Practical No.1

Aim: To study the basic structure of HTML.

Introduction to HTML (HyperText Markup Language)

HTML (HyperText Markup Language) is the standard language used to create and structure content on the web. It provides the basic building blocks for web pages, such as text, images, links, tables, forms, and more. HTML consists of a set of rules and tags that define the structure of a webpage, allowing browsers to render content in a user-friendly way.

ALGORITHM :
1. Start by defining the document type declaration (<!DOCTYPE html>).
2. Create the <html> element, which contains the entire HTML document.
3. Inside the <html> element, create the <head> element.
4. Inside the <head> element, add metadata such as the <title> of the document and any <meta> tags for specifying character encoding or other information.
5. Outside of the <head> element, create the <body> element.
6. Inside the <body> element, add the main content of your webpage, including text, images, links,

CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>My HTML Page</title>
</head>
<body>
 <p>This is a basic HTML page.</p>
</body>
</html>
```

OUTPUT 1:

This is a basic HTML page.

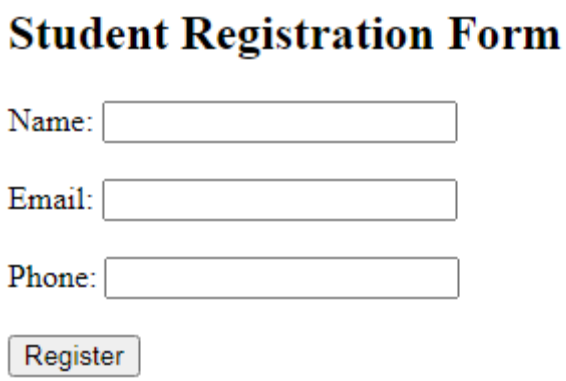
Practical No.2

Aim: HTML program to create student Registration form.

Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Student Registration Form</title>
</head>
<body>
   <h2>Student Registration Form</h2>
   <form action="#" method="post">
      <label for="name">Name:</label>
      <input type="text" id="name" name="name" required><br><br>

      <label for="email">Email:</label>
      <input type="email" id="email" name="email" required><br><br>

      <label for="phone">Phone:</label>
      <input type="tel" id="phone" name="phone" required><br><br>

      <button type="submit">Register</button>
   </form>
</body>
</html>
```

Output:



Practical No:3

Aim:HTML program that demonstrates the use of semantic elements, lists, and links:

Explanation of the HTML Program:

1. **<!DOCTYPE html>**: Declares the document type and version (HTML5).
2. **<html>**: The root element that wraps all the content.
3. **<head>**: Contains metadata about the document (e.g., character set, viewport settings, and title).
4. **<body>**: The visible content of the webpage.

Inside the <body>, there are several semantic elements:

<header>: Defines the header section of the page, typically containing the website title and introductory content.

<nav>: Defines the navigation links, providing a menu for users to navigate between sections of the page.
<main>: Represents the main content of the webpage.
<section>: Groups related content and gives it a meaningful label (e.g., Home, Services, Contact).
Inside the "Services" section, a **<ul>** (unordered list) is used to display a list of services offered by the website.
<footer>: Contains the footer of the page, usually for copyright or additional information.

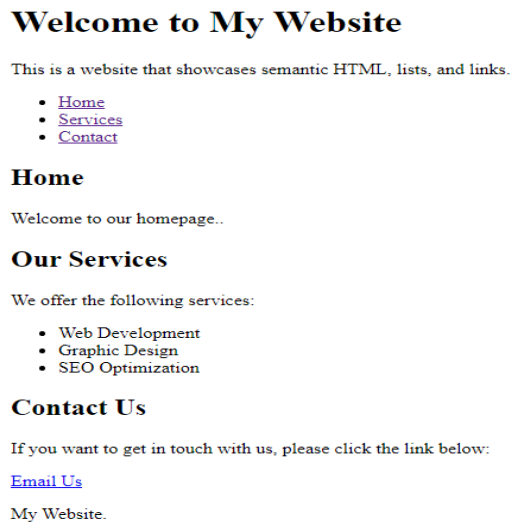
5. **Links**:The **<a>** (anchor) tag is used to create hyperlinks. In this example:
The navigation links use anchor tags with href attributes that point to section IDs (#home, #services, #contact).
There is also a mailto link that allows users to send an email by clicking on "Email Us".

