**Week 1**

1. Lists, Links and Images

a. Write a HTML program, to explain the working of lists. Note: It should have an ordered list, unordered list, nested lists and ordered list in an unordered list and definition lists.

b. Write a HTML program, to explain the working of hyperlinks using tag and href, target Attributes.

c. Create a HTML document that has your image and your friend’s image with a specific height and width. Also when clicked on the images it should navigate to their respective profiles.

d. Write a HTML program, in such a way that, rather than placing large images on a page, the preferred technique is to use thumbnails by setting the height and width parameters to something like to 100\*100 pixels. Each thumbnail image is also a link to a full sized version of the image. Create an image gallery using this technique

**1. Lists, Links, and Images**

HTML provides different types of lists to structure and organize content. The primary types of lists are:

1. **Ordered List (<ol>)**:
   * Displays items in a numbered sequence.
   * Each item is wrapped in an <li> (list item) tag.
2. **Unordered List (<ul>)**:
   * Displays items with bullets or other symbols.
   * Each item is also wrapped in an <li> tag.
3. **Nested Lists**:
   * A list (ordered or unordered) inside another list.
4. **Mixed Lists**:
   * Combining ordered and unordered lists.
5. **Definition List (<dl>)**:
   * Consists of terms and their definitions.
   * <dt>: Represents the term.
   * <dd>: Represents the definition.

**Program:**

<!DOCTYPE html>

<html>

<head>

<title>Lists Example</title>

</head>

<body>

<h1>Lists in HTML</h1>

<h2>Ordered List</h2>

<ol>

<li>Item 1</li>

<li>Item 2</li>

<li>Item 3</li>

</ol>

<h2>Unordered List</h2>

<ul>

<li>Item A</li>

<li>Item B</li>

<li>Item C</li>

</ul>

<h2>Nested Lists</h2>

<ul>

<li>Parent Item

<ol>

<li>Child Item 1</li>

<li>Child Item 2</li>

</ol>

</li>

<li>Parent Item 2</li>

</ul>

<h2>Definition List</h2>

<dl>

<dt>HTML</dt>

<dd>HyperText Markup Language</dd>

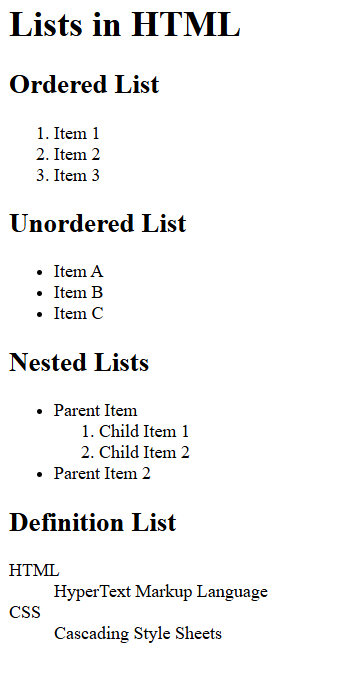
<dt>CSS</dt>

<dd>Cascading Style Sheets</dd>

</dl>

</body>

</html>



**b. HTML Program for Hyperlinks  
Theory of Hyperlinks in HTML**

Hyperlinks are fundamental elements in HTML used to navigate between different web pages or sections within the same page. They are created using the <a> (anchor) tag. The following key attributes are commonly used with hyperlinks:

1. href Attribute: Specifies the URL of the page the link goes to. It can be an absolute URL (e.g., https://www.example.com) or a relative URL (e.g., about.html).
2. target Attribute: Determines where to open the linked document. Common values for the target attribute include:
   * \_self (default): Opens the link in the same tab/window.
   * \_blank: Opens the link in a new tab/window.
   * \_parent: Opens the link in the parent frame (if the page is inside a frame).
   * \_top: Opens the link in the full body of the window (if the page is inside a frame).

