1. Discuss TCP/IP Protocol

The TCP/IP (Transmission Control Protocol/Internet Protocol) suite is a set of communication protocols that form the foundation for the Internet and most local networks. Developed in the 1970s and 1980s, TCP/IP has become the standard protocol suite for networking, and it provides a reliable and efficient means of communication between devices across diverse networks. Here's an overview of the key components of the TCP/IP protocol suite:
TCP (Transmission Control Protocol):
Functionality: TCP provides reliable, connection-oriented communication between devices. It ensures that data is delivered in the correct order and without errors. To achieve this, TCP uses a combination of sequence numbers, acknowledgments, and retransmission mechanisms.
Connection Establishment and Termination: TCP follows a three-way handshake process to establish a connection and a four-way handshake to terminate it.
IP (Internet Protocol):
Functionality: IP is responsible for the addressing and routing of packets across a network. It provides an addressing scheme to uniquely identify devices in a network and enables the routers to forward packets from the source to the destination.
Versions: There are two main versions of IP in use today: IPv4 (Internet Protocol version 4) and IPv6 (Internet Protocol version 6). IPv4 is the older version and uses 32-bit addresses, while IPv6 uses 128-bit addresses and was introduced to address the exhaustion of IPv4 addresses.

2. What is a form ? Explain form components with example.

In the context of web development and user interfaces, a "form" refers to an HTML (Hypertext Markup Language) element that allows users to input data and submit it to a server for processing. Forms are a crucial part of web applications and websites, providing a means for users to interact with and input information.

A basic form structure in HTML looks like this:
`html
```
<form action="/submit" method="post">
  <!-- Form components go here -->
  <label for="username">Username:</label>
  <input type="text" id="username" name="username" required>

  <label for="password">Password:</label>
  <input type="password" id="password" name="password" required>

  <input type="submit" value="Submit">
</form>
```

Let's break down the components of this form:

1.<form> element: This is the container for all the form components. It has attributes like `action` (specifying where the form data should be sent) and `method` (specifying the HTTP method to be used for submitting the form, commonly "get" or "post").

2. `<label>` element: This is used to associate a text label with a form control, enhancing accessibility and user experience. The `for` attribute in the `<label>` tag should match the `id` attribute of the associated form control.

3. `<input>` element:This is one of the most versatile form elements. It is used for various types of user input, such as text, password, checkbox, radio button, etc. The `type` attribute determines the kind of input field, and the `id` and `name` attributes help identify and associate the input with labels.

- Text Input Example:
  html
  <label for="email">Email:</label>
  <input type="text" id="email" name="email" required>

- Password Input Example:
  html
  <label for="password">Password:</label>
  <input type="password" id="password" name="password" required>

4. `<textarea>` element:This is used for multiline text input. It's useful when users need to enter longer passages of text.

html
<label for="message">Message:</label>
<textarea id="message" name="message" rows="4" cols="50"></textarea>

5.`<select>` and `<option>` elements:These are used for dropdown menus. Users can select an option from the list.

html
<label for="cars">Choose a car:</label>
<select id="cars" name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="mercedes">Mercedes</option>
  <option value="audi">Audi</option>
</select>

6. `<button>` element:This is used to create a clickable button within the form. It can be used to trigger form submission or perform other actions.

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

3. Explain rowspan and colspan with an example.

Rowspan and colspan are attributes used in HTML tables to control the spanning of cells across multiple rows or columns.

Rowspan:
The rowspan attribute is used to specify how many rows a cell should span vertically. It allows a cell to occupy multiple rows in a table. For example:

```html
<table>
  <tr>
    <td rowspan="2">Cell 1</td>
    <td>Cell 2</td>
    <td>Cell 3</td>
  </tr>
  <tr>
    <td>Cell 4</td>
    <td>Cell 5</td>
  </tr>
  <tr>
    <td>Cell 6</td>
    <td>Cell 7</td>
    <td>Cell 8</td>
  </tr>
</table>
```

In this example, the first cell in the first row has a rowspan of 2, which means it spans across two rows. As a result, it occupies the space of the first cell in the second row as well. The table will be rendered with the first cell spanning two rows.

Colspan:
The colspan attribute is used to specify how many columns a cell should span horizontally. It allows a cell to occupy multiple columns in a table. For example:

```html
<table>
  <tr>
    <td>Cell 1</td>
    <td colspan="2">Cell 2</td>
    <td>Cell 3</td>
  </tr>
  <tr>
    <td>Cell 4</td>
    <td>Cell 5</td>
    <td>Cell 6</td>
    <td>Cell 7</td>
  </tr>
</table>
```

In this example, the second cell in the first row has a colspan of 2, which means it spans across two columns. As a result, it occupies the space of the second and third cells in the first row. The table will be rendered with the second cell spanning two columns.

