SILVER OAK UNIVERSITY
EDUCATION TO INNOVATION

ज्ञानं परमं भूषणम्

INDEX

NAME : Domadiya isha kishorbhai ENROLLMENT NO. : 067

No.	Date	Subject	Page	Marks	Signature
1	11/11/24	Assignment NO:-01	01		
2	13/11/24	Assignment NO:-02	17		
3	14/11/24	Assignment NO:-03	29		
4	16/11/24	Assignment NO:-04	46		

Assignment No :- 01

[unit : 1]

Q.1 Describe the basic structure of an HTML document.

Ans: The basic structure of an HTML (Hyper text Markup language) document consists of the following element

1. Doctype Declaration

<! DOCTYPE Html >

Declares the document type and version.

2. HTML Element

<html>
</html>

Root element of the HTML document.

3. Head Element

<head>
</head>

Contains metadata about the document.

4. Title Element

<title>
</title>

specifies the document title.

5. Body Element

<body>
</body>

contains the document's content.

Basic structure :-

```
<!DOCTYPE html>
<html>
<head>
  <title> Document Title </title>
</head>
<body>
<!--- content goes here -->
</body>
</html>
```

Explanation :

1. <!DOCTYPE html> declares the document t

2. <html> is the root element.
3. <head> contains metadata.
4. <title> specifies the document title.
5. <body> contains the document's content.

Optional elements :

- <meta> : Provides additional metadata.
- <link> : Links to external stylesheets or scripts.
- <script> : Embeds JavaScript code.
- <style> : Defines internal CSS styles.

Best practies :-

1. use lowercase tags and attribute names.

2. close tags in the reverse order they were opened.

3. use quotes around attribute values.

4. Validate your HTML code.

Q2 what are HTML Form elements and attributes? Provide Examples of commonly used form elements and their attributes.

Ans:- Form Elements :

1. Input
   - Type (text, password, email, number, checkbox,

radio, etc.)
- name
- placeholder
- value
- required

Example : <input type = "text" name = "username" placeholder = "Enter username" required >

1. Textarea.
- Rows
- cols
- Name
- Placeholder

Example : <textarea rows = "5" cols = "30" name = "description" placeholder = "Enter description" ></textarea>

1. Select
- Name
- multiple (optional)
- Size (optional)

Example :- <select name = "color" multiple size = "3" > <option value = "red" > Red </option> <option value = "green" > Green </option> <option value = "blue" > Blue </option> </select>

1. option
- value
- selected (optional)

Example : <option value = "red" selected > Red </option>

1. Button
- type (submit, reset, button)
- name
- value

Example : <button type = "submit" name = "submit" value = "submit"> submit </button>

1. label
- for (associates label with input)

Example : <label for = "username"> username : </label>

1. Form
- Action (URL to submit form data)
- method (GET, POST, etc)
- Enctype (optional)

Example :- <form action = "/submit" method = "post" enctype = "multipart/form-data"> <!--- Form elements here --> </

Attributes :-

1. Required :- Specifies that the field must be f

2. Placeholder :- Provides a hint to the user.

3. Disabled :- Disables the field.

4. Readonly :- Prevents user input.

5. Maxlength :- Sets maximum input length.

6. Min and Max :- Sets minimum and maximum for numeric inputs.

7. Pattern : Specifies a regular Expression pattern input validation.

Example form :-

<form action = "/submit" method = "post">
<label for = "username"> username : </ label>
<input type = "text" name = "username" placeholder = "Enter username" required >

```
<br>
<label for = "password"> password : </label>
<input type = "password"
name = "password" required >
<br>
<select name = "colour">
<option value = "red"> Red </option>
<option value = "green"> Green </option>
<option value = "blue"> Blue </option>
</select>
<br>
<button type = "submit"
name = "Submit"
value = "Submit"> Submit </button>
</form>
```

This example demonstrates various form elements and attributes.