**Program:**

<!DOCTYPE html>

<html>

<head>

<title>Hyperlinks Example</title>

</head>

<body>

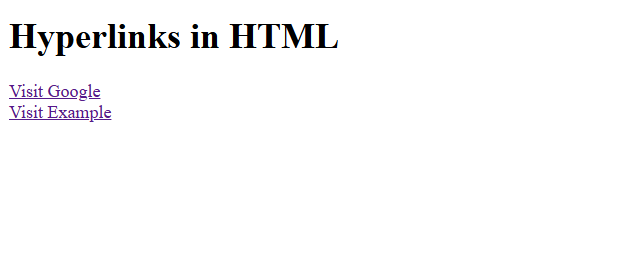
<h1>Hyperlinks in HTML</h1>

<a href="https://www.google.com" target="\_blank">Visit Google</a><br>

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

</body>

</html>



**c. Image Navigation Program  
Theory of Images as Hyperlinks in HTML**

In HTML, images can act as hyperlinks, allowing users to navigate to different pages or websites when the image is clicked. This is achieved by embedding an <img> tag inside an <a> (anchor) tag.

Key attributes used:

1. **<img> Tag**:
   * **src**: Specifies the source of the image.
   * **alt**: Provides alternative text for the image.
   * **height** and **width**: Specify the dimensions of the image.
2. **<a> Tag**:
   * **href**: Specifies the URL to navigate to when the image is clicked.

**Program**

<!DOCTYPE html>

<html>

<head>

<title>Image Navigation</title>

</head>

<body>

<h1>Images with Navigation</h1>

<a href="https://www.facebook.com/profile.php?id=your\_friend\_id">

<img src="friend.jpg" alt="Friend's Image" width="200" height="200">

</a>

<a href="https://www.facebook.com/profile.php?id=your\_profile\_id">

<img src="my.jpg" alt="My Image" width="200" height="200">

</a>

</body>

</html>

**d. Thumbnail Image Gallery**Thumbnails are smaller versions of images, often used to save space and provide a preview. When clicked, these thumbnails typically link to the full-sized image. In HTML, this is accomplished using the <a> (anchor) and <img> (image) tags together. The following steps describe how this works:

1. **Thumbnail Display**:
   * Use the <img> tag to display a smaller version of the image.
   * Set the width and height attributes of the <img> tag to control the size of the thumbnail.
2. **Link to Full-Sized Image**:
   * Wrap the <img> tag inside an <a> tag.
   * The href attribute of the <a> tag should point to the full-sized image.

This technique improves performance by initially loading smaller images while allowing users to view the full-sized version on demand.

**Program:**

<!DOCTYPE html>

<html>

<head>

<title>Image Gallery</title>

</head>

<body>

<h1>Thumbnail Gallery</h1>

<a href="image1.jpg"><img src="thumb1.jpg" alt="Image 1" width="100" height="100"></a>

<a href="image2.jpg"><img src="thumb2.jpg" alt="Image 2" width="100" height="100"></a>

<a href="image3.jpg"><img src="thumb3.jpg" alt="Image 3" width="100" height="100"></a>

</body>

</html>

**WEEK 2**

a. Write a HTML program, to explain the working of tables. (use tags: <table>, <tr>, <th>,

<td> and attributes: border, rowspan, colspan)

b. Write a HTML program, to explain the working of tables by preparing a

timetable. (Note: Use <caption> tag to set the caption to the table & also use cell spacing,

cell padding, border, rowspan, colspan etc.).

c. Write a HTML program, to explain the working of forms by designing Registration form.

(Note: Include text field, password field, number field, date of birth field, checkboxes,

radio buttons, list boxes using <select>&<option> tags, <text area> and two buttons ie:

submit and reset. Use tables to provide a better view).

d. Write a HTML program, to explain the working of frames, such that page is to be divided

into 3 parts on either direction. (Note: first frame ◊ image, second frame ◊ paragraph,

third frame ◊ hyperlink. And also make sure of using “no frame” attribute such that

frames to be fixed).