Rowspan and colspan are useful when you want to merge cells in a table to create more complex layouts or to group related data.

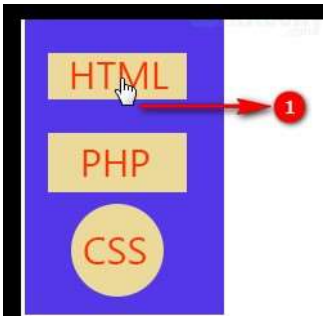
4. Write HTML code to generate the following output

Country	Population (In Crores)	
INDIA	1998	85
	1999	90
	2000	100
USA	1998	30
	1999	35
	2000	40
UK	1998	25
	1999	30
	2000	35

```
<!DOCTYPE html>
<html>

<head>
  <title>Population by Country</title>
</head>

<body>
  <h2>Population by Country</h2>
  <table border="1" style="border-collapse:collapse">
    <tr>
      <th>Country</th>
      <th colspan="2">Population(In crores)</th>
    </tr>
    <tr>
      <td rowspan="3">INDIA</td>
      <td>1998</td>
      <td>85</td>
    </tr>
    <tr>
      <td>1999</td>
      <td>90</td>
    </tr>
```

```
        <tr>
          <td>2000</td>
          <td>100</td>
        </tr> <tr>
          <td rowspan="3">USA</td>
          <td>1998</td>
          <td>30</td>
        </tr>
        <tr>
          <td>1999</td>
          <td>35</td>
        </tr>
        <tr>
          <td>2000</td>
          <td>40</td>
        </tr> <tr>
          <td rowspan="3">UK</td>
          <td>1998</td>
          <td>25</td>
        </tr>
        <tr>
          <td>1999</td>
          <td>30</td>
        </tr>
        <tr>
          <td>2000</td>
          <td>35</td>
        </tr>
    </table>
</body>

</html>
```

5. Create an Image Map for the following



Index.html:

```
 <!DOCTYPE html>

<html>
<body>

<h2>Image Maps</h2>
<p>Click on the computer, the phone, or the cup of coffee to go to a new page
and read more about the topic:</p>

<img src="workplace.jpg" alt="Workplace" usemap="#workmap" width="400"
height="379">

<map name="workmap">
  <area shape="rect" coords="34,44,270,350" alt="Computer"
href="Computer.html">
  <area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.html">
  <area shape="circle" coords="337,300,44" alt="Cup of coffee"
href="coffee.html">
</map>

</body>
</html>
```

Computer.htm:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Computer</title>
</head>
<body>
    <h2>
        Computer
    </h2>
    <img src="mac.jpg" alt="" width="300">
    <p>Wikipedia says:</p>
    <p>A computer is a device that can be instructed to carry out arbitrary
sequences of arithmetic or logical operations automatically.
    </p>
    <p>Read more about computer on https://en.wikipedia.org/wiki/Computer</p>
</body>
</html>
```

Phone.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Phone</title>
</head>
<body>
    <h2>
        Phone
    </h2>
    <img src="cellphone.jpg" alt="" width="300">
    <p>Wikipedia says:</p>
    <p>A telephone, or phone, is a telecommunications device that permits two
or more users to conduct a conversation when they are too far apart to be
heard directly.</p>
    <p>Read more about phone on https://en.wikipedia.org/wiki/Telephone</p>
</body>
</html>
```

Coffee.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>coffee</title>
</head>
<body>
    <h2>
        Coffee
    </h2>
    <img src="coffeehouse2.jpg" alt="" width="300">
    <p>Wikipedia says:</p>
    <p>Coffee is a brewed drink prepared from roasted coffee beans, which are
the seeds of berries from the Coffea plant.
    </p>
    <p>Read more about coffee on https://en.wikipedia.org/wiki/Coffee</p>
</body>
</html>
```

6. Using the HTML tags create the following form


```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Form</title>
</head>

<body>
    <h2>User Registration</h2>
    <p>Please complete the following form to register with our site:</p>
    <form action="">
        <label for="user-name">User Name:</label>
        <input type="text" id="user-name"><br>

        <label for="password">Password:</label>
        <input type="password" id="password"><br>

        <label for="comfirm-password">Comfirm Password:</label>
        <input type="password" id="comfirm-password"><br><br>

        <label for="first-name">First Name:</label>
        <input type="text" id="first-name"><br>

        <label for="last-name">Last Name:</label>
```

```
        <input type="text" id="last-name"><br><br>

        <label for="email">Email Address:</label>
        <input type="email"><br>

        <label for="gender">Gender:</label><br>

        <input type="radio" name="gender" id="male">
        <label for="male">Male</label><br>

        <input type="radio" name="gender" id="female">
        <label for="female">Female</label><br><br><br>

