LABORATORY MANUAL

Web Development Laboratory

Department of Computer Science and Engineering
Jorhat Engineering College
Jorhat, Assam - 785007

Contents

1

Introduction

HTML

HTML (HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. Other technologies besides HTML are generally used to describe a web page's appearance/presentation (CSS) or functionality/behavior (JavaScript).

"Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web.

HTML uses "markup" to annotate text, images, and other content for display in a Web browser. HTML markup includes special "elements" such as <head>, <title>, <body>, <header>, <footer>, <article>, <section>, <p>, <div>, <span>, <img>, <aside>, <audio>, <canvas>, <datalist>, <details>, <embed>, <nav>, <output>, <progress>, <video>, <ul>, <ol>, <li> and many others.

An HTML element is set off from other text in a document by "tags", which consist of the element name surrounded by "<" and ">". The name of an element inside a tag is case insensitive. That is, it can be written in uppercase, lowercase, or a mixture. For example, the <title> tag can be written as <Title>, <TITLE>, or in any other way.

CSS

CSS stands for Cascading Style Sheets. CSS describes how HTML elements are to be displayed on screen, paper, or in other media. CSS saves a lot of work. It can control the layout of multiple web pages all at once. External stylesheets are stored in CSS files.

HTML was NEVER intended to contain tags for formatting a web page. HTML was created to describe the content of a web page, like:

```
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
```

When tags like <font>, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large websites, where fonts and color information were added to every single page, became a long and expensive process.

To solve this problem, the World Wide Web Consortium (W3C) created CSS. CSS removed the style formatting from the HTML page.

JavaScript

JavaScript), often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it for client-side page behavior, and all major web browsers have a dedicated JavaScript engine to execute it.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM). However, the language itself does not include any input/output (I/O), such as networking, storage, or graphics facilities, as the host environment (usually a web browser) provides those APIs.

Originally used only in web browsers, JavaScript engines are also now embedded in server-side website deployments and non-browser applications.

Although there are similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.

2

HTML5 Skeleton

Objective

To understand HTML tags to create a skeleton page

Introduction

The basic HTML skeleton is the set of tags required of every HTML web page you build. The tags that make up the skeleton tell browsers what kind of file it is reading, and without the skeleton HTML files will not be rendered correctly in web browsers.

The Four Skeleton Tags There are four tags that need to be included in the skeleton. These are the structure tags, so called because they provide browsers with the basic strucure of HTML documents. The four structure tags are:

The <html> tag starts an HTML file and tells browsers what kind of file this is.

The <head> tag includes information about the document such as the name of the file and other technical information like meat tags and style tags, which will be covered later in the tutorial.

The <title> tag is where you place the title of the web page. The title tag goes inside the head, between the opening and closing head tags.

The <body> is where you place all the information that will actually show up on the web page once it is online and opened in a browser. Whatever goes in the body of the HTML file is the content your readers will see when they visit your site.

There can be only one head, title, and body in an HTML document.

Code

```
<!DOCTYPE html>
```

```html
<html>
    <head>
        <!-- Single line comment -->
        <title>Page Title</title>
    </head>
    <body>
        <!--
            Multi-line comment
        -->
        <h1>This is a Heading</h1>
        <p>This is a paragraph.</p>
    </body>
</html>
```

Explanation

- The <!DOCTYPE html> declaration defines this document to be HTML5

- The <html> element is the root element of an HTML page

- The <head> element contains meta information about the document

- The <title> element specifies a title for the document

- The <body> element contains the visible page content

- The <h1> element defines a large heading

- The <p> element defines a paragraph

3

HTML Head

Objective

To understand HTML tags for a HTML head

Introduction

The <head> element is a container for metadata (data about data) and is placed between the <html> tag and the <body> tag.

HTML metadata is data about the HTML document. Metadata is not displayed. Metadata typically define the document title, character set, styles, scripts, and other meta information.

The following tags describe metadata: <title>, <style>, <meta>, <link>, <script>, and <base>.

Code

```
<head>
  <title>Page Title</title>
  <meta charset="UTF-8">
  <meta name="description" content="Free Web tutorials">
  <meta name="keywords" content="HTML, CSS, XML, JavaScript">
  <meta name="author" content="John Doe">
  <style>
    h1 {
        color: red;
    }
  </style>
  <link rel="stylesheet" href="mystyle.css">
  <script>
    function myFunction {
        alert("Hello JEC!");
    }
  </script>
  <script src="myScript.js"></script>
</head>
```

Explanation

- The <title> element
  - defines a title in the browser tab
  - provides a title for the page when it is added to favorites
  - displays a title for the page in search engine results
- The meta charset define the character set used
- The meta description define a description of your web page for search engine, social media
- The meta keywords define keywords for search engines
- The meta author define the author of a page
- The <style> element is used to define style information for a single HTML page
- The <link> element is used to link to external style sheets
- The <script> element is used to define client-side JavaScripts
  - either by writing directly in the head
  - or by referencing to an external JavaScript file

4

HTML body

Objective

To understand HTML tags for a HTML body

Introduction

All HTML documents must start with a document type declaration: <!DOCTYPE html>. The HTML document itself begins with <html> and ends with </html>. The visible part of the HTML document is between <body> and </body>.