**2. HTML Tables, Forms, and Frames**

**a. Table Example**

**Theory:**

1. **<table>**: Used to define an HTML table. It organizes data into rows and columns.
2. **<tr>**: Represents a table row. Contains one or more <th> (table header) or <td> (table data) elements.
3. **<th>**: Defines a header cell within a row, typically displayed as bold and centered.
4. **<td>**: Represents a standard data cell within a row.

**Attributes:**

1. **border**: Specifies the width of the border around the table and its cells.
2. **rowspan**: Merges cells vertically by specifying the number of rows a cell should span.
3. **colspan**: Merges cells horizontally by specifying the number of columns a cell should span.

**Program:**

<!DOCTYPE html>

<html>

<head>

<title>HTML Tables</title>

</head>

<body>

<table border="1">

<tr>

<th>Name</th>

<th>Age</th>

<th>City</th>

</tr>

<tr>

<td>John</td>

<td>25</td>

<td>New York</td>

</tr>

<tr>

<td>Jane</td>

<td>30</td>

<td>Los Angeles</td>

</tr>

</table>

</body>

</html>

**b. Timetable Example**

**Theory:**

 **Tags:**

* **<table>**: Creates a table for displaying data in rows and columns.
* **<caption>**: Provides a descriptive title for the table, usually displayed above it.
* **<th>**: Defines header cells in a table row, typically bold and centered by default.
* **<td>**: Defines standard data cells in a table row.
* **<tr>**: Groups a row of table cells (either <th> or <td>).

 **Attributes:**

* **border**: Specifies the thickness of the table's borders.
* **cellspacing**: Sets the space between individual table cells.
* **cellpadding**: Defines the padding inside each table cell to control spacing between the content and cell edges.
* **rowspan**: Merges a cell vertically across multiple rows.
* **colspan**: Merges a cell horizontally across multiple columns.

**Program:**

<!DOCTYPE html>

<html>

<head>

<title>Timetable</title>

</head>

<body>

<table border="1" cellpadding="5" cellspacing="2">

<caption>Weekly Timetable</caption>

<tr>

<th>Day</th>

<th>9-10 AM</th>

<th>10-11 AM</th>

<th>11-12 PM</th>

</tr>

<tr>

<td>Monday</td>

<td>Math</td>

<td>Physics</td>

<td>Chemistry</td>

</tr>

<tr>

<td>Tuesday</td>

<td>Biology</td>

<td>Math</td>

<td>English</td>

</tr>

</table>

</body>

</html>

**c. Registration Form**

<!DOCTYPE html>

<html>

<head>

<title>Registration Form</title>

</head>

<body>

<form>

<table>

<tr>

<td>Name:</td>

<td><input type="text" name="name"></td>

</tr>

<tr>

<td>Password:</td>

<td><input type="password" name="password"></td>

</tr>

<tr>

<td>Date of Birth:</td>

<td><input type="date" name="dob"></td>

</tr>

<tr>

<td>Gender:</td>

<td>

<input type="radio" name="gender" value="male">Male

<input type="radio" name="gender" value="female">Female

</td>

</tr>

<tr>

<td>Languages Known:</td>

<td>

<input type="checkbox" name="lang" value="english">English

<input type="checkbox" name="lang" value="hindi">Hindi

</td>

</tr>

<tr>

<td>Country:</td>

<td>

<select name="country">

<option>India</option>

<option>USA</option>

<option>UK</option>

</select>

</td>

</tr>

<tr>

<td>Comments:</td>

<td><textarea name="comments"></textarea></td>

</tr>

<tr>

<td colspan="2">

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

<input type="reset" value="Reset">

</td>

</tr>

</table>

</form>

</body>

</html>

**d. Frames Example**

<!DOCTYPE html>

<html>

<head>

<title>HTML Frames</title>

</head>

<frameset rows="50%,25%,25%">

<frame src="image.html" name="image\_frame">

<frame src="paragraph.html" name="text\_frame">

<frame src="links.html" name="link\_frame">

<noframes>

Your browser does not support frames.