        <label for="select">How did you hear about us? </label>
        <select name="op" id="op">
            <option value="frst-op">First Option</option>
            <option value="sec-op">Second Option</option>
            <option value="thrd-op">Third Option</option>
            <option value="frt-op">Forth Option</option>
        </select><br>
        <br>
        <label for="checkbox">Please select this tex box if you wish to be
added to our mailing list  We will not pass on your details to any third
party.</label>
        <input type="checkbox" name="checkbox" id="checkbox"><br><br>

        <input type="button" name="button" id="button" value="Register Now">
    </form>
</body>

</html>
```

7. Create an HTML table as given below

Organization Logo			Internship Data					
			Internship Details					
S.No.	Candidate Name	DOB	Job Code	Job Title	Date of Joining	Duration	Stipend	Qualification

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Table2</title>
</head>
<body>
    <table border="1" style="border-collapse: collapse; text-align: center;">
        <tr>
            <td colspan="3">Organization Logo</td>
            <td colspan="6">Internship-Data</td>
        </tr>
        <tr>
            <td rowspan="2">Sr.No</td>
            <td rowspan="2">Candidate name</td>
            <td rowspan="2">DOB</td>
            <td colspan="5">Intership Details</td>
            <td rowspan="2">Qualifiactions</td>
        </tr>
        <tr>
            <td>Job Code</td>
            <td>Job Title</td>
            <td>Date of joining</td>
            <td>Duration</td>
            <td>Stipend</td>
        </tr>

    </table>
</body>
</html>
```

8. Explain the structure of the HTML webpage with an example.

The structure of an HTML webpage typically includes several essential elements that define the document's layout, content, and other meta-information. Here's a basic example of the structure of an HTML webpage:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My First Webpage</title>
    <!-- Additional meta tags, stylesheets, or scripts can be included here -->
</head>
<body>

    <header>
        <h1>Welcome to My Website</h1>
        <!-- Navigation menu or other header content goes here -->
    </header>

    <main>
        <section>
            <h2>About Me</h2>
            <p>This is a brief introduction about myself.</p>
        </section>

        <section>
            <h2>My Interests</h2>
            <ul>
                <li>Web Development</li>
                <li>Programming</li>
                <li>Reading</li>
            </ul>
        </section>
    </main>

    <footer>
        <p>&copy; 2023 My First Webpage. All rights reserved.</p>
        <!-- Additional footer content goes here -->
    </footer>

