Tutorials">

It specifies the list keyword which is used by search engines.

3. <meta name = "refresh" content = "50">
It specifies to provide instruction to the browser to automatically refresh the content management system automatically.

4. <meta name = "author" content = "thisauthor">

It specifies the author of the page. It is useful to extract author information by content management system automatically.

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

It specifies the viewport to control the page dimension and scaling so that our website looks good on all devices. If this tag is present, it indicates that this page is mobile device supported.

supporting browsers :

chrome
Internet Explorer
Firefox
Opera
Safari

Attribute

Tag-specific attribute:

Attribute	Value	Description
charset	character-set	It specifies the character encoding of an HTML page.
Content	text	It contains the value of the attribute "name" and "http-equiv" in an HTML document, depending on context
http-equiv	• content-type • default-style • refresh	It specifies the Information given by the web server about how the web page should be served.
name	• application name • author • description • generator • keywords	It specifies the name of document-level metadata.

Global attribute:

The <meta> tag supports the global attributes in HTML.

Event attribute:

<meta> tag supports the event attributes in HTML.

Supporting Browsers:

chrome
IE
opera
Safari
Firefox

3. Text Formatting and Styles

3.1 Heading and paragraphs

A HTML Headings or HTML h tag can be defined as a title or a subtitle which you want to display on the webpage. When you place the text within the headings tag <h1> ... </h1>, it is displayed on the browser in the bold format and size of the text depends on the number of heading.

There are six different HTML headings which are defined with the <h1> to <h6> tags, from highest level <h1> main heading to the least level <h6> (least important heading).

Headings in HTML helps the search engine to understand the structure and index the structure of web page.

Example:

```
<h1> Heading no. 1 </h1>
<h2> Heading no. 2 </h2>
<h3> Heading no. 3 </h3>
<h4> Heading no. 4 </h4>
<h5> Heading no. 5 </h5>
<h6> Heading no. 6 </h6>
```

Output:

Heading no. 1
Heading no. 2
Heading no. 3
Heading no. 4
Heading no. 5
Heading no. 6

HTML paragraph :

HTML paragraph or HTML p tag is used to define a paragraph in a webpage. It is a notable point that a browser itself add an empty line before and after a paragraph. An HTML <p> tag indicates starting of new paragraph.

Example:

```
<p> This is first paragraph </p>
<p> This is second paragraph </p>
```

Output:

This is first paragraph

This is second paragraph

3.2 HTML formatting

HTML formatting is a process of formatting text for better look and feel. HTML provides us ability to format text without using CSS. There are many formatting tags in HTML. These tags are used to make text bold, italicized or underlined.

Bold Text:

HTML <b> and formatting elements.

The HTML element is a physical tag which display text in bold font, without any logical importance. If you write anything within .... element, is shown in bold letters.

  Write your First Paragraph in bold Text.

Example:

```
<! DOCTYPE html>
<html>
<head>
   <title> formatting elements </title>
</head>
<body>
<h1> Explaination of formatting element </h1>
<p> <strong> This is an important content </strong>, and this is
normal content </p>
 </body>
</html>
```

Italic Text:

HTML and formatting elements

The HTML element is physical element, which display the enclosed content in italic font, without any added importance. If you write anything within .... element, is shown in italic letters.

  Write your first Paragraph in italic text.

The HTML tag is a logical element, which will display the enclosed content in italic font, with added semantics importance.

 This is an important content , which displayed in italic font .

Example:

```
<!DOCTYPE html>
<html>
<head>
 <title> formatting elements </title>
</head>
<body>
<h1> Explaination of italic formatting element </h1>
<p><em> This is an important content </em>, which displayed in italic font. </p>
</body>
</html>
```

Underlined Text:

If you write anything within <u>.....</u> element, is shown in underlined text.

<u> write your first paragraph in underline text </u>

Output:

write your first paragraph in underline text

3.3 Lists (ordered and unordered)

HTML Lists

HTML Lists are used to specify list of information. All Lists may contain one or more list elements.