Code

```html
<body>
    <h1>This is heading 1</h1>
    <h2>This is heading 2</h2>
    <h3>This is heading 3</h3>
    <p>This is a paragraph.</p>
    <p>This is another paragraph.</p>
    <a href="https://jecassam.ac.in/">This is a link </a>
    <br>
    <img src="https://jecassam.ac.in/jec-drone-view.jpg" alt="Bird's viewJEC" width="452" height="190">
</body>
```

Explanation

- HTML headings are defined with the <h1> to <h6> tags. <h1> defines the most important heading. <h6> defines the least important heading

- HTML paragraphs are defined with the <p> tag
- HTML links are defined with the <a> tag
- The <br> tag inserts a single line break.
- HTML images are defined with the <img> tag. The source file (src), alternative text (alt), width, and height are provided as attributes.

5

HTML formatting

Objective

To understand HTML tags to format text in body

Introduction

HTML defines special elements for defining text with a special meaning. HTML uses elements like <b> and <i> for formatting output, like bold or italic text.

Code

```
<body>
    <p>The<strong>strong</strong> sounding topic of
       the page will be written in <b>bold</b>. The
       <em>emphasised</em> text will be written in
       <i>italics</i>. Texts which are <u>underlined<
       /u> might sometimes be written <del>wrongly</
       del> <ins>incorrectly</ins>.</p>
    <p>H<sub>2</sub>O is water.</p>
    <p>a<sup>2</sup>+b<sup>2</sup>=c<sup>2</sup> is
       the Pythagoras theorem, where c is the longest
        side of the triangle, and a & b are the other
        two sides
    <p>We can insert symbols and smileys in our page
       using HTML.</p>
</body>
```

Explanation

- <strong> - Important text
- <b> - Bold text

- <em> - Emphasized text
- <i> - Italic text
- <del> - Deleted text
- <ins> - Inserted text
- <sub> - Subscript text
- <sup> - Superscript text

6

HTML list

Objective

To understand HTML tags for ordered, unordered, and description list.

Introduction

HTML offers web authors three ways for specifying lists of information. All lists must contain one or more list elements.

Code

```
<body>
    <p>Unordered list</p>
    <p>Type 1</p>
    <ul>
        <li>Item 1</li>
        <li>Item 2</li>
        <li>Item 3</li>
    </ul>
    <p>Type 2</p>
    <ul style="list-style-type:square;">
        <li>Coffee</li>
        <li>Tea</li>
        <li>Milk</li>
    </ul>
    <p>Ordered list</p>
    <p>Type 1</p>
    <ol>
        <li>Item 1</li>
        <ol>
            <li>SubItem 1.1</li>
            <li>SubItem 1.2</li>
            <li>SubItem 1.3</li>
        </ol>
        <li>Item 2</li>
        <li>Item 3</li>
    </ol>
    <dl>
        <dt>Coffee</dt>
            <dd>- black hot drink</dd>
        <dt>Milk</dt>
            <dd>- white cold drink</dd>
    </dl>
</body>
```

Explanation

- An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.
- The CSS list-style-type property is used to define the style of the list item marker
  - disc Sets the list item marker to a bullet (default)
  - circle Sets the list item marker to a circle
  - square Sets the list item marker to a square
  - none The list items will not be marked
- An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.
- HTML also supports description lists. The <dl> tag defines the description list, the <dt> tag defines the term (name), and the <dd> tag describes each term

7

HTML table

Objective

To understand HTML tags for a HTML table

Introduction

An HTML table is defined with the <table> tag. Each table row is defined with the <tr> tag. A table header is defined with the <th> tag. By default, table headings are bold and centered. A table data/cell is defined with the <td> tag.

