

Mod 1

Q1) Explain <audio> and <video> controls of HTML 5 with an appropriate example.

⇒ (i) <audio>

- (i) The audio element in HTML 5 is used to embed sound content such as music, podcasts or sound effects on a web page.
- (ii) The controls attribute provides a user-friendly interface to control the playback of the audio file without the need for custom javascript.

(iii) Features of <audio> with control

- (a) Play / pause button : Allows user to play or pause the audio .
- (b) Progress bar : shows the current playback position and allows user to seek through the audio .
- (c) Volume control : Enables volume adjustments directly within the UI .
- (d) Cross - browser support : works seamlessly on all major modern browsers .

(iv) Example :

< audio controls>

<source src = "example.mp3" type = "audio/mpeg" >

<source src = "example.ogg" type = "audio/ogg" >

Your browser does not support the audio element

</audio>

- (1) The controls attribute enables an interactive UI .
- (2) The <source> element allows multiple formats .
- (3) Fallback text is displayed if your browser does not support the audio .

② <video>

- (i) The <video> element is used to embed video content in web pages
- (ii) The controls attribute offers a complete UI for users to interact with the video content directly.
- (iii) Features of <video> control
 - (a) Play / pause button: Controls video playback.
 - (b) Progress bar: Shows the playback position and allows seeking.
 - (c) Volume control: Adjusts sound levels.
 - (d) Fullscreen toggle: Lets the users watch the video in fullscreen mode.
- (e) Multiple source formats: Ensures compatibility across browsers.

④ Example:

```
<video controls width="640" height="360">  
  <source src="example.mp4" type="video/mp4">  
  <source src="example.ogv" type="video/ogg">  
Your browser does not support the video element.  
</video>
```

- (1) The controls attribute enables the default player interface for the video.
- (2) The width and height attributes define the size of the video player.
- (3) The <source> elements allows multiple formats of video.
- (4) Fallback text appears if the browser does not support <video>.

(Q2) Explain the working of rowspan and colspan in an HTML table with a suitable example.

⇒ (1) In HTML, the `<table>` element is used to create tabular data, and the attributes `rowspan` and `colspan` are used to merge cells.

(2) They merge the cells either horizontally or vertically.

(3) These attributes make tables more organized by merging cells.

(4) `rowspan` attribute.

(i) The `rowspan` attribute merges a cell downwards by combining it with cells below it.

(ii) It is added to a `<td>` or `<th>` tag, and it shows how many rows the cell should cover.

(iii) Syntax:

`<td rowspan = "n"> Content </td>`

Here, 'n' is the number of rows the cell will cover.

(iv) Example:

`<table border = "1">`

`<tr>`

`<th> Item </th>`

`<th> Description </th>`

`<th> Price </th>`

`</th>`

`<td rowspan = "2"> Laptop </td>`

`<td> Model A </td>`

`<td> $1000 </td>`

`</td> (more) tags whole row all was merged (1)`

`<tr> (more) (2) only first of them`

`<td> Model A </td>`

`<td> $1000 </td>`

`</tr> (more) (3) whole row`

`</table> vi - going whole - text set price`

`Output:`

Item	Description	Price
Laptop	Model A	\$1000
Laptop	Model B	\$1200

(5) colspan attribute

(i) The colspan attribute merges a cell sideways if with cells in the same row.

(ii) It is added to a `<td>` or `<th>` tag, and its value shows how many columns the cell should cover

(iii) Example:

`<table border = "1">`

`<tr>`

`<th colspan = "2"> Product details </th>`

`<th> Price </th>`

`</tr>`

`<tr>`

`<td> Item </td>`

`<td> Description </td>`

`<td> $500 </td>`

`</tr>`

`</table> (more) (2) whole row`

(iv) Output:

Product Details	Price
Item	Description
	\$500

(g3) Explain how the shadow effect can be applied to text using CSS with a suitable example.

⇒ (1) The shadow effect can be applied to text using the text-shadow property.

(2) It enhances the appearance of text by creating a shadow behind it.

(3) Syntax :

`text-shadow: offsetX offsetY blurRadius color;`

(i) `offsetX` : Specifies the horizontal position of the shadow.