</noframes>

</frameset>

</html>

**WEEK 3**

a. Write a HTML program, that makes use of <article>, <aside>, <figure>, <figcaption>,

<footer>, <header>, <main>, <nav>, <section>, <div>, <span> tags.

b. Write a HTML program, to embed audio and video into HTML web page.

c. Write a program to apply different types (or levels of styles or style specification formats) - inline, internal, external styles to HTML elements. (identify selector, property and value).

**3. HTML5 and Cascading Style Sheets (CSS)**

**a. HTML Program Using Semantic Tags**

<!DOCTYPE html>

<html>

<head>

<title>HTML5 Semantic Tags</title>

</head>

<body>

<header>

<h1>My Website</h1>

</header>

<nav>

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

</nav>

<main>

<section>

<article>

<h2>Article Title</h2>

<p>This is the content of the article.</p>

<figure>

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

<figcaption>Image Description</figcaption>

</figure>

</article>

<aside>

<h3>Related Links</h3>

<ul>

<li><a href="#">Link 1</a></li>

<li><a href="#">Link 2</a></li>

</ul>

</aside>

</section>

</main>

<footer>

<p>&copy; 2024 My Website</p>

</footer>

</body>

</html>

**b. Embedding Audio and Video**

<!DOCTYPE html>

<html>

<head>

<title>Audio and Video Embedding</title>

</head>

<body>

<h1>Embedding Audio and Video</h1>

<audio controls>

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

Your browser does not support the audio element.

</audio>

<br>

<video controls width="400">

<source src="video.mp4" type="video/mp4">

Your browser does not support the video tag.

</video>

</body>

</html>

**c. Inline, Internal, and External CSS**

<!DOCTYPE html>

<html>

<head>

<title>CSS Styles</title>

<style>

body { background-color: lightblue; }

.internal { color: red; font-size: 20px; }

</style>

<link rel="stylesheet" type="text/css" href="styles.css">

</head>

<body>

<p style="color: green;">This is inline CSS</p>

<p class="internal">This is internal CSS</p>

<p class="external">This is external CSS</p>

</body>

</html>

*External CSS file (styles.css):*

.external {

font-weight: bold;

text-decoration: underline;

}

**WEEK 4**

a. Write a program to apply different types of selector forms

i. Simple selector (element, id, class, group, universal)

ii. Combinator selector (descendant, child, adjacent sibling, general sibling)

iii. Pseudo-class selector

iv. Pseudo-element selector

v. Attribute selector

**4. Selector Forms**

**a. Selector Examples**

<!DOCTYPE html>

<html>

<head>

<title>CSS Selectors</title>

<style>

/\* Simple Selectors \*/

p { color: blue; }

#unique { color: red; }

.class-name { font-size: 18px; }

/\* Group Selector \*/

h1, h2, h3 { font-family: Arial; }

/\* Combinator Selectors \*/

div > p { background-color: yellow; } /\* Child \*/

div p { color: green; } /\* Descendant \*/

p + p { font-weight: bold; } /\* Adjacent Sibling \*/

p ~ p { font-style: italic; } /\* General Sibling \*/

/\* Pseudo-classes \*/

a:hover { color: orange; }

/\* Pseudo-elements \*/

p::first-line { text-transform: uppercase; }

/\* Attribute Selector \*/

input[type="text"] { border: 1px solid black; }

</style>

</head>

<body>

<h1>Selectors Example</h1>

<div>

<p>This is a paragraph inside div.</p>

<p>This is another paragraph inside div.</p>

</div>

<p id="unique">This is a paragraph with ID.</p>

<p class="class-name">This is a paragraph with a class.</p>

<a href="#">Hover over this link</a>

<p>This paragraph will have italic siblings.</p>

<input type="text" placeholder="Type here">

</body>

</html>

**WEEK 5**

a. Write a program to demonstrate the various ways you can reference a color in CSS.

