HTML NOTES

Topics

#Introduction

1. HTML developed by Tim Berners-Lee in 1991.

2. It is product of SGML (Standard Generalised Markup Language).

3. HTML is short for Hyper Text Markup Language.

4. HTML is a markup language that tells web browsers how to structure the web pages you visit.

5. Without HTML, a browser would not know how to display things , it provides a basic structure of the page.

6. HTML document should be saved with .html extension.

7. The Current version of HTML is HTML5 (2014).

#Doctype

1. It is document type declaration.

2. <!DOCTYPE> is a declaration in HTML that specifies the version of HTML used in a web page.

3. **It is not an HTML tag.**

4. The <!DOCTYPE> declaration must be the very first line of an HTML document, before the <html> tag.

5. It is important to include the correct <!DOCTYPE> declaration in an HTML document, as it can affect how the document is rendered in different web browsers.

6. The HTML5 doctype declaration is:<!DOCTYPE html>

7. In the case of HTML5, the doctype declaration is very simple and concise, and it replaces the longer, more complex doctype declarations that were used in previous versions of HTML.

#Tag

1. An HTML tag is a keyword surrounded by angle brackets, < and >.

2. They are used to mark up the content of a web page, and to specify how the content should be displayed in a web browser.

3. Every tag in HTML perform specific tasks.

4. **Example** :
<html> : it means it is starting of Html tag.</html> : Here , '/' means it is ending of Html tag.

5. The <html> tag represents the root of an HTML document.

#We Have Two Types Of Tags

1. Container Tag (Paired tag)

1. Those tags we have to close , we call them Container tag.

2. **Example** : < html > </html> , <head></head>, <body> </body>, etc.

2. Non Container tag

1. Non Container tags are knows as void tag , empty tag and singleton tag because they don't have cloasing tag , so we cannot put content in between tag

2. Those tags we need not to close , we call them non Container tag

3. **Example** : <br>,<img> , <input> , etc.

#Elements

1. HTML elements are the basic units of HTML.

2. An HTML element consists of a opening tag, content, and an closing tag, and can also contain attributes that provide additional information about the element.
Element = opening tag + content + closing tag.

4. For example, the following code creates an HTML element called "p" (which stands for "paragraph"), with the text "This is a paragraph." as its content:`<p> This is a paragraph. </p>`In this example <p> is the start tag, 'This is a paragraph.' is the content, and </p> is the end tag. The "p" element defines a paragraph of text.

#Types Of Elements

1. Block Level Element

    1. These elements will take complete width of the container.
    2. Theses elements will always start from a new line.
    3. We can set Height and Width for the block level elements.
    4. **Example**: <p> tag, heading tags,<div> tag, etc.

2. Inline Level Element

    1. These elements will take space required by the content.
    2. These elements will not start from a new line.
    3. We cannot set Height and Width for the inline level elements.
    4. **Example**: <img> tag, <b> tag,<span> tag, etc.

#Attributes

1. Attributes are html tag properties.

2. They are used to provide additional information about tag , so that tag can behave in our desired manner

3. HTML attributes consist of a name and a value, separated by an equals sign (=). The value is usually enclosed in quotes ("), although some attributes may use single quotes (') or no quotes at all.

4. Attributes should be applied in opening tag only.

5.**Example:**`<p title="i am paragraph tag"> Welcome to Saket's Blog </p>`Here , title is attribute .

#Comments

1. Comments are used to document the source code.

2. Comments are not displayed on the browser.

3. They are useful for providing explanations for yourself or other developers who may work on the code in the future.

4. Comments are not for the user.

5. HTML comments are written within the <!-- and --> tags.

6. Anything between <!-- Comment --> tags will be ignored by the browser and will not be displayed on the web page.

#Text Formatting Tags

1. **Italics tag**

    1. It is inline level element.
    2. It is container tag.
    3. It is used to display content in italics style.
    4. The text between <i> ..text.. </i> tag will be in italics style.
    5. **Example**: *This is italics text.*

2. Bold tag

1. It is inline level element.
2. It is container tag.
3. It is used to display content in bold style.
4. The text between <b> ..text.. </b> tag will be in bold style.
5. **Example**: This is bold text.

3. Underline tag

1. It is inline level element.
2. It is container tag.
3. It is used to underline the text.
4. The text between <u> ..text.. </u> tag will be underlined.
5. **Example**: This is underlined text.