</body>
</html>
```

Let's break down the structure:

1. `<!DOCTYPE html>`: Declares the HTML version being used (HTML5 in this case).

2. `<html lang="en">`: The root element that wraps the entire HTML content. The `lang` attribute specifies the language of the document.

3. `<head>`: Contains meta-information about the HTML document, such as character set, viewport settings, and the document's title.

4. `<meta charset="UTF-8">`: Specifies the character encoding for the document.

5. `<meta name="viewport" content="width=device-width, initial-scale=1.0">`: Sets the viewport configuration for responsive design.

6. `<title>`: Defines the title of the HTML document, which appears in the browser tab.

7. `<body>`: Contains the main content of the HTML document.

8. `<header>`: Typically contains the heading or introductory content of the webpage, as well as navigation elements.

9. `<main>`: Contains the main content of the webpage, often organized into sections.

10. `<section>`: Represents a thematic grouping of content, such as different sections of a webpage.

11. `<h1>`, `<h2>`: Heading elements to define the hierarchy of the content.

12. `<p>`: Paragraph element for text content.

13. `<ul>`, `<li>`: Unordered list and list item elements for creating lists.

14. `<footer>`: Contains footer content, such as copyright information or links to related pages.

9. Write short note on the following:

Website :

A website is a collection of related web pages that are typically identified by a common domain name and are accessible over the internet. Websites are created and maintained to serve various purposes, such as providing information, offering products or services, facilitating communication, or serving as a platform for entertainment.

Key components of a website include:

1. **Web Pages:** These are individual documents containing text, images, multimedia, and other elements. Web pages are linked together to form a website.

2. **Domain Name:** A unique, human-readable address that identifies a website on the internet (e.g., www.example.com).

3. **Web Hosting:** The storage and maintenance of website files on servers, making them accessible to users worldwide.

4. **Navigation:** The structure and layout of a website, including menus, links, and other elements that enable users to move between pages.

5. **Content:** The information presented on the website, which can include text, images, videos, and interactive elements.

6. **Design and Layout:** The visual appearance and organization of a website, including the use of colors, fonts, and graphics.

7. **Responsive Design:** Ensuring that a website is accessible and functions well on various devices, such as desktop computers, tablets, and smartphones.

8. **Security:** Measures taken to protect the website and its users from potential threats, such as data breaches and cyberattacks.

Websites play a crucial role in the modern digital era, serving as a primary means of communication, information dissemination, and business transactions. They come in various types, including personal blogs, corporate sites, e-commerce platforms, social networks, and more. The design and content of a website are essential factors in providing a positive user experience and achieving the site's intended goals.

Webpage:

A webpage is a single document or file on the World Wide Web that is intended to be viewed in a web browser. It is a fundamental component of a website and is written in languages such as HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), and often includes elements of JavaScript for interactivity. Webpages can vary widely in content, structure, and purpose, catering to diverse needs and objectives.

Key features of a webpage include:

1. **HTML Structure:** HTML is used to structure the content of a webpage, defining elements such as headings, paragraphs, images, links, and more. It provides the basic framework for the page.

2. **CSS Styling:** CSS is employed to control the presentation and layout of the HTML elements. It dictates the visual aspects of the webpage, including colors, fonts, spacing, and positioning.

3. **Multimedia Elements:** Webpages can include various multimedia elements, such as images, videos, audio files, and interactive components to enhance user engagement.

4. **Hyperlinks:** Hyperlinks, or simply links, connect different webpages, allowing users to navigate between them. Links can point to pages within the same website or to external sources.

5. **Scripts (JavaScript):** JavaScript is often used to add interactivity and dynamic features to webpages. It allows for actions like form validation, animations, and real-time updates without requiring a page reload.

6. **Metadata:** Metadata, such as page titles, meta descriptions, and meta tags, provides information about the webpage to search engines and helps in search engine optimization (SEO).

7. **Responsive Design:** Modern webpages are designed to be responsive, meaning they can adapt to different screen sizes and devices, ensuring a consistent user experience across desktops, tablets, and smartphones.

8. **Forms:** Webpages may contain forms that allow users to submit information, such as contact details or online purchases. Form data is typically processed by server-side scripts.

Webpages collectively form a website, and the combination of well-designed and interconnected pages contributes to a cohesive and user-friendly online experience. Websites serve a wide range of purposes, including information dissemination, entertainment, communication, and e-commerce.

IP address:

An IP (Internet Protocol) address is a numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication. It serves two main functions: host or network interface identification and location addressing. IP addresses enable devices to send and receive data within a network and across the internet.

There are two types of IP addresses:

1. **IPv4 (Internet Protocol version 4):** This is the most widely used version of IP addresses. An IPv4 address consists of a 32-bit numerical value, often represented as four sets of numbers separated by dots (e.g., 192.168.0.1). However, the increasing number of devices connected to the internet has led to the exhaustion of available IPv4 addresses.

2. **IPv6 (Internet Protocol version 6):** To address the limitation of IPv4 addresses, IPv6 was introduced. IPv6 uses a 128-bit address format, represented as eight groups of hexadecimal values separated by colons (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334). IPv6 provides a significantly larger pool of unique addresses, accommodating the growing number of internet-connected devices.

IP addresses play a crucial role in internet communication by allowing devices to find and communicate with each other. When you access a website, send an email, or perform any online activity, your device is assigned an IP address. This address is used to route data packets between devices across the internet.

Dynamic Host Configuration Protocol (DHCP) is often used to automatically assign IP addresses to devices on a network, while static IP addresses are manually configured and remain constant. Network Address Translation (NAT) allows multiple devices within a local network to share a single public IP address.

In summary, IP addresses are the backbone of internet communication, providing a way for devices to identify each other and exchange data. They are essential for the functioning of the global network infrastructure.

WWW:

"WWW" stands for the World Wide Web. It is a system of interconnected documents and resources, linked by hyperlinks and URLs (Uniform Resource Locators), that are accessed and viewed via the internet. The World Wide Web is a major component of the internet and plays a central role in the way people access and share information online.

Key components and characteristics of the World Wide Web include:

1. **Web Pages:** These are individual documents containing text, images, multimedia, and other elements. Web pages are written in languages like HTML (Hypertext Markup Language) and are typically viewed in web browsers.

2. **Hyperlinks:** Hyperlinks, or links, connect web pages to each other, enabling users to navigate between different sites and content. Clicking on a hyperlink redirects the user to another web page or resource.

3. **URLs (Uniform Resource Locators):** URLs are addresses used to locate resources on the web. They specify the protocol (such as "http" or "https"), the domain name, and the specific path to the resource.

4. **Web Browsers:** Software applications like Chrome, Firefox, Safari, and others allow users to access and view web pages. Browsers interpret HTML and display the content of web pages.

5. **HTTP and HTTPS:** HyperText Transfer Protocol (HTTP) and its secure version, HTTPS, are communication protocols used for transferring data on the web. HTTPS provides a secure, encrypted connection, particularly important for sensitive transactions like online banking and shopping.

6. **Web Servers:** These are computers that store and deliver web pages to users when requested. Web servers host websites and respond to browser requests by providing the requested content.

7. **Web Development Technologies:** Various technologies, such as JavaScript, CSS (Cascading Style Sheets), and server-side scripting languages, contribute to the design, functionality, and interactivity of web pages.

The World Wide Web was invented by Sir Tim Berners-Lee in 1989, and its development has significantly impacted the way information is accessed, shared, and communicated globally. It has become an integral part of daily life, serving as a platform for communication, commerce, entertainment, and information dissemination.

10. Create an external style sheet "style.css" with the following specifications for the various HTML tags as follows

<BODY>:
Background color should be Aqua( )
Foreground color should be Navy( )
<H1>
Should be underlined and center aligned

<P>
First line should be indented to 60px
Font-weight should be normal

Index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>External CSS</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <h1>This is the text in navy color</h1>
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Ab animi
officia dolorem. Repellendus ab aperiam, quos adipisci temporibus animi
voluptate!</p>
</body>
</html>
```