b. Write a CSS rule that places a background image halfway down the page, tilting it

horizontally. The image should remain in place when the user scrolls up or down.

c. Write a program using the following terms related to CSS font and text:

i. font-size

ii. font-weight

iv. text-decoration v. text-transformation

iii. font-style

vi. text-alignment

d. Write a program, to explain the importance of CSS Box model using

i. Content

ii. Border

iii. Margin

iv. padding

**5. CSS with Colors, Fonts, Text, and Box Model**

**a. Referencing Colors in CSS**

<!DOCTYPE html>

<html>

<head>

<title>Colors in CSS</title>

<style>

body { color: rgb(255, 0, 0); }

h1 { color: #00FF00; }

p { color: hsl(240, 100%, 50%); }

</style>

</head>

<body>

<h1>RGB Color</h1>

<p>Hexadecimal and HSL Color</p>

</body>

</html>

**b. Background Image**

<!DOCTYPE html>

<html>

<head>

<title>Background Image</title>

<style>

body {

background-image: url('background.jpg');

background-attachment: fixed;

background-repeat: no-repeat;

background-position: center bottom;

}

</style>

</head>

<body>

<h1>Background Example</h1>

</body>

</html>

**c. Font and Text**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>CSS Font and Text Properties</title>

<style>

body {

font-family: Arial, sans-serif;

text-align: center; /\* text-alignment \*/

}

.styled-text {

font-size: 24px; /\* font-size \*/

font-weight: bold; /\* font-weight \*/

font-style: italic; /\* font-style \*/

text-decoration: underline; /\* text-decoration \*/

text-transform: uppercase; /\* text-transformation \*/

margin: 20px;

}

</style>

</head>

<body>

<h1>CSS Font and Text Properties</h1>

<p class="styled-text">This is styled text</p>

</body>

</html>

**d. Box Model**

<!DOCTYPE html>

<html>

<head>

<title>Box Model</title>

<style>

.box {

width: 200px;

height: 100px;

padding: 20px;

border: 10px solid black;

margin: 15px;

}

</style>

</head>

<body>

<div class="box">Box Model Example</div>

</body>

</html>

**WEEK 6**

a. Write a program to embed internal and external JavaScript in a web page.

b. Write a program to explain the different ways for displaying output.

c. Write a program to explain the different ways for taking input.

d. Create a webpage which uses prompt dialogue box to ask a voter for his name and age. Display the information in table format along with either the voter can vote or not

**6. Applying JavaScript - Internal and External, I/O, Type Conversion**

**a. Embed Internal and External JavaScript**

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Example</title>

<script>

function internalFunction() {

alert("This is internal JavaScript!");

}

</script>

<script src="external.js"></script>

</head>

<body>

<button onclick="internalFunction()">Internal Script</button>

<button onclick="externalFunction()">External Script</button>

</body>

</html>

*External JavaScript file (external.js):*

function externalFunction() {

alert("This is external JavaScript!");

}

**b. Displaying Output**

<!DOCTYPE html>

<html>

<head>

<title>Display Output</title>

</head>

<body>

<script>

alert("This is an alert box!");

document.write("This is written on the webpage.<br>");

console.log("This is a console message.");

</script>

</body>

</html>

**c. Taking Input**

<!DOCTYPE html>

<html>

<head>

<title>Input Methods</title>

</head>

<body>

<script>

let name = prompt("Enter your name:");

alert("Hello, " + name + "!");

let age = confirm("Are you above 18?");

document.write("User is above 18: " + age);

</script>

</body>

</html>

**d. Voting Eligibility**

<!DOCTYPE html>

<html>

<head>

<title>Voting Eligibility</title>

</head>

<body>

<script>

let name = prompt("Enter your name:");

let age = prompt("Enter your age:");

document.write("<table border='1'><tr><th>Name</th><th>Age</th><th>Eligibility</th></tr>");

document.write("<tr><td>" + name + "</td><td>" + age + "</td><td>" + (age >= 18 ? "Can Vote" : "Cannot Vote") + "</td></tr></table>");

</script>

</body>

</html>

**WEEK 7**

a. Write a program using document object properties and methods.

b. Write a program using window object properties and methods.

c. Write a program using array object properties and methods.

d. Write a program using math object properties and methods.

e. Write a program using string object properties and methods.

f. Write a program using regex object properties and methods.

g. Write a program using date object properties and methods.

h. Write a program to explain user-defined object by using properties, methods, accessors, constructors and display.