4. Emphasise tag

1. It is inline level element.
2. It is container tag.
3. It is used to display content in italics style.
4. The text between <em> ..text.. </em> tag will be in italics style and also mark the text as important for crawler.
5. **Example**: **This is italics text.**

5. Strong tag

1. It is inline level element.
2. It is container tag.
3. It is used to display content in bold style.
4. The text between <strong> ..text.. </strong> tag will be in bold style and also mark the text as important for crawler.
5. **Example**: **This is italics text.**

6. Insert tag

1. It is inline level element.
2. It is container tag.
3. It is used to mark the inserted content in webpage.
4. The text between <ins> ..text.. </ins> tag will be marked as recently added content in website and also explain the same to crawler.
5. **Example**: This is underlined text.

7. Superscript tag

1. It is inline level element.
2. It is container tag.
3. It is used to display text as superscript.
4. The text between <sup> ..text.. </sup> tag display text as superscript.
5. **Example**: 10^8 (Here, 8 is written at top of 10 and 8 is superscript here.)

8. Subscript tag

1. It is inline level element.
2. It is container tag.
3. It is used to display text as subscript.
4. The text between <sub> ..text.. </sub> tag display text as subscript.
5. **Example**: CO_2 (Here,2 is written at bottom of O and 2 is subscript here.)

9. Code tag

1. It is inline level element.
2. It is container tag.
3. The <code> tag is used to indicate that a section of text is code, typically used for programming languages or markup languages.
4. The text between <code> ..text.. </code> tag is displayed as code.
5. **Example**: <code>console.log ('Hello from JavaScript') </code> (Here, <code> tag is used to represent JavaScript code.)

10. Address tag

1. It is block level element.
2. It is container tag.
3. The <address> tag is used to provide contact information for the author or owner of a document.
4. The text between <address> ..text..</address> tag is displayed as address.
5. **Example**: <address>A-123 Main Str <br> USA</address> (Here, <address> tag is used to represent address of some place.)

11. Quote tag

1. It is inline level element.
2. It is container tag.
3. <q> tag is used for shorter quotes i.e, for single line quotation.
4. The text between <q> ..text.. </q> tag is displayed as quotation(text will be in double quotes)
5. **Example**: <q> I am quote tag </q> (Here, <q> Tag is used to represent single line quotation.)

12. Blockquote tag

1. It is block level element.
2. It is container tag.
3. The <blockquote> tag is used for longer quotes i.e, multi line quotation
4. The text between <blockquote> ..text.. </blockquote> tag is The blockquote is usually indented and styled differently from the surrounding text
5. **Example**: <blockquote>I am blockquote tag. </blockquote> (Here, <blockquote> tag is used for multi line quotation.)

13. Abbr tag

1. It is inline level element.
2. It is container tag.
3. The <abbr> tag is used to mark up the abbreviation.
4. The text between <abbr> ..text.. </abbr> tag is used for mark up the term for abbreviation and we can provide the expansion of the term by using title attribute of <abbr> tag
5. **Example**: <abbr title='World Wide Web Consortium'> W3C </abbr> (Here, <abbr> tag is used for marking abbreviation to W3C and the expansion of W3C is written within titile attribute, so when the user hovers over the term its expansion is displayed.

14. del tag

1. It is inline level element.
2. It is container tag.
3. The <del> tag is used to indicate that text has been deleted or removed from a document.
4. The text between <del> ..text.. </del> tag is displayed as deleted content.
5. **Example**: <del>Kerala</del> (Here, Kerala has been struck through using the <del> tag to indicate that it has been deleted.)

15. Br tag

1. It is non container tag
2. It is used to provide line breaks.
3. **Example**: <p> Hello, <br> I am pragraph tag. (Here, <br> tag is used to provide line break after Hello)

#Heading Tags

1. Heading tags are used to create headings in HTML.

2. In Html we have 6 types of heading.

3. All heading tags are container tags.

4. <h1> Heading-1 </h1>(Biggest and Most Important Heading of a webpage.

5. <h2> Heading-2 </h2>

6. <h3> Heading-3 </h3>

7. <h4> Heading-4 </h4>

8. <h5> Heading-5 </h5>

9. <h6> Heading-6 </h6> (Smallest and least important heading)

10. <h1> should be used only once in a webpage.

11. **Note**: Browser will automatically add one line break before and after the heading tag.

#Paragraph Tag

1. To create paragraphs in Html we have <p> tag.

2. <p> tag is block level element.

3. <p> tag is container tag.

4. **Syntax**: <p> Content </p>

5. Browser will automatically add one line break before and after the <p> tag

#Image Tag

1. <img> tag is used to attach or embed images to our webpage.