Code

```
<body>
    <p>Table 1</p>
    <table style="width:100%">
      <tr>
        <th>Firstname</th>
        <th>Lastname</th>
        <th>Age</th>
      </tr>
      <tr>
        <td>Jill</td>
        <td>Smith</td>
        <td>50</td>
      </tr>
      <tr>
        <td>Eve</td>
        <td>Jackson</td>
        <td>94</td>
      </tr>
    </table>
    <p>Table 2</p>
    <table style="width:100%">
```

```
      <tr>
        <th>Name</th>
        <th colspan="2">Telephone</th>
      </tr>
      <tr>
        <td>Bill Gates</td>
        <td>55577854</td>
        <td>55577855</td>
      </tr>
    </table>
    <p>Table 3</p>
    <table style="width:100%">
      <tr>
        <th>Name:</th>
        <td>Bill Gates</td>
      </tr>
      <tr>
        <th rowspan="2">Telephone:</th>
        <td>55577854</td>
      </tr>
      <tr>
        <td>55577855</td>
      </tr>
    </table>
</body>
```

Explanation

- Each table row is defined with the <tr> tag. A table header is defined with the <th> tag. By default, table headings are bold and centered. A table data/cell is defined with the <td> tag.
- The <td> elements are the data containers of the table. They can contain all sorts of HTML elements; text, images, lists, other tables, etc.
- Use the colspan attribute to make a cell span many columns
- Use the rowspan attribute to make a cell span many rows

8

HTML form

Objective

To understand HTML tags for HTML forms

Introduction

An HTML form contains form elements. Form elements are different types of input elements, like: text fields, checkboxes, radio buttons, submit buttons, and more.

Code

```
<body>
    <form action="next_page.php" method="GET">
        <input name="fname" type="text">
        <input name="lname" type="text" value="Smith">
        <input name="email" type="email" placeholder="Your email id" required />
        <input name="password" type="password" placeholder="Password" required />
        <select name="cars">
          <option value="tata">Tata</option>
          <option value="maruti">Maruti</option>
          <option value="fiat">Fiat</option>
          <option value="mahindra" selected>Mahindra</option>
        </select>
        <hr />
        <textarea name="address" rows="4" cols="30">Your address</textarea>
        <hr />
        <input type="radio" name="gender" value="male"> Male <br />
        <input type="radio" name="gender" value="female" checked> Female <br />
        <hr />
        <input type="checkbox" name="vehicle_bike" value="Bike"> I want a bike <br />
        <input type="checkbox" name="vehicle_car" value="Car"> I want a car <br />
        Rating: <input type="number" name="quantity" min="1" max="5"> <br />
        Signature: <input type="file" name="myFile">
        <input type="reset"><br />
        <input type="submit" value="Submit">
</body>
```

Explanation

- The HTML <form> element defines a form that is used to collect user input
- The form-handler is typically a server page with a script for processing input data. The form-handler is specified in the form's action attribute
- The method attribute specifies the HTTP method (GET or POST) to be used when submitting the form data
- <input type="text"> defines a one-line text input field
- The value attribute specifies the initial value for an input field
- The placeholder attribute specifies a hint that describes the expected value of an input field (a sample value or a short description of the format). The hint is displayed in the input field before the user enters a value. The placeholder attribute works with the following input types: text, search, url, tel, email, and password.
- The required attribute specifies that an input field must be filled out before submitting the form. The required attribute works with the following input types: text, search, url, tel, email, password, date pickers, number, checkbox, radio, and file.
- <input type="password"> defines a password field. The characters in a password field are masked (shown as asterisks or circles).
- <input type="radio"> defines a radio button. Radio buttons let a user select ONE of a limited number of choices
- The <select> element defines a drop-down list

- The <option> elements defines an option that can be selected. By default, the first item in the drop-down list is selected. To define a pre-selected option, add the selected attribute to the option
- The <textarea> element defines a multi-line input field (a text area)
- <input type="checkbox"> defines a checkbox. Checkboxes let a user select ZERO or MORE options of a limited number of choices
- The <input type="file"> defines a file-select field and a "Browse" button for file uploads
- <input type="reset"> defines a reset button that will reset all form values to their default values
- <input type="submit"> defines a button for submitting form data to a form-handler.

9

CSS Introduction

Objective

To understand CSS attributes for inline CSS

Introduction

CSS is a language that describes the style of an HTML document. CSS describes how HTML elements should be displayed.