**7. JavaScript Pre-defined and User-defined Objects**

**a. Document Object**

<!DOCTYPE html>

<html>

<head>

<title>Document Object</title>

</head>

<body>

<script>

document.title = "Updated Title";

document.write("The current URL is: " + document.URL);

</script>

</body>

</html>

**b. Window Object**

<!DOCTYPE html>

<html>

<head>

<title>Window Object</title>

</head>

<body>

<script>

alert("Window Width: " + window.innerWidth + ", Height: " + window.innerHeight);

window.open("https://www.google.com", "\_blank", "width=500,height=500");

</script>

</body>

</html>

**c. Array Object**

<!DOCTYPE html>

<html>

<head>

<title>Array Object</title>

</head>

<body>

<script>

let arr = [1, 2, 3, 4];

arr.push(5);

document.write("Array: " + arr + "<br>");

document.write("Length: " + arr.length + "<br>");

</script>

</body>

</html>

**d. Math Object**

<!DOCTYPE html>

<html>

<head>

<title>Math Object</title>

</head>

<body>

<script>

document.write("Random Number: " + Math.random() + "<br>");

document.write("Square Root of 16: " + Math.sqrt(16));

</script>

</body>

</html>

**e. String Object**

<!DOCTYPE html>

<html>

<head>

<title>String Object</title>

</head>

<body>

<script>

let str = "Hello, JavaScript!";

document.write("String: " + str + "<br>");

document.write("Length: " + str.length + "<br>");

document.write("Uppercase: " + str.toUpperCase());

</script>

</body>

</html>

**f. Regex Object**

<!DOCTYPE html>

<html>

<head>

<title>Regex Object</title>

</head>

<body>

<script>

let text = "The rain in Spain";

let regex = /ain/g;

document.write("Matches: " + text.match(regex));

</script>

</body>

</html>

**g. Date Object**

<!DOCTYPE html>

<html>

<head>

<title>Date Object</title>

</head>

<body>

<script>

let date = new Date();

document.write("Current Date: " + date.toDateString() + "<br>");

document.write("Time: " + date.toTimeString());

</script>

</body>

</html>

**h. User-defined Object**

<!DOCTYPE html>

<html>

<head>

<title>User-defined Object</title>

</head>

<body>

<script>

function Person(name, age) {

this.name = name;

this.age = age;

this.greet = function() {

return "Hello, " + this.name;

};

}

let person = new Person("John", 25);

document.write("Name: " + person.name + "<br>");

document.write("Age: " + person.age + "<br>");

document.write(person.greet());

</script>

</body>

</html>

**WEEK 8**

a. Write a program which asks the user to enter three integers, obtains the numbers from the user and outputs HTML text that displays the larger number followed by the words “LARGER NUMBER” in an information message dialog. If the numbers are equal, output HTML text as “EQUAL NUMBERS”.

b. Write a program to display week days using switch case.

c. Write a program to print 1 to 10 numbers using for, while and do-while loops.

d. Write aprogram to print data in object using for-in, for-each and for-of loops

e. Develop a program to determine whether a given number is an ‘ARMSTRONG NUMBER’ or not. [Eg: 153 is an Armstrong number, since sum of the cube of the digits is equal to the number i.e.,13 + 53+ 33 = 153]

f. Write a program to display the denomination of the amount deposited in the bank in terms of 100’s, 50’s, 20’s, 10’s, 5’s, 2’s & 1’s. (Eg: If deposited amount is Rs.163, the output

should be 1-100’s, 1-50’s, 1- 10’s, 1-2’s & 1-1’s)