2. It is non-container tag.

3. It is inline level element.

#Attributes Of Image Tag

1. **src attribute** - It is used to provide path or location of the image.(Syntax:- src='path')

2. **alt attribute** -It is used to provide alternative content or text to the image and this alternative content will be displayed only when the image is not displayed due to any reason and we should write meaningful content within alt attribute.(Syntax:- alt='text')

3. **height attribute** - It is used to set height of the image.(Syntax:- height='value')

4. **width attribute** - It is used to set width of the image.(Syntax:- width='value')

#Example

1. <img src = 'path' alt='text' height='300px' width='500px'>

2. **Note**: - Path should be relative (The path with respect to current directory is called as relative path)

#Hyperlink

1. Hyperlinks are used to link our webpage with some other documents or other part of document.

2. Hyperlinks are created using the <a> (anchor) tag and are usually displayed as underlined text in blue color.

3. To provide path or hyper reference we have to use href attribute of <a> tag.

#Anchor Tag

1. Anchor tag is used to create a hyperlink on a web page.

2. It is container tag.

3. It is inline level element.

4. The content that we want to create as a hyperlink should be written within <a>..</a> tag.

5. **Syntax**:- <a> ..text.. </a> (Here, we have marked the text to create hyperlink.)

#Attributes Of Anchor Tag

1. **href attribute** - It is used to provide path or hyper reference to the marked content. When user clicks on hyperlink that hyperlink takes the user to the specified path given in href attribute.

(<a href ="path"></a>)

2. **target attribute** - It is used to specify where to open the hyperlink. <a target="_blank"></a> means that our hyperlink will open in a new tab.

#Example

1. <a href = 'https://www.google.com' target="_blank"> Google</a> is a search engine.
Here, we have marked Google text to create hyperlink and when we click on the text,it will take us to the path specified in the href attribute i.e., google website.

#List

1. List is collection of related items.

2. There are three types of lists in HTML:- Ordered List, Unordered List and Description List.

#Ordered List

1. Ordered list is used to display list items in a sequential order.

2. It is block level element.

3. To create an ordered list, we use the <ol> tag. (It is container tag)

4. Here, we can represent sequence as alphabets,digits or roman numbers.

5. To represent list items we have <li> tag.(It is container tag)

#Attributes Of Ordered List

1. Type attribute : It is used to specify the type of numbering used for the list items.
Syntax: type = 'value'(This value can be: '1' (for numbers), 'A' (for uppercase alphabets), 'a' (for lowercase alphabets), 'I' (for uppercase roman), 'i' (for lowercase roman)).

2. Start attribute : Defines the starting number or sequence of the ordered list.
Syntax: start='digit/number'(Note: start attribute only accepts digits.)

3. Reversed attribute : It reverses the numbering/sequece of the list items.
Syntax: reversed

#Unordered List

1. Unordered list is used to display list items in a bulleted list format(unordered way).

2. It is block level element.

3. To create an unordered list, we use the <ul> tag. (It is container tag)

4. Here, we can represent sequence as disc,square or circle.

5. To represent list items we have <li> tag.(It is container tag)

#Attributes Of Unordered List

1. Type attribute : It is used to Specify the type of bullet used for the list items.
Syntax: type = 'value'(This value can be: 'disc', 'circle', 'square', 'none').

#Description List/ Definition List / Dictionary List

1. A Description list is used to define terms and their corresponding definitions.

2. To create a description list in Html, we use the <dl> tag.(It is container tag)

3. To represent term and it's definition we have <dt> ..term.. </dt> tag and <dd> ..definition.. </dd> tag respectively.(Both <dt> and <dd> are container tags)

4. We have to write <dd> ..definition.. </dd> tag within the <dt> ..term.. </dt> tag.

#Table

1. Table is a grid organised in rows and columns much like spreadsheets(Combination of rows and columns).

2. Cell - Intersection of rows and columns.

3. HTML tables allow you to display tabular data in a structured format.

4. In Html we have <table>..</table> tag to create tables.

5. <table> tag is a container tag.

6. In Html, tables are created row by row.

7. To represent table row we have <tr>..</tr> tag.(It is container tag)

8. To create cells we have <th>..</th> and <td>..</td> tags where <th> represents column heading cells and <td> represents data cells.

9. Both <th> and <td> tags are container tags.

#Caption Tag

1. It is used to give the title to the table.

2. <caption>..</caption> tag is a container tag.