Style.css:

```
body{
     background-color: aqua;

}

h1 {
     color: navy;
    text-decoration: underline;
    text-align: center;
    text-indent: 60px;
}
p{
    text-indent: 60px;
    font-weight: normal;
}
```

11. Write the HTML code to create a link from the HTML document to the stylesheet (style.css).Where in the HTML document should the code be placed?

Index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>External CSS</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <h1>This is the text in navy color</h1>
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Ab animi
officia dolorem. Repellendus ab aperiam, quos adipisci temporibus animi
voluptate!</p>
</body>
</html>
```

Style.css:

```
body{
     background-color: aqua;

}

h1 {
     color: navy;
    text-decoration: underline;
    text-align: center;
    text-indent: 60px;
}
p{
    text-indent: 60px;
    font-weight: normal;
}
```

12. Write a Javascript code which checks the contents entered in a forms text element. If the text entered is in the lower cases convert to upper case.

```
<!DOCTYPE html>
<html>
<head>
  <script>
    function convertToUpperCase() {
      var inputText = document.getElementById("myInput").value;
      var convertedText = inputText.toUpperCase();
      document.getElementById("myInput").value = convertedText;
```

```
    }
  </script>
</head>
<body>
  <form>
    <label for="myInput">Enter Text:</label>
    <input type="text" id="myInput" name="myInput">
    <button type="button" onclick="convertToUpperCase()">Convert to Uppercase</button>
  </form>
</body>
</html>
```

13.Explain various datatypes used in Javascript.

JavaScript has several built-in data types that are used to store different kinds of values. Here are the main data types in JavaScript:

1. Number: Represents numeric values, including integers and floating-point numbers. For example: `let age = 25;` or `let price = 9.99;`

2. String: Represents a sequence of characters enclosed in single quotes (') or double quotes (""). For example: `let name = "John";` or `let message = 'Hello, world!';`

3. Boolean: Represents a logical value, either `true` or `false`. It is often used in conditional statements and comparisons. For example: `let isLogged = true;` or `let isFound = false;`

4. Null: Represents the intentional absence of any object value. It is a special value that indicates the absence of a value. For example: `let myVariable = null;`

5. Undefined: Represents a variable that has been declared but has not been assigned a value. It is the default value of variables that have not been initialized. For example: `let myVariable;` or `let myVariable = undefined;`

6. Object: Represents a collection of key-value pairs, where each key is a string (or symbol) and each value can be of any data type. Objects are used to store and organize complex data structures. For example:

```javascript
let person = {
  name: "John",
  age: 25,
  isStudent: true
};
```

7. Array: Represents an ordered list of values, enclosed in square brackets ([]). Arrays can store multiple values of any data type, and each value is assigned an index starting from 0. For example:

```javascript
let fruits = ["apple", "banana", "orange"];
```

8. Symbol: Represents a unique identifier. Symbols are often used as keys in objects to avoid naming conflicts. They are created using the `Symbol()` function. For example:

```javascript
let id = Symbol("uniqueId");
```

These are the main data types in JavaScript. Understanding and working with these data types is fundamental to programming in JavaScript.

14.Write a Javascript to find factorial of a number.

```
<!DOCTYPE html>
<html>
<head>
  <script>
    function calculateFactorial() {
      var number = parseInt(document.getElementById("numberInput").value);
      var factorial = 1;

      for (var i = 1; i <= number; i++) {
        factorial *= i;
      }

      document.getElementById("result").innerHTML = "The factorial of " + number + " is: "
+ factorial;
    }
  </script>
</head>
<body>
  <label for="numberInput">Enter a number:</label>
  <input type="number" id="numberInput">
  <button onclick="calculateFactorial()">Calculate Factorial</button>
  <p id="result"></p>
</body>
</html>
```

15.Create a form for student information. Write Java Script code to find total, average, result and grade.