HTML Ordered List or Numbered list.

In the Ordered HTML lists, all the lists are marked with numbers by default. It is known as numbered list also. The ordered list starts with <ol> tag and the list items start with <li> tag.

```
<ol>
<li> Aries </li>
<li> Bingo </li>
<li> Leo </li>
<li> Oracle </li>
<ol>
```

Output:

1. Aries
2. Bingo
3. Leo
4. Oracle

HTML Unordered List or Bulleted List :

In HTML Unordered list, all the list items are

marked with bullets. It is also known as bulleted list also. The Unordered list starts with <ul> tag and list items start with the <li> tag.

```
<ul>
<li> Aries </li>
<li> Bingo </li>
<li> Leo </li>
<li> Oracle </li>
</ul>
```

Output:

- Aries
- Bingo
- Leo
- Oracle

3.4 Adding special characters:

There are many mathematical, technical and currency symbols which are not present on a normal keyboard. We have to use HTML entity names to add such symbols to an HTML page.

If there no entity name exists, you can use an entity number, a decimal, or hexadecimal reference

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

```
<h3> The Currency Symbols </h3>
   <p> This is India Rupee Symbol <b> ₹ <b> </p>
<p> This is Euro Symbol <b> € </b> </p>
<p> This is Dollar Symbol <b> &#36; </b> </p>
</body>
</html>
```

Mathematical Symbols Supported by HTML.

char	Number	Entity	Description
∀	∀	∀	FORALL
∂	∂	∂	PARTIAL DIFFRENTIAL
∃	∃	∃	THERE EXISTS
∅	∅	∅	EMPTY SETS
∇	∇	∇	NABLA
∈	∈	∈	ELEMENT OF
∉	∉	∉	NOT AN ELEMENT OF
∋	∋	∋	CONTAINS AS MEMBER
∏	∏	∏	N-ARY PRODUCT
∑	∑	∑	N-ARY SUMMATION

Greek symbols :

A	GREEK CAPITAL LETTER ALPHA
B	GREEK CAPITAL LETTER BETA
Γ	GREEK CAPITAL LETTER GAMMA
Δ	GREEK CAPITAL LETTER DELTA
E	GREEK CAPITAL LETTER EPSILON

Some Important Symbols Supported by HTML

©	COPYRIGHT SIGN
®	REGISTERED SIGN
€	EURO SIGN
™	TRADEMARK
←	LEFTWARDS ARROW
↑	UPWARDS ARROW
→	RIGHTWARDS ARROW
↓	DOWNWARDS ARROW

4. Hyperlinks and Anchors

4.1 Creating Hyperlinks:

The HTML anchor tag defines a hyperlink that links one page to another page. It can create hyperlink to other web page as well as files, location, or any URL. The "href" attribute is the most important attribute of the HTML a tag. and which links to destination page or URL.

Syntax:

```
<a href ='. . . . . ">Link Text</a>
```

You can open links in a new tab or window by adding the 'target' attribute with the value "blank" to the anchor tag.

```
<a href = "https://www.example.com" target = "_blank">
Visit Example.com </a>
```

4.2 Linking to external Websites:

To create a hyperlink to an external website, simply provide the full URL in the 'href' attribute. Users can click the link to visit the external site.

```
<a href = " https://www.example.com ">
 Visit Example.com Website </a>
```

When a user clicks on the link, they will be directed to

"https://www.example.com" webpage.

4.3 HTML linking to the internal section of a webpage

Sometimes you may want to create links that navigate to specific actions within the same webpage. HTML allows you to achieve this using anchor tags <a> and the 'id' attribute.

First assign an 'id' to the target section using the 'id' attribute.