3. We have to write <caption> tag within the <table> tag.

#Cell Spanning

1. The process of merging or combining two or more adjacent cells in a table

2. We can span the cells in two ways.

3. **Column-wise** - To span the cells on column basis we have colspan attribute.
   Syntax: colspan = 'number of cell space'.
   Example: **colspan**='2'(Here, two cells are spanned on column basis)

4. **Row-wise** - To span the cells on row basis we have rowspan attribute.
   Syntax: rowspan = 'number of cell space'.
   Example: rowspan='3'(Here, three cells are spanned on row basis)

5. **Note**: rowspan and colspan can be applied only on <td> and <th> tags and not on <tr> tag.

#Cell Spacing

1. The space around the cell is known as cell spacing.

2. To set the cell spacing, you can use the cellspacing attribute on the <table> element.

3. **Syntax**: cellspacing='value'(value in px)

4. **Example**: cellspacing='5px' (Here, 5px space is created around the cell)

#Cell Padding

1. The space between the cell content and cell borders is known as cell padding.

2. To set the cell padding, you can use the cellpadding attribute on the <table> element.

3. **Syntax**: cellpadding='value'(value in px)

4. **Example**: cellpadding='5px' (Here, 5px space is created between the cell content and cell border)

#Column Grouping

1. In Html we can group the cells on column basis by using <colgroup>..</colgroup> tag.

2. <colgroup> tag is a container tag.

3. To represent table columns in <colgroup> we have <col> tag.

4. <col> tag is a non-container tag.

5. The sequence of <col> tag represents the table columns respectively.

#Thead, Tbody And Tfoot Tag

1. The <thead>, <tbody>, and <tfoot> tags are used to group content within an HTML table.

2. These tags help to structure the table and make it easier to read and understand.

3. **Thead tag** - <thead>..</thead> tag represents the top most part of our table.<thead> tag is a container tag.The column headings of table should be written within <thead> tag.<thead> should be placed before the <tbody> tag.

4. **Tbody tag** - <tbody>..</tbody> tag represents the main content of the table.<tbody> tag is a container tag.The row data/content should be written within the <tbody> tag.<tbody> tag should be placed after the <thead> tag

5. **Tfoot tag** - <tfoot>..</tfoot> tag represents the bottom part of the table(i.e., footer content).<tfoot> tag is a container tag.The conclusion/summary of the table should be written within the <tfoot> tag.<tfoot> tag should be placed after the <tbody> tag.

6. **Note**:- <thead>, <tbody> and <tfoot> tags are not mandatory but highly recommended to use while working with tables.

#Form

1. Forms are used to collect user data or information.

2. To create a form in HTML, we have to use <form>..</form> tag.

3. <form> tag is a container tag.

4. It is a block level element

5. **Syntax**:- <form> --All form content-- </form>

#Input Tag

1. Introduction

1. The Input tag is used to create input field and it is one of the most commonly used form elements in HTML.

2. The <input> tag allows the user to enter information or make selections.

3. The <input> tag is a non-container tag.

4. It is an inline level element.

5. **Syntax**:- <input> (Here, an empty input field will be created)

2. Type Attribute of <input> tag

1. The type attribute of the <input> tag specifies the type of input the input field can accept.

2. **Syntax:** type='value'(Here, this value can be text, email, password,etc)

3. **type = 'text'** (This creates a single-line text input field where users can enter any text.)

4. **type = 'email'** (This creates a text input field that validates the user's input to ensure that it is a valid email address.)

5**. type = 'tel'** (This creates an input field that is specifically designed for telephone numbers.)

6. **type = 'password'** (This creates a text input field where the entered text is hidden and replaced with bullets, to protect the user's password from being seen by others.)

7. **type = 'submit'** (This creates a button that allows users to submit the form to the server.)

8. **type = 'date'** (This creates a text input field that allows users to select a date from a calendar popup.)

9. **type = 'time'** (This creates a text input field that allows users to select a time from a dropdown menu or enter the time manually into the input field.')

10**. type = 'datetime-local'** (This creates a text input field that allows users to select both a date and time.)

11. **type = 'month'** (This creates a text input field that allows users to select a month and year from dropdown menus.)

12. **type = 'week'** (This creates a text input field that allows users to select a week from a dropdown menu.)

13. **type = 'color'** (This creates a color picker that allows users to select a color.)

14. **type = 'range'**(This creates a slider that allows users to select a value from a range of values.)