```
<!DOCTYPE html>
<html>
<head>
  <script>
```

```
    function calculateResult() {
      var marks1 = parseInt(document.getElementById("marks1").value);
      var marks2 = parseInt(document.getElementById("marks2").value);
      var marks3 = parseInt(document.getElementById("marks3").value);

      var total = marks1 + marks2 + marks3;
      var average = total / 3;

      document.getElementById("total").innerHTML = "Total Marks: " + total;
      document.getElementById("average").innerHTML = "Average Marks: " +
average.toFixed(2);

      if (average >= 50) {
        document.getElementById("result").innerHTML = "Result: Pass";
        document.getElementById("grade").innerHTML = "Grade: A";
      } else {
        document.getElementById("result").innerHTML = "Result: Fail";
        document.getElementById("grade").innerHTML = "Grade: B";
      }
    }
  </script>
</head>
<body>
    <h2>Marks Obtained by Student</h2>
  <form>
    <label for="marks1">Marks of 1<sup>st</sup>  subject:</label>
    <input type="number" id="marks1" name="marks1"><br>

    <label for="marks2">Marks of 2<sup>nd</sup> subject:</label>
    <input type="number" id="marks2" name="marks2"><br>

    <label for="marks3">Marks of 3<sup>rd</sup> subject:</label>
    <input type="number" id="marks3" name="marks3"><br>

    <button type="button" onclick="calculateResult()">Calculate</button>
  </form>

  <p id="total"></p>
  <p id="average"></p>
  <p id="result"></p>
  <p id="grade"></p>
</body>
</html>
```

16. What is an event? Explain all the JavaScript Dom events with the help of examples

.

17.Explain GET and POST request methods.

GET and POST are two commonly used HTTP request methods used to send data from a client (such as a web browser) to a server.

1. GET Method:
- GET is the default method used by web browsers when you enter a URL or click on a link.
- It is used to retrieve data from a server.
- The data is appended to the URL as query parameters.
- GET requests are visible in the browser's address bar and can be bookmarked and cached.
- GET requests are limited in the amount of data they can send (typically around 2048 characters).
- GET requests are considered "safe" and "idempotent" as they should not have any side effects on the server.

2. POST Method:
- POST is used to send data to the server to create or update a resource.
- The data is sent in the body of the request, not visible in the URL.
- POST requests are not cached and cannot be bookmarked.
- POST requests can send larger amounts of data compared to GET requests.
- POST requests are not considered "safe" or "idempotent" as they can have side effects on the server, such as creating a new resource or updating an existing one.

18.Design A JavaScript to display whether given number is prime or not.

```
<html>
<head>
  <script>
    function checkPrime() {
      var number = parseInt(document.getElementById("numberInput").value);
      var isPrime = true;

      if (number === 1) {
        isPrime = false;
      } else if (number > 1) {
        for (var i = 2; i <= Math.sqrt(number); i++) {
          if (number % i === 0) {
            isPrime = false;
            break;
          }
        }
      }

      if (isPrime) {
        document.getElementById("result").innerHTML = number + " is a prime number.";
      } else {
        document.getElementById("result").innerHTML = number + " is not a prime number.";
      }
```

```
    }
  </script>
</head>
<body>
  <label for="numberInput">Enter a number:</label>
  <input type="number" id="numberInput">
  <button onclick="checkPrime()">Check Prime</button>
  <p id="result"></p>
</body>
</html>
```

19.Explain briefly about Built in Java script Objects?

Built-in JavaScript objects are pre-defined objects that provide useful functionality and methods to perform various operations in JavaScript. These objects are available globally and can be accessed and used without the need for explicit instantiation.

Here are some commonly used built-in JavaScript objects:

1. Math: Provides mathematical operations and constants.
   - Example methods: `Math.random()`, `Math.floor()`, `Math.max()`, `Math.min()`

2. Date: Represents dates and times.
   - Example methods: `Date.now()`, `Date.getFullYear()`, `Date.getMonth()`, `Date.getDate()`

3. String: Represents a sequence of characters.
   - Example methods: `String.length`, `String.charAt()`, `String.indexOf()`, `String.toUpperCase()`

4. Array: Represents an ordered collection of elements.
   - Example methods: `Array.length`, `Array.push()`, `Array.pop()`, `Array.join()`

5. Object: Represents a collection of key-value pairs.
   - Example methods: `Object.keys()`, `Object.values()`, `Object.assign()`

6. JSON: Provides methods for working with JSON (JavaScript Object Notation) data.
   - Example methods: `JSON.parse()`, `JSON.stringify()`

7. RegExp: Represents regular expressions for pattern matching and manipulation.
   - Example methods: `RegExp.test()`, `RegExp.exec()`, `RegExp.match()`

These built-in objects offer a wide range of functionality and methods to perform common tasks in JavaScript. By utilizing these objects, you can enhance your code and leverage their capabilities to simplify and optimize your JavaScript programs.

20.Write a JavaScript program to create registration form and validate all fields using form validation.

21.Write a JavaScript program to calculate area of rectangle using

Function.