Code

```html
<body style="background-color:powderblue;">
    <h1 style="color:blue;">This is a heading</h1>
    <p style="color:red;">>This is a paragraph.</p>
    <p style="font-family:impact;">This is a paragraph.</p>
    <p style="font-family:courier;">This is a paragraph.</p>
    <p style="font-size: 48px;">This is a paragraph.</p>
    <p style="text-align:center;">Centered paragraph.</p>
</body>
```

Explanation

- The CSS background-color property defines the background color for an HTML element.
- The CSS color property defines the text color for an HTML element
- The CSS font-family property defines the font to be used for an HTML element

- The CSS font-size property defines the text size for an HTML element
- The CSS text-align property defines the horizontal text alignment for an HTML element

10

CSS Selectors

Objective

To understand CSS selectors elements

Introduction

CSS selectors are used to "find" (or select) the HTML elements you want to style. We can divide CSS selectors into five categories: Simple selectors (select elements based on name, id, class), Combinator selectors (select elements based on a specific relationship between them), Pseudo-class selectors (select elements based on a certain state), Pseudo-elements selectors (select and style a part of an element), Attribute selectors (select elements based on an attribute or attribute value).

Code

```html
<head>
    <style>
        body {
            background-image: url("image.jpg");
        }
        h1 {
            color: red;
            text-align: center;
        }
        p {
            color: blue;
        }
        #myID {
            border-style: solid;
            border-width: 1px;
            border-color: tomato;
            color: #ff6347;
        }
        .class_1 {
            color: rgb(255, 99, 71);
        }
        .class_2 {
            color: rgba(255, 99, 71, 0.5);
        }
    </style>
</head>
<body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
    <p id="myID">This is a paragraph.</p>
    <p class="class_1">This is a paragraph.</p>
    <p class="class_1">This is a paragraph.</p>
    <p class="class_2">This is a paragraph.</p>
    <p class="class_2">This is a paragraph.</p>
</body>
```

Explanation

- The body is given an background image
- The h1 element will be colored red,and center-aligned.
- All p elements will be colored blue, unless overwritten.
- The id selector uses the id attribute of an HTML element to select a specific element. The id of an element is unique within a page, so the id selector is used to select one unique element. To select an element with a specific id, write a hash (#) character, followed by the id of the element.
- The class selector selects HTML elements with a specific class attribute. To select elements with a specific class, write a period (.) character, followed by the class name.

11

CSS Padding and Margin

Objective

To understand CSS tags for padding and margin

Introduction

The CSS margin properties are used to create space around elements, outside of any defined borders. The CSS padding properties are used to generate space around an element's content, inside of any defined borders. With CSS, you have full control over the margins and padding. There are properties for setting the margin and padding for each side of an element (top, right, bottom, and left).

Code

```
<head>
    <style>
        .class_1 {
            background: red;
            padding-top: 100px;
            padding-left: 50px;
        }
        .class_2 {
            background: blue;
            margin-top: 100px;
            margin-left: 50px;
        }
    </style>
</head>
<body>
    <p class="class_1">This is a paragraph.</p>
    <p class="class_2">This is a paragraph.</p>
</body>
```

Explanation

- The CSS padding properties are used to generate space around an element's content, inside of any defined borders.

- The CSS margin properties are used to create space around elements, outside of any defined borders.

12

CSS Positions

Objective

To understand CSS tags to position elements in a page

Introduction

The position property specifies the type of positioning method used for an element (static, relative, fixed, absolute or sticky).

Code

```html
<head>
    <style>
        .fixed_header {
            position: fixed;
            height: 100px;
        }
        .class_2 {
            position: absolute;
            top: 500px;
            left:0px;
            width: 100px;
        }
    </style>
</head>
<body>
    <header class="fixed_header">This is a fixed header.</header>
    <aside class="share_button">Share on Facebook</aside>
</body>
```

Explanation

- An element with position: fixed; is positioned relative to the view-port, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element. A fixed element does not leave a gap in the page where it would normally have been located.

- An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the view-port, like fixed). However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.

13

CSS animations

Objective

To understand CSS tags for animation

Introduction

CSS allows animation of HTML elements without using JavaScript or Flash! An animation lets an element gradually change from one style to another. You can change as many CSS properties you want, as many times you want.

Code

```
<head>
    <style>
        div {
            width: 100px;
            height: 100px;
            background-color: red;
            position: relative;
            animation-name: example;
            animation-duration: 4s;
            animation-iteration-count: 2;
            animation-direction: alternate;
        }
        @keyframes example {
            0%   {background-color:red; left:0px; top:0px;}
            25%  {background-color:yellow; left:200px; top:0px;}
            50%  {background-color:blue; left:200px; top:200px;}
            75%  {background-color:green; left:0px; top:200px;}
            100% {background-color:red; left:0px; top:0px;}
        }
    </style>
</head>
<body>
    <div> </div>
</body>
```

Explanation

- When you specify CSS styles inside the @keyframes rule, the animation will gradually change from the current style to the new style at certain times. To get an animation to work, you must bind the animation to an element.

- Specifies the name of the keyframe you want to bind to the selector.