15. **type = 'file'** (This creates a file input field that allows users to select a file from their device.
If we want to accept multiple files then we can use 'multiple' attribute, without the 'multiple' attribute, the file input field only allows the user to select a single file.
We can also specify the types of files that the user can select in a file upload field by using 'accept' attribute
(**Example**:- accept='image/*'(Here, this allows the user to select only image files.), accept='application/pdf' (Here, this allows the user to select only PDF files , etc.)))

16. **type = 'reset'** (This creates a reset button that allows users to reset the form to its initial state without reloading the page.)

17**. type = 'url'** (This creates an input field that is specifically designed for URLs, the input field will typically provide some form of validation and formatting for URLs.)

18. type = 'search' (This creates an input field that is specifically designed for search purpose.)

19. **type ='number'** (This creates a number input field where users can enter only numerical values.The input can be controlled using the 'min' and 'max' attributes.)

20**. type = 'checkbox'** (A checkbox allows the user to select one or more options from a list of options.Each checkbox is independent of other checkboxes on the same page, and multiple checkboxes can be selected at the same time.When a checkbox is selected, the value attribute associated with it is sent to the server when the form is submitted.)

21. **type = 'radio'** (A radio button allows the user to select only one option from a list of options.When a user selects a radio button, any previously selected radio button in the same group is automatically deselected. Each radio button in a group must have the same 'name' attribute, but different 'value' attributes.)

3. Name Attribute

1. The Name attribute specifies the name of the input field, which is used to identify it when the form is submitted to the server.

2. **Example**:- Username:<input type='text' name='username'> (The name attribute is set to 'username', which is the name that will be sent to the server when the form is submitted.)

4. Value Attribute

1. The Value attribute specifies the default value for the input field.

2. **Example**:- Username:<input type='text' name='username' value='John'> (The value attribute is set to 'John', so when the page loads, the text input will display 'john' as its initial value. The user can then modify the value as needed. When the form is submitted, the value entered by the user (which may or may not be 'john') will be sent to the server.)

5. Required Attribute

1. The Required attribute specifies that the user must fill out the input field before submitting the form.

2. **Example**:- <input type='text' name='username' required> (The required attribute is added to the text input. This means that the user must enter a value in the input field before they can submit the form. If they try to submit the form without entering a value, the browser will display an error message indicating that the input is required.)

6. Readonly Attribute

1. When an input element is set as readonly, it means that the user can view the value of the input field, but cannot modify it.

2. The readonly attribute can be used with input types such as text, number, date, time, and others.

3. When an input field is read-only, its value is still submitted with the form if the user does not modify it.

4. **Example**:- <input type='text' name='Name' value='John Doe' readonly> (In this case the readonly attribute is used to display the user's name but prevent any changes to it.)

7. Disabled Attribute

1. When an input element is set as disabled, it means that the user cannot interact with it or submit its value.

2. The disabled attribute can be used with input types such as text, number, date, time, and others, as well as with buttons.

3. When an input field is disabled, its value is not submitted with the form, even if it has a default value set.

4. **Example**:- <input type='text' value='John Doe' disabled> (In this case the disabled attribute is used to disable the input field i.e., the user cannot interact with it or submit its value.)

8. Placeholder Attribute

1. The Placeholder attribute specifies a hint to the user as to what kind of input is expected in the input field.

2. It is usually displayed as grayed-out text within the input field.

3. **Example**:- <input type='email' name='email' placeholder='Enter your email address'> (The placeholder text 'Enter your email address' is displayed in the input field as a hint to the user about what information should be entered in that field. Once the user begins typing in the input field, the placeholder text disappears.)

9. Autofocus Attribute

1. The Autofocus attribute is used with the <input> field to specify that the input field should have focus when the page loads.

2. This means that the cursor will automatically be placed in the input field, ready for the user to begin typing.

3. **Example**:- <input type='text' autofocus> (The text input field will have focus when the page loads.)

10. Min Attribute

1. The Min attribute is used with number, date, time, and range input types to specify the minimum allowed value for the input.

2. **Example**, <input type='number' min='10'> (It creates a number input field that only accepts values greater than equal to 10.)

11. Max Attribute

1. The Max attribute is used with number, date, time, and range input types to specify the maximum allowed value for the input.

2. **Example**, <input type='number' max='50'> (It creates a number input field that only accepts values less than equal to 50.)

12. Minlength Attribute

1. The Minlength attribute is used to specify the minimum number of characters that can be entered into an input field.

2. **Example**, <input type='text' minlength='5'> (It creates a text input field that only accepts input greater than equal to 5 characters.)