```
<html>
<head>
<title> Area of Rectangle </title>
</head>
<body>
<script>
var length = parseInt(prompt("Type the length of a Rectangle : "));;
var width = parseInt(prompt("Type the width of a Rectangle : "));;
function areaRectangle(width, length)
{
var area = width * length;
return area;
}
document.write("Area of
Rectangle="+areaRectangle(width,length)+"<br>");
</script>
</body>
</html>
```

22.Explain the CSS selectors .

1 Element Selectors
Element selectors select all instances of a given HTML element. You can select all elements by using the universal element selector, which is the * (asterisk) character.
2 Class Selectors
A class selector allows you to simultaneously target different HTML elements regardless of their position in the document tree. If a series of HTML elements have been labeled with the same class attribute value, then you can target them for styling by using a class selector, which takes the form: period (.) followed by the class name.
3 Id Selectors
An id selector allows you to target a specific element by its id attribute regardless of its type or position. If an HTML element has been labeled with an id attribute, then you can target it for styling by using an id selector, which takes the form: pound/hash (#) followed by the id name.

4 Attribute Selectors
An attribute selector provides a way to select HTML elements either by the presence of an element attribute or by the value of an attribute. This can be a very powerful technique, but because of uneven support by some of the browsers, not all web authors have used them.

23. What is JQuery? Explain JQuery Selectors?

24. What is a box model? And what are the different elements of a box model?

25. Difference between JavaScript and JQuery

Parameters	JavaScript	JQuery
Basics	JS is a programming language. It is a scripting language that is dynamic and helps in web development.	It is a JS library.
Complexity	It is more complex since the programmer has to write the complete JavaScript code in a program.	It is comparatively less complex. A programmer only has to write the important JQuery code.
Consumption of Time	Since one has to write the whole script, it consumes much more time.	It consumes comparatively much less time since it makes programming easy and fast.
Multi-Browser Compatibility	To handle multi-browser compatibility, a developer has to develop code of their own.	One doesn't need to focus on their app's issue with compatibility on multiple browsers. This feature comes as a prerequisite.
Browser Support	Every browser supports JavaScript. We don't need to include any additional plugins to make JS work.	We have to include the JQuery library's URL in the page's header so that JQuery works.
Inter-Dependency	JQuery forms a part of JS. The JS code does not necessarily depend on	JQuery always depends on JS- since it is a JavaScript library.

	JQuery.	
Lines of Code	The code can be long and complicated. It takes time to program using JavaScript.	It consists of very few lines of code; it is easy to operate and work with.
Uses	It is a very crucial programming language used for web designing and programming desktop programs and servers.	It optimises the working of the JavaScript language. We can make all apps and websites more interactive, fast, and efficient while still decreasing the complexity of development.
Type of Approach	It is a very weakly typed approach used in programming.	It is a fast, simple, and easy approach used in programming.
DOM	It is very slow for the creation of DOM.	It is capable of creating DOM much faster.

26.Define AJAX. Explain working of Ajax

AJAX, which stands for Asynchronous JavaScript and XML, is a set of web development techniques that allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This enables a more dynamic and interactive user experience without requiring the entire page to be reloaded.

Here's a breakdown of how AJAX works:

1. **Asynchronous Request:** The term "asynchronous" in AJAX means that web applications can send and receive data from a server without interfering with the display and behavior of the existing page. Traditional web applications would typically require a full page reload when interacting with the server, causing a noticeable interruption in the user experience.

2. **JavaScript:** AJAX relies heavily on JavaScript, a scripting language that runs in the web browser. JavaScript is used to create dynamic content, manipulate the Document Object Model (DOM), and make asynchronous requests to the server.

3. **XMLHttpRequest Object:** AJAX uses the XMLHttpRequest object to interact with the server. This object allows JavaScript to make HTTP requests to the server and handle the server's response. Despite its name, XMLHttpRequest can be used to send and receive data in various formats, not just XML.

4. **Server-Side Processing:** On the server side, there are scripts or programs that handle the incoming AJAX requests. These scripts process the request, perform any necessary computations or database queries, and then send a response back to the client.

5. **Data Exchange Formats:** While XML is part of the name, AJAX is not limited to using XML for data exchange. JSON (JavaScript Object Notation) has become a popular alternative due to its simplicity and ease of use. The server typically sends data back to the client in a format (XML or JSON) that can be easily processed by JavaScript.

6. **DOM Manipulation:** Once the client receives the server's response, JavaScript can dynamically update the content of the web page without requiring a full reload. This manipulation of the DOM allows for seamless and quick updates to specific parts of a page.