Q3 Discuss the importance of browser support in HTML development. How can developers ensure cross-browser compatibility ?

Ans: Importance of Browser Support :

1. Reach wider audience : Support multiple browsers to cater to diverse user base.

2. Consistent user experience :- Ensure uniform layout, functionality, and performance.

3. Enhanced credibility : Cross - browser compati
builds trust and professionalism.

4. Improved accessibility : Ensure equal access for
user with disabilities.

5. Reduced maintenance : Test and fix issues on
rather than repeatedly.

Challenges in cross - Browser compatibility :

1. Browser rendering differences.
2. CSS and Javascript implementation variation
3. HTML 5 feature support disparities.
4. Device and screen size diversity
5. Operating System inconsistencies.

Ensuring Cross - Browser compatibility :-

Pre - Development :

1. Define target browsers and devices.

2. Research browser market share and then

3. Set up testing environments.

During Development :

1. Use HTML 5 and CSS 3 standards

2. Validate code using w3c validators

3. Test with browser - specific tools (e.g. Chrome Dev Tools)

4. Utilize cross - browser testing frameworks (e.g. selenium.)

5. Implement responsive design

Post - Development :-

1. test on multiple browsers and devices
2. conduct user testing and feedback.
3. Monitor browser updates and adapt
4. continuously iterate and refine

tools for cross - Browser testing

1. BrowserStack
2. CrossBrowser testing
3. Sauce Labs
4. Selenium
5. VirtualBox

Best practices :-

1. Separate CSS and Javascript files
2. use CSS prefixes (-webkit- moz ms)
3. Avoid browser specific hacks

4. use feature detection (e.g Modernizr)
5. Document browser-specific issues

Front end frameworks for cross Browser Compat

1. Bootstrap
2. Foundation
3. Material-ui
4. React
5. Angular

By prioritizing cross-browser compatibility, de ensure a seamless user experience, enhance a ility, and reduce maintenance costs.

Q4 How can background colours and images be a to an HTML webpage? Provide code Exameles demonstrating both techniques?

Answer Background colors and images can be added HTML weppage using css (cascading style sheets

Method 1 : inline CSS

Add background color or image directly to HTML element using the style attribute.

Background color :

<div style = " background-color :

url ('image.jpg'); height:
500px;"> content

Method 2: Internal CSS

Define styles in the <head> section of the HTML document.

```
<head>
  <style>
  body {
    background-color: #F2F2F2;
  }
  .background-image {
    background-image:
  url ('image.jpg');
    height: 500px
  }
  </style>
</head>
<body>
<div
class = "background-image">> content </div>
</body>
```

Method 3: External CSS

Link an External CSS file to the HTML document.

style.css:

```
body {
    background-color : #F2F2F2 ;
}

.background-image {
    background-image :
    url('image.jpg') ;
    height : 500px ;
}
```

index.html :

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

<body>
  <div
  class = "background-image"> content </div>
</body>
```

Background image properties :

- background-repeat : repeat, no-repeat, repeat repeat-y.
- background-position : center, top, bottom, right.

Example :-

```
• background-image {
  background-image :
url('image.jpg');
  background-repeat : no-repeat ;
  background-position : center ;
  background-size : cover;
}
```

Responsive Background image :-

```
• background-image {
  background-image :
url('image.jpg');
  background-size : 100% 100%;
  background-repeat : no-repeat;
  height : 100vh;
```

iv) These methods demonstrate how to add background colours and images to an html webpage using CSS.

Q.5 Explain div and span tag in details.

Ans:- div and span are two essential HTML elements used for structuring and styling content.

Div Tag :

div stands for "division" or "container". It's a

Example :-

```
• background-image {
  background-image:
url('image.jpg');
  background-repeat: no-repeat;
  background-position: center;
  background-size: cover;
}
```

Responsive Background Image :-