13. Maxlength Attribute

1. The Maxlength attribute is used to specify the maximum number of characters that can be entered into an input field.

2. **Example**, <input type='text' maxlength='25'> (It creates a text input field that only accepts input less than equal to 25 characters.)

14. Size Attribute

1. The Size attribute is used with text input fields to specify the visible width of the input field.

2. **Example**, <input type='text' size='30'> (It creates a text input field that is 30 characters wide.)

#Label Tag

1. The Label tag is used to bind/connect the text with the input field.

2. Label tag is a container tag.

3. It is inline-level element.

4. The text we want to connect should be written within the label tag.

5. In input field we have to provide id and that same 'id' we have to pass to the 'for' attribute of label tag.

6. This creates a connection between the two elements, so when the user clicks on the text, the respective input field is focused.

7. **Example**:- <label for='username'>Username:</label> <input type='text' id='username' name='username'> (Here, the label tag is used to bind the text 'username:' with the text input field.So, when the user clicks on the text 'username:' the input field is focused.)

#Fieldset Tag And Legend Tag

1. Fieldset Tag

1. The Fieldset tag is used to group the related form controls and create a box around the group

2. Fieldset tag is a container tag.

3. It is block level elemeent.

2. Legend Tag

1. The Legend tag is used to provide title/caption for the fieldset.

2. It is a container tag

3. It is block level element.

3. Example

1. <fieldset> <legend>Personal Information</legend> Name <input type='text' name='username'> <br><br> Email: <input type='email' name='email'> <br><br> </fieldset>

2. In the above example, the fieldset element groups the two form controls (name and email) together, and the legend element provides a title for the group(fieldset).

#Select Tag And Option Tag

1. Select Tag

1. The Select tag in HTML is used to create a drop-down list or a select list, where a user can select one or more options from a list of predefined options.

2. The select tag is a container tag.

3. It is an inline level element.

4. To mention the options we have to use option tag.

5. The content we want to display as an option should be written within the option tag.

2. Option Tag

1. The Option tag is used with select tag to provide the list of options from which a user can choose.

2. The option tag is a container tag.

3. It is an inline level element.

4. The content that we want as an option should be written within the option tag.

3. Example

1. <select name='fruits'> <option value='apple'>Apple</option> <option value='banana'>Banana</option> <option value='orange'>Orange</option> <option value='grape'>Grape</option> </select>

2. In this example, the <select> tag creates a drop-down list of fruits, with the name 'fruits'. The <option> tags provide the list of fruits that a user can choose from. Each <option> tag has a value attribute that specifies the value that will be sent to the server when the form is submitted. The text between the opening and closing <option> tags is the label that will be displayed in the drop-down list as an option.

4. Disadvantage of Select Tag

1. The main disadvantage of the select tag is that it limits the user's input options to the predefined list of options provided by the developer using the option tags.

2. The user cannot enter any text of their choice or customize the options available in the list.

#DataList Tag

1. The Datalist Tag is introduced in Html5.

2. The Html datalist tag is used to provide an autocomplete feature on the form element.

3. Datalist tag is a container tag.

4. It is block level element.

5. It is used to provide a list of predefined options to the users.

6. Datalist tag is used to create suggestion list or autocomplete list.

7. The <datalist> tag contains a set of <option> tags that define the options in the list.

8. We are binding the suggestion list with the input field, for this we have to provide 'list' attribute in the input tag and 'id' attribute in the datalist tag, this same 'id' we have to provide in the 'list' attribute of input tag.

9. Whenever the user inputs in the input field related suggestions are displayed.

10. The advantage of using the datalist tag is that it allows users to enter values that are not present in the options list.

11. **Example** :- Choose a fruit <input type ='text' list ='fruits' name ='my-fruit'> <datalist id='fruits'> <option value='apple'> <option value='banana'> <option value='orange'> <option value='cherry'> <option value='grape'> <option value='pineapple'> </datalist>

12. In the above example, the <datalist> tag with the id of 'fruits' provides a list of fruit options for the <input> tag. When the user types into the input field, the available options are displayed in a dropdown list, and the user can select an option or continue typing a custom value.

#Textarea Tag

1. The Textarea tag is used to create multi-line input field where the user can enter text.

2. Textarea tag is a container tag.

3. It is block level element.

4. We can control the dimensions of textarea using rows ans cols attribute.

5. The rows and cols attributes are optional and determine the visible height and width of the text area respectively.

6. **Syntax**:- rows = 'number of rows' and cols = 'number of columns'

7. Whatever content we write within the textarea tag is directly displayed inside the textarea.

8. **Example**:- <textarea name='message' rows='5' cols='40'>Default text inside the field...</textarea>

9. In the above example, the rows attribute specifies the height of the textarea, while the cols attribute specifies the width of the textarea. The default text is specified between the opening and closing <textarea> tags.