For example:

```
<h3 id = "section1"> section 01 </h3>
<p> This is the content of section1
</p>
```

Then, create a link that points to that section using the 'href' attribute preceeded by a hash symbol ('#')

For example:

```
<a href = "#section1"> Go to section1
</a>
```

When this link is clicked, the webpage will scroll to the section with the 'id' of "section1"

4.4 Linking to E-mail addresses:

To create a link that opens the user's emails client with pre-filled email address, use the 'mailto:' scheme in the 'href' attribute. For example:

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

When a user clicks on this link, their default email client will open with a new email addressed to 'example@-example.com".

In the next chapter, we will explore how to add images to your webpages, allowing you to enhance visual appeal and engage users with visual content.

5. Images And Multimedia

5.1 Inserting images in HTML:

Images are a powerful way to enhance the visual appeal and engagement of your webpages. In HTML, you can insert images using the <img> tag.

To insert an image, you need to provide the source (URL) of the image using the 'src' attribute.

e.g:

```
<img src = "image.jpg" alt = "Description of image'>
```

In this example, the image with the filename "image.jpg' will be displayed on the webpage.
The alt attribute provides alternative text that describes the image, which is important for accessability purpose.

5.2 Image formats and optimization

1) JPEG
2) PNG
3) GIF

When choosing image formats for your webpages its important to consider factors such as file size,

quality and browser compatibility.

JPEG (Joint photographic Experts Group)

JPEG is commonly used format for photographs and complex images. It provides good image quality with compression. Use JPEG for colorful and detailed images.

PNG (portable Network Graphics)

PNG is a versatile format that supports loss less compression, making it suitable for images with transperent backgrounds or simple graphics. Use PNG for loops, icons or images with transperancy.

GIF (Graphics Interchange Format)

GIF is a format that supports animations and simple graphics. It uses loss less compression but also limited color support. Use GIF for small animated images or simple graphics

When optimizing images for the web, consider reducing the file size without significantly impacting image quality. This can be achieved through compression tools or online image optimization services.

5.3 Adding alternate text (Atribute)

The 'alt' attribute is used to provide alternative text for images. It serves two main purposes.

- **Accessibility:**

Alternative text allows screen readers to describe the image to visually impaired users, ensuring that they can understand the content.

- **SEO (Search Engine Optimization)**

Search engines use alternative text to understand the context of an image. Providing descriptive and relevant alternative text can improve your webpages visibility in search results.

It's important to include descriptive and concise alternative text that accurately represents the image's content or function.

```
<img src = "Image.jpg" alt = "A happy
family enjoying a picnic in the park">
```

5.4 Embedded videos and audio files

HTML provides tags to embed videos and audio files directly into webpages, offering an

immersive multimedia experience

- **Video**

To embed a video, use the <video> tag and specify the video source using the <source> tag within it

For example:

```
<Video controls>
   <source src ="video.mp4" type ="video/mp4">
</video>
```

The 'controls' attribute adds playback controls to the video, allowing users to play, pause and adjust volume.

- **Audio:**

To embed audio, use the <audio> tag and specify the audio source using the <source> tag within it.

For example:

```
<audio controls>
   <source src = "audio.mp3" type = "audio/mp3">
</audio>
```

Similarly, the 'controls' attribute adds playback controls to the audio, enabling users to play, pause and adjust volume.

In the next chapter, we will explore how to create forms in HTML, allowing users to interact with and submit data on your web pages.

6. Tables in HTML

6.1 Creating tables in HTML:

Tables are useful for organizing and presenting structured data on a webpage. In HTML you can create tables using the <table> tag.

To create a table, use the following structures;

```
<table>
    <!-- Table content goes here -->
</table>
```

Inside the <table> element, you will add the content of your table such as rows, columns, headers and cells.

step1: Firstly we have to type the HTML code in any text editor or open the existing HTML file on the text editor in which we want to make a table.

step2: Now, place the cursor the at the point between the body tag where we want to display a table on the page. And, then type the <table> tag at that point.