```
• background-image {
  background-image:
url('image.jpg');
  background-size: 100% 100%;
  background-repeat: no-repeat;
  height: 100vh;
```

These methods demonstrate how to add background colours and images to an html webpage using CSS.

Q.5 Explain div and span tag in details.

Ans: div and span are two essential HTML elements used for structuring and styling content.

Div Tag :

div stands for "division" or "container". It's a

block-level element used to group elements to
for:

1. layout purposes
2. styling
3. semantic meaning

Characteristics :-

1. Block-level element (occupies full width)
2. Can contain other elements (including text, i
   etc.,)
3. Can be styled with CSS.
4. Can have attributes (e.g. id, class, style)

Example :

```
<div class = "container">
    <h2> Heading </h2>
    <p> Paragraph text </p>
    <img src = "image.jpg" alt = "Image">
</div>
```

Span Tag :-

span is an inline element used to :

1. Apply styles to text
2. Provide semantic meaning
3. Group text or inline elements

characteristics :-

1. Inline element (only occupies space needed)
2. can contain text or other inline elements.
3. Can be styled with CSS
4. Can have attributes (e.g. id, class, style)

Example :-

 This is a important announcement.

Key differences :

1. Display type: div is block-level, while span is inline.

2. Purpose : div is for layout and containerization, while span is for text-level styling and semantics.

When to use each :

1. Use div for :
- Creating layout containers
- Grouping elements for styling
- Creating sections or divisions

2. use span for :
- Grouping inline elements
- Applying styles to text

Best Practices :-

1. use semantic HTML elements (e.g. header, footer, nav) instead of div when possible)

2. use span for text level styling, rather than div

3. Avoid using div or span unnecessarily ; use other semantic elements instead.

By understanding the differences and uses of div & span you can create more structured, accessible and maintainable HTML documents.

Assignment No :- 02
[unit :- 02]

Q.1 Explain the difference between headings (h1 - h6) and paragraphs (<p> tag) in HTML.

Answer

Headings (h1-h6)

Headings are used to define the title or heading of a section or page. There are six levels of headings, each representing a different level of importance :-

1. h1 : Most important, typically the main title of the page.
2. h2 : Secondary title, subdividing the content.
3. h3 : Tertiary title, further subdividing the content.
4. h4, h5, h6 : less important, used for subheadings.

Characteristics :-

1. Displayed in a larger font size and bold weight.
2. Typically used for titles, headings and subheadings

3. Search engines use headings to understand page structure.

4. Screen readers announce headings to user

Example :-

<h1> main Title </h1>
<h2> Secondary Title </h2>
<h3> Tertiary Title </h3>

Paragraphs (P)

Paragraphs are used to define a block of text forms a coherent thought or idea.

Characteristics :-

1. Displayed in a standard font size and weight
2. Typically used for body text, articles and co
3. can contain text, images, links, and other ini elements.
4. Automatically adds a line breaks after the e

Example :-

<P> This is a paragraph of text. It can multiple sentences. </P>

Key differences :-

1. Purpose :- Headings define titles, while paragraphs define body text.

2. Font size and weight :- Headings are larger and bolder, while paragraphs are standard.

3. Structure :- Headings are used to create a hierarchical structure, while paragraphs are used to present continuous text.

4. Search engine optimization (SEO) : Headings are more important for SEO than paragraphs.

★ Best Practices :

1. use headings to create a clear hierarchical structure

2. use paragraphs to separate blocks of text.

3. Avoid using headings for styling text ; use CSS instead

4. use only one h1 per page, and use subsequent headings (h2 - h6) to create subheadings.

By understanding the differences between headings and paragraphs, you can create well-structured readable and accessible HTML documents.

Q.2 How can you create a horizontal rule (<hr>) in HTML? Describe its purpose and provide an example of its usage.

Answer Purpose :-

The <hr> tag serves several purposes :-

1. Visual separation :- Divides content into distinct sections.

2. Structural separation :- Indicates a break in the content theme.

3. Layout control : Helps organize content and whitespace.

Syntax :

<hr>

Attributes :-

1. align :- (deprecated) : left, right, center
2. size :- (deprecated) : specifies line thickness (in
3. width : (deprecated) : specifies line width (in = 50 %)

Note :- The align, size, width, and color are deprecated in HTML 5.

★ CSS styling :