(ii) `offsetY` : Specifies the vertical position of the shadow.

(iii) `blurRadius` : Defines the softness or spread of the shadow.

(iv) `color` : Specifies the color of the shadow.

(4) Example :

<html>

<head> <title> "S" is regular </title>

</head>

<body>

<style>

h1 {

text-shadow: 2px 2px 4px gray;

g

</style>

<h1> Shadow effect example </h1>

</body>

</html>

(i) CSS rule : `text-shadow: 2px 2px 4px gray;`

2px : The shadow is moved 2 pixels to the right horizontally

2px : The shadow is moved 2 pixels downward vertically

4px : The blur radius creates a soft shadow effect of 4px

gray : The shadow color is gray

(ii) Output : The text "shadow effect example" will display a gray shadow.

(5) Use cases :

- (i) Headings
- (ii) Highlighting
- (iii) Styling

(Q4) Explain any 5 semantic tags of HTML5 with an example.

- ⇒ (1) Semantic tags in HTML5 provides meaningful structure to the web content.
- (2) It makes it easier for developers, browsers, and search engines to understand the purpose of the content.
- (3) Five semantic tags in HTML:

- (i) `<header>`
- (a) The `<header>` tag defines the introductory section of a webpage or a section within it.
- (b) It typically contains headings, navigation links, or logo elements.

(C) Example:

`<header>`

`<h1> Welcome </h1>`

`<nav>`

``

```
<li><a href = "#home"> Home </a> </li>
<li><a href = "# contact"> Contact </a> </li>
<li><a href = "# about"> About </a> </li>
```

``

`</nav>`

`</header>`

(ii)

`<nav>`

- (a) The `<nav>` tag is used to define a section of navigation links.

- (b) It is commonly used for menus or table of contents.

(C) eg: `<nav>`

``

```

<li><a href = "#services">Services </a></li>
<li><a href = "#portfolio">Portfolio </a></li>
</ul>

```

(ii) `<section>`

(a) The `<section>` tag represents a thematic grouping of content.

(b) Each group should ideally have a heading and be logically distinct from other sections.

(c) eg: `<section>`

`<h2> Our services </h2>`

`<p> Web dev, app dev, etc </p>`

`</section>`

(iv) `<article>`

(a) The `<article>` tag represents self-contained content that can stand independently such as blogs, news articles, etc.

(b) eg: `<article>`

`<h3> How to learn HTML </h3>`

`<p> HTML is foundation of web development </p>`

`</article>`

(v) `<footer>`

(a) The `<footer>` tag represents the footer section of a document or a section.

(b) It usually contains copyright information, links to policies and contact details.

(c) eg: `<footer>`

`<p> 2024 website. All Rights reserved. </p>`

`<p> Privacy Policy | Terms of use </p>`

`</footer>`

(Q5) What is HTTP? Describe the structure of HTTP request and response message.

- ⇒ (1) HTTP also known as hypertext transfer protocol is used for transferring data over the web.
- (2) HTTP operates on a request-response model, where clients send request to the servers and servers respond with the requested data.
- (3) Structure of HTTP request messages

(i) An HTTP request message is sent by a client to request data or perform an action on a server.

(ii) It consists of three parts:

(a) Request line

(1) The request line contains the action to be performed, the resource & the protocol version. (2) It contains:

HTTP method: The type of request (eg: GET)

URL: The path to the resource on the server

HTTP version: The version of HTTP being used.

(3) Eg: GET /index.html HTTP/1.1

(b) Headers

(1) Headers provide additional information about the request, such as metadata and client details.

(2) Common headers:

Host: Specifies the server address

User-Agent: Identifies the client making the request.

(3) Example: Host: www.github.com

User-Agent: Mozilla/5.0

④ Body

(i) The body contains data sent by the client to the server, typically in a POST or PUT requests.

(ii) For eg: While submitting a form, the form data is included in the body.

username = Kanistubh & password = 1234

⑤ Structure of HTTP response messages

(i) An HTTP response message is sent by the server to reply to a client's request.