step3: After then, we have to type the <tr> tag in the next line, which indicates the table row. It is the first <tr>

tag in the table. So, it indicates the first row of the table.

```
<!DOCTYPE Html>
<Html>
<Head>
<Title>
Make a Table
</Title>
</Head>
<Body>
Hello User! <br>
The following table shows the student details <br><br>
<table> <!-- The table tag which allows page to create table-->
<tr> <!-- First Row in the table-->
<th>
Roll No.
</th>
<th>
student name
</th>
<th>
Marks
</th>
<th>
Address
</th>
</tr>  <!-- First row closed -->
<tr>  <!-- Second row started -->
<td>
101
</td>
```

```
<td>
Arun
</td>
<td>
85
</td>
<td>
Mumbai
</td>
</tr> <!-- Second row start closed -->
<tr> <!-- Third row started -->
<td>
102
</td>
<td>
Akshay
</td>
<td>
90
</td>
<td>
Goa
</td>
</tr> <!-- Third row closed -->
</table>
</Body>
</Html>
```

Output

Hello User!
The following table shows the student details

Roll No	Student - Name	Marks	Address
101	Arun	85	Mumbai
102	Akshay	90	Goa

6.2 Table structure (Rows, Columns, cells)

Tables are composed of rows, columns and cells.

- **Rows**

Use the <tr> tag (table row) to define a row within the table

For example:

```
<tr>
  <!-- cells for the first rows -->
</tr>
```

- **Columns:**

Inside each row, you can add cells that represent the columns of your table. Use the <td> (table data) tag to define a cell.

For example:

```
<tr>
  <td> cell1 </td>
  <td> cell2 </td>
</tr>
```

- Cells:

Cells contains the actual content of your table. You can add text, images or other HTML elements within the cell tags.

6.3 Table headers and Captions

Tables can have headers and captions to provide additional context and organization.

Table header

Use the <th> (table Header) tag to define header cells within the '<thead>' (table head section)

```
<thead>
  <tr>
  <th> Header 1 </th>
  <th> Header 2 </th>
  </tr>
</thead>
```

Table Caption:

To add a caption to your table, use the <caption> tag immediately after the opening of <table> table tag.

For example:

```
<table>
  <caption> This is my table Caption </caption>
  <!-- Table content goes here -->
</table>
```

6.4 Merging cells and spanning rows/columns

HTML allows you to merge cells and span rows or columns to create more complex table layouts.

Merging cells:

Use the 'colspan' attribute to merge cells horizontally (across columns) or the 'rowspan' attribute to merge cells vertically (across rows)

For example:

```
<tr>
  <td colspan = "8" > Merged cell </td>
</tr>
```

In this example, the cell will span across two columns.

- Spanning rows/columns

Use the 'colspan' or 'rowspan' attribute within the <td> or <th> tags to specify the number of columns or rows the cell should span

For example:

```
<tr>
 <td rowspan = "2"> spanned cell </td>
 <td>cell2' </td>
</tr>
<tr>
 <td> cell3 </td>
 </tr>
```

In this example, the first cell will span across two rows.

7. Forms & Input Elements

7.1 HTML forms and input elements:

HTML forms are essential for gathering user inputs and enabling interactions on webpages. To create a form use the <form> tag.

The basic structure of form includes the opening and closing the <form> tags and various form elements within them. You can also specify attributes to control the behaviour of the form, such as the 'action' attribute to define the URL where the form data will be submitted

```
<form action = "/submit" method = "post">
  <!-- form elements go here -->
</form>
```

7.2 Text Input fields:

Text input fields allow users to enter text or numeric value. Use the <input> tag with the 'type' attribute act set to 'text' to create a text input field.

For example:

```
<input type = "text" name = "username"
  placeholder = "Enter your username">
```

In this example, the 'name' attribute identifies the input field, and the 'placeholder' attribute provides a hint to the user about the expected input.

7.3 checkbox and Radio Buttons

checkbox and Radio Buttons are used to provide options for uses to select from.

- **checkboxes**

Use the <input> tag with the 'type' attribute set to "checkbox" to create checkboxes.

```
<input type = "checkbox" name = "interest"
values = "music">
<label for = "music"> Music
</label>
```

In this example, the 'name' attribute groups the checkboxes, and the 'values' attribute specifies the value when checkbox is selected.