```
hr {
  border : none ;
  border-top : 1px solid #ccc ;
  width : 50 % ;
  margin : 20px auto ;
}
```

Example usage :-

```
<h1> Introduction </h1>
<p> Welcome to our website. </p>
<hr>
<h2> Features </h2>
<p> Our product offers.... </p>
<hr>
<h3> Contact us </h3>
<p> Get in touch with us. </p>
```

In this example, the <hr> tag separates the introduction, features and contact sections.

★ Best Practices :

1. Use <hr> sparingly to avoid clutter.
2. Use CSS to customize appearance.
3. Ensure accessibility by providing alternative text for screen readers.

★ Semantic Alternatives :-

Consider using semantic elements instead of

1. <section> : Defines a self-contained section

2. <article> : Defines an independent piece of

3. : with CSS : Creates a styled container.

By using the <hu> tag effectively, you can en your website's readability and structure.

Q.3 Discuss the use of subscript <sub> and superscr <sup> tags in HTML. Provide Examples demons their applications in text formating.

Answer Subscript (<sub>)

The <sub> tag reduces the font size and lo the baseline of the text, making it appear a subscript.

Example :

H2O (water) <sub>2K I sub)

Result : H2O (water)$_2$

Applications :

1. chemical formulas (e.g. H2O)
2. Mathematical Expressions (e.g. x_3)
3. Footnotes or citations

superscript (<sup>)

The <sup> tag increases the font size and raises the baseline of the text, making it appear as superscript.

Example :

2 <sup>3 <|sup> = 8

Result : $2^3 = 8$

Applications :-

1. Mathematical Expressions (e.g. 2^3)
2. Exponts (e.g. x^2)
3. Dates (e.g. 24 <sup> th <|sup> January)
4. copyright symbols (e.g. & copy; <sup> 2022 <|sup>)

Comparison :-

Tag	Effect	Example
<sub>	subscript	H2O (water) <sub>2<|sub>
<sup>	superscript	2 <sup> 3 <|sup> = 8

Styling Alternatives :-

Instead of using <sub> and <sup> tags, you use CSS to achieve similar effects :

- Subscript :- font-size : smaller ; vertical-alig sub ;
- Superscript :- font-size : larger ; vertical-alig Super ;

Best Practices :-

1. use <sub> and <sup> tags for semantic m ng, not just visual styling.
2. use CSS for custom styling and layout cont

Browser Support :-

Both <sub> and <sup> tags are widely supported across modern browsers, including.

- Google Chrome
- Mozilla Firefox
- Safari
- Microsoft Edge

By using the <sub> and <sup> tags effectively you can enhance your website's text formatt

Q.5 compare and contrast unordered lists (<ul>), ordered lists (<ol>), and definition lists (<dl>) in HTML.

Ans unordered lists (<ul>)

- Purpose :- Present a list of items without numerical ordering.
- Characteristics :-
  - Bullet points (dics, circle, or square)
  - No inherent order
  - Items are enclosed in <li> tags.
- Example

```
<ul>
  <li> Apple </li>
  <li> Banana </li>
  <li> cherry </li>
</ul>
```

- Result :

- Apple
- Banana
- Cherry

Ordered list (<ol>)

- Purpose : Present a list of items with numberical

ordering.

- characteristics

- Numbered list items (1, 2, 3 etc.)
- Items are enclosed in <li> tags.
- Supports start attribute to specify starting n

* Example