- The animation-duration property defines how long time an animation should take to complete. If the animation-duration property is not specified, no animation will occur, because the default value is 0s (0 seconds). In the example above we have specified when the style will change by using the keywords "from" and "to" (which represents 0% (start) and 100% (complete)).

- The animation-iteration-count property specifies the number of times an animation should run.

- The animation-direction property specifies whether an animation should be played forwards, backwards or in alternate cycles.

14

JavaScript Introduction and Output

Objective

To understand JavaScript function to display output

Introduction

JavaScript is the programming language of HTML and the Web. JavaScript can "display" data in different ways: Writing into an HTML element, using innerHTML; Writing into the HTML output using document.write(); Writing into an alert box, using window.alert(); Writing into the browser console, using console.log().

Code

```
<head>
  <script>
    function myFunction {
        alert("Hello JEC!");
        document.getElementById("myID").innerHTML = "This text goes in myID";
        console.log("Finished displaying alert and innerHTML");
    }
  </script>
</head>
<body>
  <p id="myID"></p>
  <button onclick="myFunction()">Click Here</button>
</body>
```

Explanation

- You can use an alert box to display data
- To access an HTML element, JavaScript can use the document.getElementById(id) method. The id attribute defines the HTML element. The innerHTML property defines the HTML content.
- For debugging purposes, you can use the console.log() method to display data.

15

JavaScript Array

Objective

To understand arrays and related methods in JavaScript

Introduction

JavaScript arrays are used to store multiple values in a single variable. An array can hold many values under a single name, and you can access the values by referring to an index number.

Code

```
<head>
  <script>
    var fruits = ["Mango", "Apple", "Grapes"];
    console.log("The first element is "+fruits[0]);
    fruits[0] = "Banana";
    console.log("The first element is changed to "+fruits[0]);
    var popped = fruits.pop();
    console.log("The last element i.e. "+popped+" was popped from the array.");
    fruits.push("Kiwi");
    console.log("The length of the array is "+fruits.length);
    fruits.sort();
    console.log("Sorted: "+fruits);
    fruits.reverse();
    console.log("Reversed: "+fruits);
    var points = [40, 100, 1, 5, 25, 10];
    points.sort(function(a, b){return a - b});
  </script>
</head>
```

Explanation

- Using an array literal is the easiest way to create a JavaScript Array. Spaces and line breaks are not important. A declaration can span multiple lines.
- You access an array element by referring to the index number. Array indexes start with 0. [0] is the first element. [1] is the second element.
- The value of an index can be changed after declaration.
- The pop() method removes the last element from an array. The pop() method returns the value that was "popped out".
- The push() method adds a new element to an array (at the end).
- The length property of an array returns the length of an array (the number of array elements).
- The sort() method sorts an array alphabetically. However, if numbers are sorted as strings, "25" is bigger than "100", because "2" is bigger than "1".
- The reverse() method reverses the elements in an array. You can use it to sort an array in descending order
- The sort() method will produce incorrect result when sorting numbers. You can fix this by providing a compare function.

16

JavaScript AJAX

Objective

To understand the AJAX technology using JavaScript

Introduction

AJAX (Asynchronous JavaScript And XML) is not a programming language. AJAX just uses a combination of: (1) A browser built-in XMLHttpRequest object (to request data from a web server), and (2) JavaScript and HTML DOM (to display or use the data).

AJAX allows web pages to be updated asynchronously by exchanging data with a web server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

Code

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function loadDoc() {
        var xhttp = new XMLHttpRequest();
        xhttp.onreadystatechange = function() {
          if (this.readyState == 4 && this.status == 200) {
            document.getElementById("demo").innerHTML = this.responseText;
          }
        };
        xhttp.open("GET", "ajax_info.txt", true);
        xhttp.send();
      }
    </script>
  </head>
  <body>
    <div id="demo">
      <h2>Let AJAX change this text</h2>
      <button type="button" onclick="loadDoc()">Change Content</button>
    </div>
  </body>
</html>
```

Explanation

-
-
  - The HTML page contains a <div> section and a <button>.
  - The <div> section is used to display information from a server.
  - The <button> calls a function (if it is clicked).
  - The function requests data from a web server and displays it
  - The XMLHttpRequest object can be used to exchange data with a web server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.
  - For security reasons, modern browsers do not allow access across domains. This means that both the web page and the XML file it tries to load, must be located on the same server.
  - The XMLHttpRequest object is used to exchange data with a server. To send a request to a server, we use the open() and send() methods of the XMLHttpRequest object.
  - The readyState property holds the status of the XMLHttpRequest. The onreadystatechange property defines a function to be executed when the readyState changes. The status property and the statusText property holds the status of the XMLHttpRequest object.