- **Radio Buttons**

Use the <input> tag with the 'type' attribute Also set to "radio" to create radio buttons.

```
<input type = "radio" name = "gender"
value = "male">
<label for = "male"> Male </label>
```

7.4 Dropdown menus and selection lists

Dropdown menus and selection lists allow users to choose an option from a predefined set.

Dropdown Menu :-

Use the <select> tag with nested <option> tags to create a dropdown menu.

For example:

```
<select name = "country">
  <option value = "USA"> USA </option>
  <option value = "uk"> Uk </option>
  <option value = "canada"> Canada </option>
</select>
```

The name attribute identifies the dropdown menu, and each '<option>' tag depends represents an option within the menu.

Selection lists :-

Use the '<select>' tag with the 'multiple' attribute to create a selection list that allows multiple options to be chosen.

7.5 Submitting forms and form validation

To submit a form, use the <input> tag with the 'type' attribute set to "submit".

For example:

```
<input type="submit" value="submit">
```

When the user clicks the submit button, the form data will be sent to the URL specified in the form's 'action' attribute.

Form validation ensures that the data entered by the user meets the specified requirements.

HTML provides built in validation attributes such as 'required' 'minlength' ;maxlength' and 'pattern' which can be applied to form elements to validate user input.

8. HTML5 Semantics Elements

8.1 Introduction to Semantic Markup

HTML5 introduced a set of semantics elements that provide a more meaningful and structured way to describe the content and organization of a webpage. Semantic markup improves accessibility, search engine optimization and overall understanding of the page's structure.

Semantic elements are designed to convey the purpose and meaning of the content they contain. By using these elements, you can write more descriptive and semantically rich webpages.

Semantic elements = elements with a meaning.
Semantic elements have a simple and clear meaning for both, the browser and the developer.

Semantic Elements in HTML5:

<article>
<aside>
<details>
<figcaption>
<figure>
<footer>
<header>
<main>
<mark>
<nav>
<section>
<summary>
<time>

8.2 Header, Nav and Footer element

- Header ('<Header>')

The header element represents the introductory or navigational content of a webpage or content within it It typically includes logos, site titles and navigation menus.

```
<header>
  <h1> My website </h1>
  <nav>
    <!-- navigation links go here -->
  </nav>
</header>
```

- Navigation ('<nav>')

The 'nav' element is used to define a section of a webpage that contains navigation links.

```
<nav>
   <ul>
   <li> <a href = "/ "> Home </a> </li>
   <li> <a href = "/about "> About </a> </li>
   <li> <a href = "/ Contact"> Contact </a> </li>
   </ul>
</nav>
```

- footer ('<footer>')

The footer elements represents the footer or cloning content of a webpage. It often contains copyright information, contact details or links to relevant resources.

```
<footer>
  <p> &copy; 2023 My website. All rights
    reserved </p>
</footer>
```

8.3 Section, Article and Aside elements

- ('<section>') Section

The Section element represents a standalone section within a webpage. It helps in dividing the content into meaningful blocks or themes.

```
<section>
  <h2> About us </h2>
  <p> Lorem ipsum dolor sit amet,
    consectetur adipiscing elit... </p>
</section>
```

- Article ('<articles>')

The 'article' elements represents a self contained composition that can be indepently distributed or reused. It can represent blogspots, news artides from threads or any other independent pieces of content.

```
<article>
  <h2> How to bake a cake </h2>
  <p> Lorem ipsum dolor sit amet,
    consectetur adipscing elit... </p>
</article>
```

- Aside ('<aside>')

The 'aside' element represents, content that is tangentially related to the main content. It can be used for sidebar, pull quotes or advertisements.

```
<aside>
   <h3> Related Artides </h3>
    <ul>
     <li> <a href = "/article1> Article1 </a> </li>
     <li> <a href = "/article2> Article2 </a> </li>
     <li> <a href = "/article3> Article3 </a> </li>
    </ul>
```