```
<ol>
  <li> Apple </li>
  <li> Banana. </li>
  <li> Cherry </li>
</ol>
```

Result :

1. Apple
2. Banana
3. Cherry

Definition lists (<dl>)

- Purpose : Present a list of terms with definitio
- characteristics :-
  - Consists of <dt> (term) and <dd> (defi tags.
  - No inherent order
  - Supports multiple definitions per term.

Example :-

<dl>
<dt> Fruit </dt>
<dd> A sweet, edible plant product. </dd>
<dt> Vegetable </dt>
<dd> A plant product used in cooking </dd>
</dl>

Result

Fruit
: A sweet, edible plant product
Vegetable
: A plant product used in cooking

Comparison

List Type	Purpose	Ordering	Tags
	unordered	No order	
	ordered	Numerical	
<dl>	Definition	No order	<dt>, <dd>

Contrast

- and <ol> use <li> tags, while <dl> uses <dt> and <dd> tags.
- has inherent numerical ordering, while

 and <dl> do not.

- <dl> supports multiple definitions per term, wh
 and <ol> do not.

Best practices :-

1. use <ul> for lists without numerical ordering

2. use <ol> for lists with numerical ordering.

3. use <dl> for definition lists.

4. Ensure accessibility by providing alternative for screen readers.

5. use css for custom styling and layout contro

By understanding the differences between unordered lists, ordered lists, and definition lists, you can eff present information on your website.

Assignment no :- 03

[unit :- 03]

Q: Explain the structure of an HTML table. Provide an example of a basic structure.

Ans Basic Structure :-

```
<table>
  <tr>
    <th> Header 1 </th>
    <th> Header 2 </th>
  </tr>
  <tr>
  <td> Data 1 </td>
  <td> Data 2 </td>
  </tr>
  <tr>
  <td> Data 3 </td>
  <td> Data 4 </td>
  </tr>
</table>
```

Elements :-

1. <table> : Defines the table.
2. <tr> : Defines a table row.
3. <th> : Defines a table header cell.

★ Table structure :-

1. Header Row :- The first row, containing <th> e defines the column headers.

2. Data Rows :- Subsequent rows, containing <td> e define the data.

3. Columns :- Each <th> or <td> element represents columns.

4. cells :- Each <th> or <td> element represents single cell.

Attributes :-

1. border :- specifies the border width and style.

2. cellpadding :- specifies the space between cel content and borders.

3. cellspacing :- specifies the space between cells

4. width and heigh :- specify the table dimension

example with Attributes :

<table border = "1" cellpadding = "5"
cellspacing "0" width "50%" >

```
<tr>
  <th> Header 1 </th>
  <th> Header 2 </th>
</tr>
<tr>
  <td> Data 1 </td>
  <td> Data 2 </td>
</tr>
<tr>
  <td> Data 3 </td>
  <td> Data 4 </td>
</tr>
</table>
```

Additional elements :

1. <caption> : Defines the table caption.

2. <col> and <colgroup> : Define column groups and styles.

Best practices :-

1. use semantic HTML table structure.
2. use CSS for styling and layout control.
3. Ensure accessibility by providing alternative text for screen readers.
4. use scope attribute to define header scope.

By understanding the basic structure and elem of an HTML table, you can crete well-struc and accessible tables.

Q.2 How can you align a table and its cell content HTML? Describe the CSS properties or HTML attrib used for alignment, and provide Examples demon their application.

Answer

HTML Attributes (Deprecated) :

Although deprecated, some HTML attributes are supported for backward compatibility.

- align (left, right, center)
- valign (top, middle, bottom)
- char (character, alignment)
- charoff (character offset)

Example :

```
<table align = "center">
  <tr>
    <td valign = "top"> Top-aligned </td>
    <td
valign = "middle"> Middle-aligned </td>
    <td
valign = "bottom"> Bottom-aligned </td>
  </tr>
</table>
```

CSS properties :

Recommended approach using CSS properties.

* text-align (left, right, center, justify)
* vertical-align (top, middle, bottom, baseline)
* padding (cell padding)

Examples :

Table Alignment :