#Button Tag

1. The Button tag in HTML is used to create a clickable button on a web page.

2. Button tag is a container tag.

3. It is block level element.

4. It is often used to trigger some sort of action, like submitting a form or performing a JavaScript function when clicked.

5. **Example**:- <button>Submit</button> (It is used to create a submit button.)

#Difference Between <Input Type = 'Submit'> And <Button>Submit</Button>

1. <input type='submit'> is a self-contained tag/non-container tag whereas <button>Submit</button> is a container tag.

2. When we create submit button using <input> tag then on submit it automatically tries to connect to the server, if server is present it will send the data and reload the page whereas button tag doesn't try to conect to the server and it reloads the page when clicked.

3. <input type = 'submit'> is not fully compatible with Javascript whereas <button>Submit</button> is fully compatible with Javascript.

4. We can only mention text in input tag whereas we can mention any content like text, image, etc within the button tag.

#Iframe

1. An iframe or inline frame is an HTML element used to embed another HTML document within the current document.

2. To create iframe Html provides <iframe>...</iframe> tag.

3. It is a container tag.

4. It is an inline level element.

5. To mention the path of the document we have to use 'src' attribute.

6. We can also set the height and width of iframe using height and width attribute.

7. **Syntax**:- <iframe src='path-of-the-document' height='value' width='value'></iframe>

8. It is not recommended to use multiple iframes in our webpage because it slows down the webpage and hence creating a negative impact on the website's performance and user experience.

#Video Tag

1. The <video> tag was introduced in HTML5.

2. The <video> tag is used to embed or attach video to a web page.

3. It is a container tag.

4. It is block level element.

5. To mention the path of video we have to use <source> tag(Source tag is a non container tag.)

6. In <source> tag we have to use 'src' attribute in which path of video will be provided.(Syntax:- <source src='path-of-video'>

7. We can provide multiple sources using multiple <source> tags.

8. The content written within <video> tag will be displayed when <video> tag is not supported by the browser.

9. **Syntax**:- <video> <source src='path-of-video'> <source src='path-of-video'> Video tag is not supported by the browser. </video>

#Attributes Of Video Tag

1. **Controls** :- It is used to enable the controls of the video like play,pause,volume control, etc.

2. **Autoplay** :- It is used to automatically play the video when the page is loaded.

3. **Loop** :- It is used to play the video continuously in a loop i.e, when the video ends it is started again automatically.

4. **Muted** :- It is used to mute the audio of video.

5. **Poster**: - It is used to apply cover image or technically called as poster to the video before the video plays.

6. **Width**: - It is used to specify the width of the video element.

7. **Height** :- It is used to specify the height of the video element.

#Audio Tag

1. The <audio> tag was introduced in HTML5.

2. The <audio> tag is used to embed or attach audio to a web page.

3. It is a container tag.

4. It is block level element.

5. To mention the path of audio we have to use <source> tag(Source tag is a non container tag.)

6. In <source> tag we have to use 'src' attribute in which path of audio will be provided.(Syntax:- <source src='path-of-audio'>

7. We can provide multiple sources using multiple <source> tags.

8. The content written within <audio> tag will be displayed when <audio> tag is not supported by the browser.

9. **Syntax**:- <audio> <source src='path-of-audio'> <source src='path-of-audio'> Audio tag is not supported by the browser. </audio>

#Attributes Of Audio Tag

1. **Controls** :- It is used to enable the controls of the audio like play,pause,volume control, etc.

2. **Autoplay** :- It is used to automatically play the audio when the page is loaded.

3. **Loop** :- It is used to play the audio continuously in a loop i.e, when the audio ends it is started again automatically.

4. **Muted** :- It is used to mute the audio.

#Div And Span

1. Div and Span both are used to create containers so that we can group the content and applying styling.

2. Div is a block level element whereas Span is an inline level element.

#Class And Id

1. class and id both are attributes in HTML and are used to target the elements in a web page.

2. 'class' is used to target multiple elements whereas 'id' is used to uniquely identify the element.

3. One element can have multiple classes whereas one element should have only one id.

4. We can apply same class to multiple elements whereas id should be unique in a webpage and should not be applied to multiple elements.