</h3>
</aside>

8.4 Figure and Figcaption elements

- figure ('<figure>')

The 'figure' element is used to encapsulate media content, such as images, illustrations, diagrams or videos, along with an optional caption.

```
<figure>
    <img src = "image.jpg" alt = "A beautiful
     sunset">
     <figcaption> A beautiful sunset </figcaption>
</figure>
```

9. CSS Fundamentals

9.1 Introduction to CSS

Css stands for Cascading Style sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. It can also be used with any kind of XML documents including plain XML, SVG and XUL

Css is used along with HTML and javaScript in most websites to create user interfaces for web applications and user interfaces for many mobile application.

What does css do:

- You can add new looks to your old HTML documents.
- You can completely change the look of your websites with only a few changes in Css code.

9.2 Inline, Internal and External stylesheets

We can apply Css in a single element by inline Css technique.

The inline Css is also a method to insert style sheets in HTML documents. This method mitigates some advantages

of style sheets so it is advised to use this method.

Syntax:

```
<htmltag style = "cssproperty1 : value; cssproperty2 : value;">
</htmltag>
```

Example:

```
<h2>style = "color:red; margin-left:40px;" > Inline CSS is
applied on this line. </h2>
<p> This is not affected. </p>
```

output:

Inline CSS is applied on this line.

This is not affected.

Internal CSS :

The internal style sheet is used to add a unique style for a single document. It is defined in <head> section of the HTML page inside the <style> tag.

for example:

```
<head>
  <style>
    p {
      color : blue;
      font-size : 16px;
    }
  </style>
</head>
```

Internal stylesheets are useful when you want to apply styles to multiple elements within a single webpage.

External stylesheets.

An external stylesheet is a seprate css file generally used when you want to make changes on multiple pages. It is ideal for this condition because it facilitates you to change the look of the entire web site by changing just one file.

It uses the <link> tag on every pages and the <link> tag should be put inside the head section.

Example:

```
<head>
 <link rel = "stylesheet" type = "text/css" href = "
   my.css">
</head>
```

9.3 CSS selectors and properties:

Selectors:

CSS selectors target specific HTML elements to which styles should be applied. Common selectors include element selectors (e.g 'p', 'h1', 'div'), class selectors (e.g 'highlight', 'container') and ID selectors (e.g '#header', '#logo')

Properties:

CSS properties define the specific styles to be applied to the selected elements.

Example of CSS properties include 'color', 'font-size', 'background-color', 'margin' and 'padding'. Properties are assigned values to control the appearance of elements.

9.4 Applying styles to HTML elements:

To apply styles to HTML elements, you need to combine selectors with properties. Here's an example.

```
<style>
    /* Element Selector */
    p {
```

```
    color : Blue;
    font-size : 16px;
}
/* class selector */
.highlight {

    background-color : yellow;
    font-weight : bold;
}
/* ID selector */
#header
 }
   text-align : center ;
   font-size : 24px;
}
</style>

<p> This is a styled paragraph </p>

<p class = "highlight"> This paragraph has
a yellow background and bold text."
</p>

<h1 id = " selector header "> Welcome to
my website.
</h1>
```

10. CSS Layout & Positioning

10.1 Box Model (Margin, Padding, Border)

In CSS, every HTML element is represented as a rectangular box. The box model consists of four components.

1) Content
2) Padding
3) Border
4) Margin

Content:

The area (content area) is where the actual content of an element is displayed. It can contain text, images, or other HTML elements. The content's size is determined by properties like 'width' and 'height'.

Padding:

Padding is the space between the content and the element's border. It provides inner spacing and can be adjusted individually for each side.

e.g

padding-top, Padding-right,
padding-bottom, padding-left.

Border:

The border surrounds the padding and content. It can be customized with properties such as 'border-width', 'border-style' and 'border-color'. The border can take various styles, including solid, dashed and dotted.