**8. JavaScript Conditional Statements and Loops**

**a. Larger or Equal Numbers**

<!DOCTYPE html>

<html>

<head>

<title>Larger Number</title>

</head>

<body>

<script>

let a = parseInt(prompt("Enter first number:"));

let b = parseInt(prompt("Enter second number:"));

let c = parseInt(prompt("Enter third number:"));

if (a === b && b === c) {

document.write("EQUAL NUMBERS");

} else {

let largest = Math.max(a, b, c);

document.write(largest + " LARGER NUMBER");

}

</script>

</body>

</html>

**b. Display Weekdays**

<!DOCTYPE html>

<html>

<head>

<title>Weekdays</title>

</head>

<body>

<script>

let day = parseInt(prompt("Enter a number (1-7):"));

switch (day) {

case 1: document.write("Monday"); break;

case 2: document.write("Tuesday"); break;

case 3: document.write("Wednesday"); break;

case 4: document.write("Thursday"); break;

case 5: document.write("Friday"); break;

case 6: document.write("Saturday"); break;

case 7: document.write("Sunday"); break;

default: document.write("Invalid Day");

}

</script>

</body>

</html>

**c. Print Numbers (1 to 10) Using Loops**

<!DOCTYPE html>

<html>

<head>

<title>Loops</title>

</head>

<body>

<script>

document.write("<strong>Using for loop:</strong><br>");

for (let i = 1; i <= 10; i++) {

document.write(i + " ");

}

document.write("<br><strong>Using while loop:</strong><br>");

let j = 1;

while (j <= 10) {

document.write(j + " ");

j++;

}

document.write("<br><strong>Using do-while loop:</strong><br>");

let k = 1;

do {

document.write(k + " ");

k++;

} while (k <= 10);

</script>

</body>

</html>

**d. Print Data in an Object Using Loops**

<!DOCTYPE html>

<html>

<head>

<title>Loops in Objects</title>

</head>

<body>

<script>

let student = { name: "John", age: 20, grade: "A" };

document.write("<strong>Using for-in:</strong><br>");

for (let key in student) {

document.write(key + ": " + student[key] + "<br>");

}

let numbers = [1, 2, 3, 4];

document.write("<br><strong>Using for-of:</strong><br>");

for (let value of numbers) {

document.write(value + " ");

}

document.write("<br><br><strong>Using forEach:</strong><br>");

numbers.forEach(num => document.write(num + " "));

</script>

</body>

</html>

**e. Armstrong Number**

<!DOCTYPE html>

<html>

<head>

<title>Armstrong Number</title>

</head>

<body>

<script>

let num = parseInt(prompt("Enter a number:"));

let sum = 0, temp = num;

while (temp > 0) {

let digit = temp % 10;

sum += digit \*\* 3;

temp = Math.floor(temp / 10);

}

document.write(num === sum ? `${num} is an Armstrong number.` : `${num} is not an Armstrong number.`);

</script>

</body>

</html>

**f. Denominations of Amount**

<!DOCTYPE html>

<html>

<head>

<title>Denominations</title>

</head>

<body>

<script>

let amount = parseInt(prompt("Enter the amount:"));

let denominations = [100, 50, 20, 10, 5, 2, 1];

document.write("<strong>Denominations:</strong><br>");

denominations.forEach(den => {

let count = Math.floor(amount / den);

if (count > 0) {

document.write(`${count} x ${den}<br>`);

amount %= den;

}

});