In summary, AJAX enables web pages to request and exchange data with a server asynchronously, creating a smoother and more responsive user experience. This technology has become a standard in modern web development, contributing to the development of interactive and dynamic web applications.

27.Explain AJAX request

An AJAX request is a way for a webpage to communicate with a server in the background without having to reload the entire page. It's like sending a secret message to someone and getting a response without interrupting what you're doing.

Imagine you're playing a game on your computer and you need some extra information from a friend who has that information. Instead of pausing the game and calling your friend, you can send them a quick message without leaving the game. Your friend receives the message, finds the information you need, and sends it back to you. All of this happens without you having to stop playing the game.

In the same way, an AJAX request allows a webpage to send a message to a server asking for some information or data. The server receives the request, processes it, and sends back a response with the requested information. This happens in the background, without the user having to wait for the entire page to reload.

For example, let's say you're on a webpage that shows the weather. When you want to know the current temperature, instead of refreshing the entire page, an AJAX request can be sent to the server asking for the temperature. The server then responds with the temperature, and the webpage updates that information without reloading the whole page.

AJAX requests are really useful because they make webpages more interactive and responsive. They allow us to fetch data from a server without disrupting the user's experience. It's like having a secret messaging system that lets us get information without having to pause or reload everything.

an AJAX request is a way for a webpage to send a message to a server and get a response without reloading the entire page. It's like sending a secret message to someone and getting a quick response without interrupting what you're doing.

Discuss the Jquery Dom manipulation methods

Anwer on this link https://www.tutorialspoint.com/jquery/jquery-dom.htm

Explain Jquery Dom traversing

Answer to the above question in the link given below

https://www.tutorialspoint.com/jquery/jquery-traversing.htm

Explain jQuery Dom traversing with examples.

Discuss the jQuery Dom manipulation methods with illustrative examples.

Unit 4

Write all the looping statements. Write any PHP code to explain for a loop.

List the functions to create a pattern. Preg_match, Preg_matchall, Preg_replace, Preg_split

Discuss the Scalar and compound and special data types in Php .Illustrate with the help of an example

Write three types of an array and write PHP code to explain any one of them.

Write PHP code to explain for each loop.

PHP program to print array elements using foreach loop.

Search an array for the value "RED", and then replace it with "pink"

List the functions to create a pattern.

Explain any 5 Predefined Array functions

Why we use isset(), unset() functions?

What are gettype(), settype() functions?

Explain 5 built-in functions in PHP.(Excluding string functions)

Explain Indexed arrays and Associative Arrays with example

7. Explain 5 built-in functions in PHP.(Excluding string functions) 10 2 108. Explain Indexed arrays and Associative Arrays with example

Why we use array_unshift(), array_shift(), array_push(), array_pop(), array_splice(). Explain with example

Why we use key(), current(), prev() and end() functions. Explain with example. 10 2 141. Why we use array_unshift(), array_shift(), array_push(), array_pop(), array_spl

String Related Library function:

strtolower():

strtoupper():

ucfirst():

lcfirst():

ucwords():

strrev():

What is pass by reference? Write a program in Php to pass value by reference to a function

Discuss the purpose of the following functions array_unshift(), array_shift(), array_push(), array_pop(), array_splice().Write a PHP script to illustrate the use of these functions using associative or indexed arrays.

Unit 5

List and Explain different super global arrays

What is cookie? Give example in PHP

How to create /Retrieve a cookie in PHP

Why we use $_REQUEST variable? Write a note on various library functions available in PHP?

Write about the following built-in functions:

1) Require()

10) Join()

11) substr()

3) Include()

12) Strcmp()

4) Chop()

13) Strnotcmp()

5) Trim()

14) strlen()

15) strchar()

7) strtoupper()

16) stropos()

8) strtolower()

17) strreplace()

9) explode()

18)substr_replace()

Write a PHP script which takes the username, password and e-mail values from user and checks whether the user has filled the textboxes or not. Also, check the e-mail field whether it is incorrect format or not.

Explain the concept of anonymous function in detail.

Explain the following string functions with example. i) str_replace() ii) ucwords() iii) strlen() iv) strtoupper() c) Explain the concept of anonymous function in detail.

Write the syntax and explain the following with example -settype function -gettype function

Why we use array_unshift(), array_shift(), array_push(), array_pop(), array_splice(). Explain with example

Write a PHP code for sorting of an array by taking the user input from single textbox of HTML (Comma Separated).

Write a PHP script to read an integer from user and print the multiplication table of that integer. The number of rows in the Multiplication table is also input by the user.

Discuss Form Sanitization and write a code to illustrate the different form sanitization filters