```
table {
    margin : 0 auto ; /* center table horizontally */
}
```

Cell content Alignment :-

```
td {
    text-align : center ; /* Horizontal alignment */
    vertical-align : middle ; /* Vertical alignment */
}
```

Specific cell Alignment :

```
td : nth-child (1) {
    text-align : left ;
    vertical-align : top ;
}
```

using css classes :

```
• centered-table {
    margin: 0 auto;
  }
• centered-cell {
    text-align: center;
    vertical-align: middle;
  }

<table class="centered-table">
  <tr>
    <td class="centered-cell">centered cell</td>
  </tr>
</table>
```

Best practices :-

1. use css properties instead of HTML attributes.
2. Define table and cell style in External CSS fi
3. use semantic html table structure.
4. Ensure accessibility by providing alternative te for screen readers.

By applying these alignment techni you can effectively control the layout of your and cell contents.

Q3 Describe the process of nesting tables in HTML, provide an Example illustrating nested tables.

Process :-

1. Create the outer table with <table>, <tr>, and <td> elements.
2. within a table cell (<td>) add another <table> element.
3. Define the inner table's structure with <tr> and <td> elements.
4. close the inner table (</table>) within the outer table cell.
5. close the outer table (</table>)

Example :-

```
<table border = "1">
<tr>
  <th> Name </th>
  <th> Address </th>
</tr>
<tr>
  <td> John Doe </td>
  <td>
    <table border = "1" >
    <tr>
```

```
        <th> Street </th>
        <th> City </th>
        <th> State </th>
    </tr>
    <tr>
        <td> 123 Main St </td>
        <td> Anytown </td>
        <td> CA </td>
    </tr>
</table>
</td>
</tr>
<tr>
    <td> Jane Smith </td>
    <td>
        <table border = "1">
        <tr>
            <th> Street </th>
            <th> City </th>
            <th> State </th>
        </tr>
        <tr>
            <td> 456 Elm St </td>
            <td> Othertown </td>
            <td> NY </td>
        </tr>
        </table>
    </td>
</tr>
</table>
```

Q

Result :- A table with columns, where the second column contains another table with three columns.

Tips :-

1. Use semantic HTML table structure.
2. Ensure accessibility by providing alternative text for screen readers.
3. Use CSS for styling and layout control.
4. Avoid excessive nesting, as it can affect readabili

Common use cases :-

1. Displaying complex data relationships
2. Creating master-details views.
3. Organizing hierarchical information.

Best practices :-

1. Keep nested tables simple and readable.
2. Use clear and concise table headers.
3. Avoid using nested tables for layout purposes.

By nesting tables effectively, you can present complex data in a clear and organized manner.

Q.4 What are frames in HTML, and how are they created? Explain the purpose of the <frameset> and <frame> tags.

Answer Process :

1. Create the outer table with <table>, <tr>, and <td> elements.
2. Within a table cell (<td>), add another <table> element.
3. Define the inner table's structure with <tr> and <td> elements.
4. Close the outer table (</table>)

Example :-

```
<table border = "1">
 <tr>
    <th> Name </th>
    <th> Address </th>
 </tr>
 <tr>
    <td> John Doe </td>
    <td>
 <table border = "1">
 <tr>
 <th> street </th>
```

```
<th> City </th>
<th> State </th>
</tr>
<tr>
<td> 123 main St </td>
<td> CA </td>
</tr>
</table>
</td>
</tr>
<tr>
<td> Jane Smith </td>
<td>
<table border = "1">
<tr>
<th> Street </th>
<th> City </th>
<th> State </th>
</tr>
<tr>
<td> 456 Elm St </td>
<td> Othertown </td>
</tr>
</table>
</td>
</tr>
</table>
```

Result :-

A table with two columns, where the second colu
contains another table with three columns.

Tips :

1. use semantic HTML table structure.
2. Ensure accesibility by providing alternative
for screen readers.
3. use CSS styling and layout control.
4. Avoid Excessing nesting, as it can affect

Common use cases :

1. Displaying complex data relationships.
2. Creating master - details views.
3. organizing hierarchical information.

Best practices :-

1. Keep nested tables simple and readable.
2. use clear and concise table headers.
3. Avoid using nested tables for layout purpos

By nesting tables effectively, you can presen
complex data in a clear and organized man

Additional Examples :

• Nested tables with multiple levels :