Margin:

Margins are the spaces outside the border, creating sepration between elements. Margins can be set indivi dually (e.g margin-top, margin-right, margin-bottom, margin-left) and collapse under certain circumstances such as adjacent vertical margins.

10.2 Display - properties (Block, Inline, Inline-Block)

CSS provides different display properties to full width control how elements are rendered on webpage.

Block-level elements:

Block level elements span the full width of their parent container and stack vertically, creating a new line. Examples of Block elements include ' ', <p>, <h1> to <h6> and <section>.

Block level elements can have width, height, margins and padding and borders applied to them.

Inline-Elements

Inline elements flow within the content and do not create line breaks. They take up only the space they need to accomodate their content. Examples of Inline elements include '', '<a>' '' and '' Inline elements generally do not have width, height, margins, padding or borders applied to them.

Inline-block elements:

Inline block elements combine properties of both block and Inline element. They flow within the content like Inline elements but can have width, height, margins padding and borders applied to them.

10.3 Positioning element (Relative, Absolute, Fixed)

Css positioning allows you to precisely control the placement of elements of a webpage.
There are three commonly used positioning properties:

Relative positioning:

Elements with relative positioning are positioned relative to their normal position within the document flow.
You can use properties like 'top', 'bottom', 'left', and 'right' to move the elements from its original position.

Absolute positioning:

Elements with absolute positioning are positioned relative to their closest positioned ancestor. If no ancestor is positioned, they are positioned relative to the document body.

Fixed positioning:

Elements with fixed positioning are positioned relative to the browser window. They remain fixed in their position even when the page is scrolled.

10.4 Float and clear properties

The float and clear properties are commonly used for creating multi-column layouts for positioning elements around each other.

Float:

The float property allows you to move an element to the right or left of its containing element.
When an element is floated, other content flows around it. This is useful for creating columns or wrapping text around images.

clear:

The clear property specifies which sides of an

element should not allow floating elements. It ensures that no floating elements are present on the specified sides of the element.

11. Responsive Web Design

11.1 Understanding responsive design principles.

Responsive web design is an approach to create websites that adopt and respond to different screen sizes and devices.

With the increasing use of smartphones and tablets, It has become essential to ensure that websites look and function well across various devices.

The principles of responsive design involves:

- Fluids grids:

Using relative units, such as percentage, to create flexible grid between screen sizes.

Flexible images:

Scaling images proportionally to fit the screen while maintaing their aspect ratios.

Media queries

Using css media queries to apply different styles based on the chardcteristics of the device, such as screen width, orientation or resolution.

By following these principles, websites can provide a semlesss and user-friendly experience on devices of all sizes.

11.2 Media queries and Viewport meta tag

Media queries are essential components of responsive design. They allow you to apply specific CSS style based on the characteristics of the device being used to view the webpage. Media queries typically target properties like screen width.

Here is an example of media query:

```
@media screen and (max-width : 768 px) {
    /* styles applied when the screen
       width is 768 px or less */
  body {
      font-size : 14 px;
    }
}
```

In addition, to media queries the viewport meta tag is crucial for responsive design

It ensures that the webpage is rendered correctly on mobile devices by adjusting the viewport width and scaling.

The following meta tag is commonly used:

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

11.3 Creating fluid layouts with CSS Grids and flexbox:

CSS grids and flexboxes are powerful tools for creating responsive layouts. They allow you to create flexible and adaptive grids, columns and rows without relying on fixed pixel based measurements.

CSS Grid layouts:

- Provides a two dimensional grid system that allows you to create complex layouts with ease.
- You can define the number of rows and columns set their sizes and control the placement of elements within the grid.

CSS Flexbox:

- Provides one dimensional layout model that allows you to create flexible and dynamic layouts.

- you can control the alignment, order and spacing.

11.4 Reposive images and media

Responsive images:

- Use the 'max-width: 100%' Css role to ensure that images automatically scale down to fit their container.

Responsive media:

- Use Css media queries to adjust the size and positioning of media elements based on the screen size.