CODE:

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

  <title>Semantic Elements with Lists and Links</title>
</head>
<body>

  <!-- Header section -->
  <header>
    <h1>Welcome to My Website</h1>
    <p>This is a website that showcases semantic HTML, lists, and links.</p>
  </header>

  <!-- Navigation bar -->
  <nav>
    <ul>
      <li><a href="#home">Home</a></li>
      <li><a href="#services">Services</a></li>
      <li><a href="#contact">Contact</a></li>
    </ul>
```

```
    </nav>

    <!-- Main content section -->
    <main>
        <section id="home">
            <h2>Home</h2>
            <p>Welcome to our homepage..</p>
        </section>

        <section id="services">
            <h2>Our Services</h2>
            <p>We offer the following services:</p>
            <ul>
                <li>Web Development</li>
                <li>Graphic Design</li>
                <li>SEO Optimization</li>
            </ul>
        </section>

        <section id="contact">
            <h2>Contact Us</h2>
            <p>If you want to get in touch with us, please click the link below:</p>
            <p><a href="mailto:contact@website.com">Email Us</a></p>
        </section>
    </main>

    <!-- Footer section -->
    <footer>
        <p>My Website.</p>
    </footer>

</body>
</html>
```

Output:



Welcome to My Website

This is a website that showcases semantic HTML, lists, and links.

- Home
- Services
- Contact

Home

Welcome to our homepage..

Our Services

We offer the following services:

- Web Development
- Graphic Design
- SEO Optimization

Contact Us

If you want to get in touch with us, please click the link below:

Email Us

My Website.

Practical 4

Aim:Program to implement Inline, Internal and External CSS.

Theory: CSS stands for **Cascading Style Sheets** and it is used for designing and responsiveness of web pages. It helps in maintaining the positioning of each and every element that should be displayed on the web page. It is highly used for creating interactive user interfaces.

CODE :

```
<html>
<head>
<title>CSS Example</title>
<!—Internal CSS-->
        <style>

            /*InternalCSS*/ h1 {

                color:blue;
                }
            p{
                font-size:16px;
            }

         </style>
         <!—External CSS-->

         <linkrel="stylesheet"type="text/css"href="styles.css">
</head>
<body>
<!—Inline CSS-->

<h1style="font-family:Arial,sans-serif;">Inline CSS Example</h1>
<pstyle="color:green;">This is a paragraph with inline CSS.</p>

<!—Internal CSS-->
<h1>Internal CSS Example</h1>
<p>This is a paragraph with internal CSS.</p>

<!—External CSS-->
<h1class="external">External CSS Example</h1>
<pclass="external">This is a paragraph with external CSS.</p>

</body>
</html>
```

OUTPUT:

Inline CSS Example

This is a paragraph with inline CSS.

Internal CSS Example

This is a paragraph with internal CSS.

External CSS Example

This is a paragraph with external CSS.

Practical No.5

Aim: HTML program using web topography.

Theory: The webpage uses Arial font with a line height for readability.Headings are styled with a specific color.Links are colored and styled without underlines.
This provides a clean and visually appealing layout using basic web typography principles.

Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Web Typography Example</title>
   <style>
      body { font-family: Arial, sans-serif; line-height: 1.6; margin: 20px; }
      h1 { color: #2c3e50; }
      p { font-size: 18px; }
      a { color: #3498db; text-decoration: none; }
   </style>
</head>
<body>
   <h1>Welcome to Web Typography</h1>
   <p>Typography is a crucial part of web design.</p>
   <p>Learn more about it <a href="#">here</a>.</p>
</body>
</html>
```

OUTPUT:

Practical No.6

Aim:HTML Program using responsive web design using css.

Theory:

- The `meta viewport` tag ensures the layout is responsive on mobile devices.
- The `flex` property in the CSS allows the boxes to adapt to different screen sizes.
- The media query changes the box layout to stack vertically when the screen width is 600px or less.

Code:

```
 <!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Responsive Web Design</title>
   <style>
      body { display: flex; flex-wrap: wrap; margin: 0; padding: 0; }
      .box { flex: 1 1 200px; margin: 10px; padding: 20px; background: #4CAF50; color: white;
text-align: center; }
      @media (max-width: 600px) {
         .box { flex-basis: 100%; }
      }
   </style>
</head>
<body>
   <div class="box">Box 1</div>
   <div class="box">Box 2</div>
   <div class="box">Box 3</div>
</body>
</html>
```

OUTPUT:

Practical No:7

Aim:HTML Program to insert images and multimedia file.

Theory:

- Replace `path/to/your/image.jpg`, `path/to/your/audio.mp3`, and `path/to/your/video.mp4` with the actual paths to your media files.
- You can adjust the `width` in the `<img>` and `<video>` tags as needed.
- This code will create a simple web page that displays an image, plays an audio file, and plays a video file.

CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Images and Multimedia</title>
</head>
<body>
  <h1>My Favorite Media</h1>

  <h2>Image</h2>
  <img src="path/to/your/image.jpg" alt="Description of Image" width="300">

  <h2>Audio</h2>
  <audio controls>
    <source src="path/to/your/audio.mp3" type="audio/mpeg">
    Your browser does not support the audio element.
  </audio>

  <h2>Video</h2>
  <video width="320" height="240" controls>
    <source src="path/to/your/video.mp4" type="video/mp4">
    Your browser does not support the video tag.
  </video>
</body>
</html>
```

OUTPUT:

Practical No:8

Aim:HTML Program to understand web accessibility.

Theory:

1. **Semantic HTML**: Using elements like `<header>`, `<main>`, `<section>`, and `<footer>` helps screen readers understand the structure of the page.
2. **Alt Text**: The `alt` attribute in the `<img>` tag provides a text description of the image for users who cannot see it.
3. **Labels for Form Elements**: Each `<input>` field has a corresponding `<label>` with the `for` attribute, which improves keyboard navigation and accessibility.

This simple page structure is designed to be accessible, making it easier for users with disabilities to interact with the content.

CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Accessible Web Page</title>
</head>
<body>
  <header>
    <h1>Welcome to Our Accessible Website</h1>
  </header>

  <main>
    <section>
      <h2>About Us</h2>
      <p>We strive to make our site accessible to all users.</p>

    </section>

    <section>
      <h2>Contact Us</h2>
      <form>
        <label for="name">Name:</label>
        <input type="text" id="name" name="name" required>

        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required>

        <button type="submit">Submit</button>
      </form>
    </section>
  </main>
```

```
  <footer>
    <p>2024-25 FY AIML</p>
  </footer>
</body>
</html>
```

OUTPUT:

Welcome to Our Accessible Website

About Us

We strive to make our site accessible to all users.

Contact Us

Name: Email: Submit

2024-25 FY AIML

Practical No:9

Aim:HTML Program using Web Development tools.

Theory:
1. **Meta Tags**: The `<meta name="viewport">` tag makes the page responsive, ensuring it looks good on different devices.
2. **CSS Styling**: The `<style>` tag is used for basic CSS to style the body and button.
3. **JavaScript Interactivity**: The `<script>` tag defines a simple JavaScript function that updates the content of a `<p>` element when the button is clicked.
This code provides a straightforward example of how to use essential web development tools in an HTML document.

CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Web Development Tools Example</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>
        body { font-family: Arial, sans-serif; margin: 20px; }
        button { background-color: #4CAF50; color: white; padding: 10px; border: none; cursor:
pointer; }
    </style>
</head>
<body>
    <h1>Welcome to Web Development</h1>
    <p>This is a demonstration of basic web development tools.</p>
    <button onclick="showMessage()">Click Me!</button>
    <p id="message"></p>

    <script>
        function showMessage() {
            document.getElementById('message').innerText = 'Button clicked! You are using
JavaScript.';
        }
    </script>
</body>
</html>
```

OUTPUT:

Practical No:10

Aim: To Study the Basic use of Javascript.

Theory:

1. **Inline Event Handling**: The `onclick` attribute in the `<button>` tags is used to call JavaScript functions when the buttons are clicked.

2. **JavaScript Functions**:
   - **`showMessage()`**: This function updates the text content of a paragraph with the ID `result` when the first button is clicked.
   - **`changeColor()`**: This function changes the background color of the webpage to a random color when the second button is clicked.

3. **Document Object Model (DOM) Manipulation**: JavaScript is used to access and modify HTML elements dynamically (e.g., updating text and changing styles).

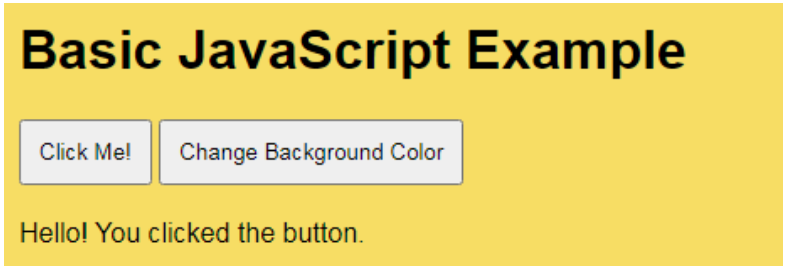
Copy the code into an HTML file and open it in a web browser.
Click the "Click Me!" button to see a message, and click the "Change Background Color" button to see the background color change.

CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Basic JavaScript Example</title>
   <style>
      body { font-family: Arial, sans-serif; margin: 20px; }
      button { padding: 10px; cursor: pointer; }
      #result { margin-top: 20px; }
   </style>
</head>
<body>
   <h1>Basic JavaScript Example</h1>
   <button onclick="showMessage()">Click Me!</button>
   <button onclick="changeColor()">Change Background Color</button>

   <p id="result"></p>

   <script>
      // Function to display a message
      function showMessage() {
         document.getElementById('result').innerText = 'Hello! You clicked the button.';
      }

      // Function to change background color
      function changeColor() {
         document.body.style.backgroundColor =
```

```
            'rgb(' + Math.floor(Math.random() * 256) + ',' +
            Math.floor(Math.random() * 256) + ',' +
            Math.floor(Math.random() * 256) + ')';
        }
    </script>
</body>
</html>
```

OUTPUT:



Conclusion: This simple example illustrates basic JavaScript concepts and how they can interact with HTML to create dynamic web pages.