(ii) It consists of three parts:

(a) Status line

(i) The status line contains the outcome of request

(ii) It includes:

HTTP version : The version of HTTP used

status code : The three-digit code representing the result of the request

Reason phrase : A short description of the status code

(iii) Eg : HTTP/1.1 200 OK

⑥ Headers

(i) Headers in the response provide metadata about the response, such as content type & length

(ii) Common headers: content-type, content-length, server.

⑦ Body

(i) The body contains the actual data requested by the client, such as HTML code, JSON data or a image file.

(Q6) Explain basic internet protocols essential for transferring data and sending mails.

⇒ (1) HTTP

(i) Hypertext transfer protocol is used for transferring web pages over the internet.

(ii) Working:

(a) The client sends an HTTP request

(b) The server responds with an HTTP response.

(2) HTTPS

(i) HTTPS is a secure version of HTTP, using encryption protocols to secure data during transfer.

(ii) It ensures data exchanged between the server and the client is encrypted.

(3) SMTP

(i) Simple mail transfer protocol is used for sending emails from a client to a server and between servers.

(ii) Working:

(a) When an email is sent, SMTP sends it from client's email server to the recipient's mail server.

(4) POP3

(i) Post office Protocol version 3 is used by email clients to retrieve emails from a remote server to a local device.

(ii) Working:

(a) The client connects to the server

(b) Downloads the emails

(c) Disconnects from the server.

(5) IMAP

(i) Internet message access protocol is used for retrieving and managing emails directly on the server, allowing synchronization across multiple devices.

(ii) Working

(a) IMAP allows you to read, organize & delete emails on the server without downloading them on the local device.

(b) Changes made on one device are reflected on all devices.

(6) FTP

(i) File transfer protocol is used for transferring files between computers over the internet or a network.

(ii) Working:

(a) A client sends a request to the FTP server to list, upload, or download files.

(b) FTP supports two modes, active and passive mode, determining how data is transferred.

(7) Telnet

(i) Teletype network is a protocol used to remotely access and manage devices over a network.

(ii) Working:

(a) A Telnet client connects to a remote machine using its IP address and port number.

(b) The user logs into the machine and can run commands.

Q7. List and explain the 3 ways to add a stylesheet (CSS) to an HTML web page with suitable examples.

=>

1. In HTML, you can apply styles to elements using CSS (Cascading Style Sheets) in three different ways:

- i. Inline CSS
- ii. Internal CSS
- iii. External CSS.

2. Inline CSS

- i. Inline CSS is used to apply styles directly within an HTML element using the **style** attribute.
- ii. This method is best for applying a single style to a specific element on a page.
- iii. Syntax : <element style="property: value;">
- iv. Example :

```
<html>
  <head>
    </head>
  <body>
    <h1 style="color: blue; text-align: center;">Welcome to Inline CSS</h1>
    <p style="font-size: 20px; color: green;">This is a paragraph with inline CSS.</p>
  </body>
</html>
```

3. Internal CSS

- i. Internal CSS is written within the <style> tag in the <head> section of the HTML document.
- ii. This method is useful when you want to apply styles to a specific page but don't want to create an external stylesheet.
- iii. Syntax:

```
<head>
  <style>
    selector {
      property: value;
    }
  </style>
</head>
```

- iv. Example:

```
<html>
<head>
    <style>
        body {
            background-color: lightgray;
        }
        h1 {
            color: red;
            text-align: center;
        }
    </style>
</head>
<body>
    <h1>Welcome to Internal CSS</h1>
</body>
</html>
```

4. External CSS

- i. External CSS involves linking a separate CSS file to an HTML document using the `<link>` tag.
- ii. This is the most recommended and scalable way of adding CSS, especially for large websites, as it allows you to reuse the same stylesheet across multiple pages.
- iii. Syntax :

```
<head>
    <link rel="stylesheet" href="styles.css">
</head>
```

- iv. Example :

index.html

```
<html>
<head>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <h1>Welcome to External CSS</h1>
</body>
</html>
```

styles.css

```
body {  
    background-color: lightblue;  
}  
h1 {  
    color: darkblue;  
    text-align: center;  
}
```