```
<table>
<tr>
<td>
<table>
<tr>
<td>
<table>
<tr>
<td> Deeply nested
table </td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
```

• Nested tables with different orientations :

```
<table>
<tr>
<td>
<table style = " width : 100.0% ">
```

Additional Examples:

1. Nested tables with multiple levels:

```
<table>
<tr>
<td>
<table>
<tr>
<td>
<table>
<tr>
<td> Deeply nested
table </td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
```

2. Nested tables with different orientations:

```
<table>
<tr>
<td>
<table style = " width : 700px ">
```

```
<tr>
  <td> Horizontal nested
table </td>
    <tr>
    </table>
  </td>
</tr>
<tr>
  <td>
    <table style = "heigh: 100%" >
    <tr>
      <td> vertical nested
table </td>
      <tr>
      </table>
    </td>
  </tr>
</table>
```

Q.5 Discuss the concept of applying hyperlink targets to frames.

Answer Frame targets:

In HTML, you can specify a target attribute for hyperlinks (<a>) to determine where the linked page opens.

1. _blank : opens in a new window.
2. _self : opens in the same frame (default)
3. _parent : opens in the parent frame.
4. _top : opens in the full browser window.

Example:

 open in mainframe

Frames Names :

To use frame targets, you must assign names to your frames using the name attribute.

<frame src = "page1.html" name = "mainframe" >

Targeting frames :

1. Intra-frame targeting : linking within the same frame set.

 open in mainframe

1. Inter-frame targeting : linking within same frame set.

<a href = "page2.html"
target = "sideBar"> open in sideBar frame

1. Parent frame targeting : linking to the parent frame.

<a href = "page2.html"
target = "parent"> open in parent frame

Best practices :-

1. Use descriptive frame names
2. Ensure frame names match target target attribute.
3. Test frame targeting thoroughly.

Alternatives to frame targets.

1. CSS layout
2. Javascript libraries (e.g. Ajax)
3. HTML5 semantic elements.

Browser Support :

- Google Chrome
- Safari

conclusion

Applying hyperlink targets to frames enables precise control over where linked pages open within a framed website. Understanding frame targets and their usage is essential for creating user-friendly and accessible framed websites.

Assignment :- 04
unit :- 04

Q.1 Explain the different ways to create hyperlink in HTML :

Answer 1. Absolute URL Hyperlink

 visit Example

2. Relative URL Hyperlink

 About us

3. Anchor Link (Intra-page linking)

 Back to top

4. Email link

 Email us

5. Phone Number link

 call us

6. Link to a specific section (ID)

 section 1

 section 1
content

3. link to a file Download.

<a href = "document.pdf"
download > Download pdf

4. External link with target Attribute.

<a href = "(link unavilable)
target = "_blank"> open in new tab

5. Internal link with title Attribute

learn more

HTML 5 Semantic link elements.

<nav>
<a href "#"> navigation link
</nav>

button link

 click
here

or

<button onclick = "location.href = '#'" > click here </button>

Image link

<img src = "image.jpg" alt = "Image Description">.

★ Best practices

1. use descriptive link text.
2. Ensure accessibility with alt and title attri
3. use relative URLs for internal links.
4. Test links thoroughly

★ Browser Support

- Google Chrome
- Mozilla Firefox
- Safari
- Microsoft Edge
- Internet Explorer (IE + 3)

Q.2 How can you style hyperlinks in HTML to ch their color?

Answer Inline styling

 Blue link

Internal CSS

<head>
<style>
a {
color : red ;
}
</style>
</head>
<body>
 Red link
</body>

External CSS

Create a CSS file (style.css) and link it to your HTML file :

style.css

a {
color : green ;
}

index.html

<head>

<link rel = "stylesheet"
type = "text/css" href = "style.css">
</head>
<body>
 Green link
</body>

Pseudo-classes

Style different states of a hyperlink :