</script>

</body>

</html>

**WEEK 9**

a. Design a appropriate function should be called to display

i. Factorial of that number

ii. Fibonacci series up to that number

iii. Prime numbers up to that number

iv. Is it palindrome or not

b. Design a HTML having a text box and four buttons named Factorial, Fibonacci, Prime,

and Palindrome. When a button is pressed an appropriate function should be called to

display

i. Factorial of that number

ii. Fibonacci series up to that number

iii. Prime numbers up to that number

iv. Is it palindrome or not

c. Write a program to validate the following fields in a registration page

**9. JavaScript Functions and Events**

**a. Functions for Factorial, Fibonacci, Prime Numbers, Palindrome**

<!DOCTYPE html>

<html>

<head>

<title>Functions</title>

</head>

<body>

<script>

function factorial(n) {

return n === 0 ? 1 : n \* factorial(n - 1);

}

function fibonacci(n) {

let series = [0, 1];

for (let i = 2; i < n; i++) {

series.push(series[i - 1] + series[i - 2]);

}

return series.slice(0, n);

}

function isPrime(num) {

if (num < 2) return false;

for (let i = 2; i <= Math.sqrt(num); i++) {

if (num % i === 0) return false;

}

return true;

}

function palindromeCheck(str) {

return str === str.split("").reverse().join("");

}

let num = parseInt(prompt("Enter a number:"));

document.write("Factorial: " + factorial(num) + "<br>");

document.write("Fibonacci: " + fibonacci(num) + "<br>");

document.write("Is Prime: " + isPrime(num) + "<br>");

document.write("Palindrome: " + palindromeCheck(String(num)));

</script>

</body>

</html>

**b. Text Box with Buttons**

<!DOCTYPE html>

<html>

<head>

<title>Function Buttons</title>

<script>

function factorial(n) {

return n === 0 ? 1 : n \* factorial(n - 1);

}

function fibonacci(n) {

let series = [0, 1];

for (let i = 2; i < n; i++) {

series.push(series[i - 1] + series[i - 2]);

}

return series.slice(0, n);

}

function isPrime(num) {

if (num < 2) return false;

for (let i = 2; i <= Math.sqrt(num); i++) {

if (num % i === 0) return false;

}

return true;

}

function palindromeCheck(str) {

return str === str.split("").reverse().join("");

}

function handleClick(action) {

let num = parseInt(document.getElementById("number").value);

if (isNaN(num)) {

alert("Please enter a valid number");

return;

}

let result;

switch (action) {

case "factorial":

result = "Factorial: " + factorial(num);

break;

case "fibonacci":

result = "Fibonacci: " + fibonacci(num);

break;

case "prime":

result = "Is Prime: " + isPrime(num);

break;

case "palindrome":

result = "Palindrome: " + palindromeCheck(String(num));

break;

}

document.getElementById("output").innerHTML = result;

}

</script>

</head>

<body>

<input type="text" id="number" placeholder="Enter a number">

<button onclick="handleClick('factorial')">Factorial</button>

<button onclick="handleClick('fibonacci')">Fibonacci</button>

<button onclick="handleClick('prime')">Prime</button>

<button onclick="handleClick('palindrome')">Palindrome</button>

<p id="output"></p>

</body>

</html>

**c. Validation for Registration Fields**

<!DOCTYPE html>

<html>

<head>

<title>Form Validation</title>

<script>

function validateForm() {

const name = document.getElementById("name").value;

const mobile = document.getElementById("mobile").value;

const email = document.getElementById("email").value;

const namePattern = /^[A-Za-z][A-Za-z0-9]{5,}$/;

const mobilePattern = /^[0-9]{10}$/;

const emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

if (!namePattern.test(name)) {

alert("Name must start with an alphabet, be alphanumeric, and at least 6 characters long.");

return false;

}

if (!mobilePattern.test(mobile)) {

alert("Mobile number must be 10 digits.");

return false;

}

if (!emailPattern.test(email)) {

alert("Invalid email format.");

return false;

}

alert("Form submitted successfully!");

return true;

}

</script>

</head>

<body>

<form onsubmit="return validateForm()">

<label>Name: <input type="text" id="name"></label><br>

<label>Mobile: <input type="text" id="mobile"></label><br>

<label>Email: <input type="text" id="email"></label><br>

<button type="submit">Submit</button>

</form>

</body>

</html>