5. **Sytax** :- class = 'c1, c2,c3,-----,cn' whereas id = 'id1'

#Details

1. The details element represents a disclosure widget technically called as Accordian.

2. It is a container tag.

3. It is inline level element.

4. The details tag is used for the content/ information which is initially hidden but could be displayed if the user wishesto see it.

#Summary

1. The summary element is used to provide caption or title to the details(Accordian).

2. It is a container tag.

3. It is inline level element.

4. We have to use the summary tag within the details tag.

#Example

1. <details> <summary>New Delhi</summary> It is the capital of India. </details>

2. In the above example, the <details> tag creates an Accordian and the <summary> tag is used to provide title to it. When the user clicks on the summary, the content inside the <details> tag is shown.

#Semantic Tags

1. Semantic tags are self explanatory tags.

2. Semantic tags in HTML are designed to give meaning to the content they contain.

3. **Example**:- <form> , <table>, <video>, <audio>, etc are semantic tags whereas <ol>,<tr>,<td>,etc are non semantic tags.

#Page Layout

1. It is the arrangement of visual elements on a webpage.

#Header Tag

1. The Header semantic tag was introduced in Html5.

2. Header tag is used to represent the header or topmost part of a website where we have logo and navigation links.

3. It is a container tag

4. It is block level element.

#Nav Tag

1. The Nav semantic tag was introduced in Html5.

2. All navigation links like home link, contact us, services, etc, should be written within the Nav tag.

3. It is a container tag

4. It is block level element.

#Section Tag

1. The Section semantic tag was introduced in Html5.

2. Section tag is used to define sections in a web page.

3. It is a container tag

4. It is block level element.

#Article Tag

1. The Article semantic tag was introduced in Html5.

2. Article tag is used to create divisions in section tag.

3. Here, we can write independent content.

4. It is a container tag

5. It is block level element.

#Main Tag

1. The Main semantic tag was introduced in Html5.

2. Main tag is used to define the main content of a web page.

3. The main content inside a section or article can be written within the main tag.

4. It is a container tag

5. It is block level element.

#Aside Tag

1. The Aside semantic tag was introduced in Html5.

2. The content which is not directly related to our website like advertisements should be written within the aside tag.

3. It is a container tag

4. It is block level element.

#Footer Tag

1. The Footer semantic tag was introduced in Html5.

2. Footer tag represents the footer/ bottom part of a web page or website.

3. It is a container tag

4. It is block level element.

#Contenteditable

1. contenteditable is an HTML attribute that allows users to edit the content of an element directly on the web page.

2. By default it is set to false i.e, user cannot edit or modify the content.

3. When the attribute is set to true, users can double-click on the element to edit/change the content.

4. **Example**:- <p contenteditable='true'>This text can be edited by the user.</p> (Here, the user can modify the content of the paragraph.)

#Accesskey

1. accesskey attribute is used to specify a shortcut key to activate or trigger something.

2. Example:- <a href='https://www.google.com' accesskey='k'>Google</a> (Here, this link can be triggered by using the accesskey 'k' along with the binding keys.)

3. The binding key for chrome and Edge is Alt + accesskey , binding key for firefox is Alt+Shift+accesskey.

4. **Note** :- These key bindings may vary depending on the operating system and web browser being used.

#Html5 Assignment

1. What is HTML and why do we use HTML?

2. Are HTML tags and elements the same thing?

3. What are tags and attributes in HTML?

4. What are void elements in HTML?

5. Why do we use forms in HTML? How to create a form in HTML?

6. What is the difference between <strong>, <b> tags and <em>, <i> tags?

7. What is the difference between <div> tag and <span> tag?

8. How many types of headings do we have in HTML?

9. How to link an image to a webpage in HTML?

10. What is the Internet and its history?

11. What is a client and a server?

12. What is page layout?

13. What is WWW, W3C, and the difference between them?

14. How to create a hyperlink in HTML? Explain with an example.

15. Difference between class and id in HTML?

16. What are HTML5 semantic tags?

17. Difference between HTML4 and HTML5?

18. Why do we use <!DOCTYPE html> in HTML documents?

19. How to create a drop-down list in HTML? Explain with an example.

20. How to create a suggestion list in HTML? Explain with an example.

21. What is an iframe? What are the disadvantages of using iframes?

22. How to add video and audio to a webpage in HTML?

23. What is a fieldset in HTML? Why do we use a fieldset?

24. What is the label tag in HTML? Explain with an example.

25. Difference between <button>Submit</button> and <input type='submit'> in HTML.