```
a:link {
    color : blue ; /* unvisited link */ } :

a:visited {
    color : purple ; /* visited link */ }

a:hover {
    color : red ; /* Hovered link */ }

a:active {
    color : green ; /* Active link */ }
```

Class-Based Styling

Apply styles to specific hyperlinks :

 Blue link

```
<style>
    .blue-link {
        color: blue;
    }
</style>
```

- Best practices

1. Use External CSS files for better organization.
2. Use meaningful class names
3. Avoid inline styling
4. Test for accessibility.

Browser Support

- Google chrome
- Mozilla firefox
- Safari
- Internet Explorer (IE11)

Q3 Describe the process of inserting an image on a webpage using the <img> tag in HTML.

Answer Basic syntax

```
<img src = "image_url"
alt = "image_description">
```

Attributes :

1. src : Specifies the image URL.
2. alt : Provides alternative text for accessibi
3. width : height : Define image dimensions
4. title : Adds a tooltip with image information

Example :

<img src = " logo.png " alt = " company logo " width = " 200 " height = " 100 ">

Image sources :

1. local files : src = " image.jpg "
2. web URLs : src = " (link unavailable) "
3. Relative paths : src = " ... \images\image.jpg "

Image formats :

1. JPEG (.jpg, .jpeg)
2. PNG (.png)
3. GIF (.gif)
4. SVG (.svg)
5. BMP (.bmp).

Best practices :

1. use descriptive alt text.
2. Optimize image file size.

3. specify image dimensions.
4. use relative paths for local images.

* common issues:

1. broken image links.
2. image not displaying due to incorrect file path.
3. image dimensions not specified.

* browser support:

The <img> tag is supported by all modern browsers including:

- Google Chrome
- Safari
- Internet Explorer (IE 11+)

Additional Tips:

1. Use CSS to style images.
2. Use JavaScript to create interactive images.

semantic HTML.

Use the <figure> and <figcaption> tags to provide context for images:

<Figure>
<img src = "image.jpg" alt = "Image Description">
<figcaption> Image caption </figcaption>
</Figure>.

Q5 Explain how multimedia content can be embedded on a webpage using HTML5 <video> and <audio> tags.

Answer:- Video Tag:

<video width = "640" height = "480" controls>
<source src = "video.mp4" type = "video/mp4">
Your browser does not support the video tag.
</video>

Attributes :-

1. width and heigh : Define video dimensions.
2. controls : Display video controls (play, pause, e
3. autoplay : play video automatically.
4. loop: loop video playback.

Audio Tag :

```
<audio controls>
    <source src = "audio.mp3"
type = " audio | mp3">
Your browser does not support the audio
tag.
</audio>
```

Attributes :

1. controls : Display audio controls (play, pause)
2. loop : loop audio playback

source options :

1. mp4 (video)
2. webm (video)
3. ogg (video and audio)
4. wav (audio)

Browser support

The <video> and <audio> tags are supported by

- Google Chrome
- Safari

Fallback options :

1. Flash Player
2. Third-party

Best Practices :-

1. Use semantic HTML structure.

Additional Features

1. Custom controls using JavaScript.
2. Adaptive bitrate streaming

Example with Subtitles :-

```
<video width = "640" height = "480" controls>
    <source src = "video.mp4"
    <track src = "subtitles.vtt"
kind = "subtitles" srclang = "en"
label = "English">

    Your browser does not support the
    video tag.
</video>
```

By using HTML5's multimedia tags, you can easily embed engaging content on your web pages.