SEIF MWITA MGENI – BCS/18/23/024/TZ

index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>HTML Structure and Tags</title>
   <link rel="icon" href="edugamez-logo.png" type="image/x-icon">
   <link rel="stylesheet" href="style.css">
</head>
<body>
   <!-- Navigation bar -->
   <div class="navbar-container">
      <nav class="navbar">
         <div class="logo">
            <img src="edugamez-logo.png" alt="EduGamez Logo" width="40" height="40">
            <span>Web Basics</span>
         </div>
         <div class="menu-toggle">
            <div class="bar"></div>
            <div class="bar"></div>
            <div class="bar"></div>
         </div>
         <ul class="nav-menu">
            <li><a href="index.html" class="active">Home</a></li>
            <li><a href="form.html">Form Demo</a></li>
            <li><a href="#books">Books</a></li>
            <li><a href="#hobbies">Hobbies</a></li>
            <li><a href="#schedule">Schedule</a></li>
         </ul>
      </nav>
   </div>

   <!-- Headings, paragraphs, horizontal rules, and line breaks -->
   <h1>Main Heading: HTML Basics</h1>
   <p>This is a paragraph demonstrating the use of paragraph tags. It contains text that flows
normally until a line break is inserted.</p>
   <br>
   <p>This is a second paragraph after a line break. Notice how it continues on a new
line.</p>

   <hr>

   <h2>Secondary Heading: More HTML Elements</h2>
   <h3>Tertiary Heading: Lists and Images</h3>
   <h4>Fourth Level Heading</h4>
   <h5>Fifth Level Heading</h5>
   <h6>Sixth Level Heading</h6>
```

```
    <hr>

    <!-- Unordered list of favorite books -->
    <section id="books">
        <h2>My Top 5 Favorite Books</h2>
        <ul>
            <li>To Kill a Mockingbird by Harper Lee</li>
            <li>1984 by George Orwell</li>
            <li>The Great Gatsby by F. Scott Fitzgerald</li>
            <li>Pride and Prejudice by Jane Austen</li>
            <li>The Catcher in the Rye by J.D. Salinger</li>
        </ul>
    </section>

    <!-- Ordered list of hobbies -->
    <section id="hobbies">
        <h2>My Top 5 Hobbies</h2>
        <ol>
            <li>Reading books</li>
            <li>Hiking in nature</li>
            <li>Cooking new recipes</li>
            <li>Playing the piano</li>
            <li>Photography</li>
        </ol>
    </section>

    <hr>

    <!-- Images with alt text -->
    <section>
        <h2>Images</h2>
        <figure>
            <img src="seif.jpg" width="500px" height="500px" alt="image of seif">
            <figcaption>This is my image</figcaption>
        </figure>

        <figure>
            <img src="soldier.jpg" width="500px" height="300px" alt="image of Terrorist">
            <figcaption>This is Terrorist image </figcaption>
        </figure>
    </section>

    <hr>

    <!-- Links to favorite websites -->
    <section>
        <h2>My Favorite Websites</h2>
        <ul>
            <li><a href="https://www.wikipedia.org" target="_blank">Wikipedia</a> - The free 
encyclopedia</li>
```

```
      <li><a href="https://www.github.com" target="_blank">GitHub</a> - Where developers collaborate</li>
      <li><a href="https://www.youtube.com" target="_blank">YouTube</a> - For video content</li>
      <li><a href="https://www.google.com" target="_blank">Google</a> - The Browser </li>
    </ul>
  </section>

  <hr>

  <!-- Weekly schedule table -->
  <section id="schedule">
    <h2>My Weekly Schedule</h2>
    <table border="5">
      <thead>
        <tr>
          <th>Time</th>
          <th>Monday</th>
          <th>Tuesday</th>
          <th>Wednesday</th>
          <th>Thursday</th>
          <th>Friday</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>9:00 - 11:00</td>
          <td>DS</td>
          <td>OS</td>
          <td>Android</td>
          <td>Web</td>
          <td>Team Project</td>
        </tr>
        <tr>
          <td>11:00 - 13:00</td>
          <td>Study Time</td>
          <td>Lunch Break</td>
          <td>Group Study</td>
          <td>Lunch Break</td>
          <td>Lab Work</td>
        </tr>
        <tr>
          <td>14:00 - 16:00</td>
          <td>Web</td>
          <td>Research</td>
          <td>Free Time</td>
          <td>Programming</td>
          <td>Study Time</td>
        </tr>
      </tbody>
    </table>
  </section>

  <hr>

  <!-- Semantic HTML demo -->
  <section>
    <h2>Semantic HTML Example</h2>
    <article>
      <h3>What is Semantic HTML?</h3>
      <p>Semantic HTML introduces meaning to the web page rather than just presentation. Examples include &lt;article&gt;, &lt;section&gt;, &lt;nav&gt;, &lt;header&gt;, &lt;footer&gt;, and more.</p>
    </article>
  </section>

  <footer>
    <p>&copy; 2025 My HTML Demo Page</p>
  </footer>
</body>
</html>
```

What is the purpose of the `<head>` section?

The `<head>` section contains metadata about the HTML document that isn't displayed on the page itself. It includes:

- Character encoding (`<meta charset="UTF-8">`)
- Viewport settings for responsive design
- The page title that appears in browser tabs
- Links to external resources (CSS stylesheets, JavaScript files)
- Favicon definitions
- SEO-related metadata
- Other information that helps browsers properly render the page

Why use semantic tags like `<section>`, `<article>`, and `<footer>`?

Semantic tags provide meaning to the structure of web content rather than just defining its appearance. Benefits include:

1. **Improved accessibility**: Screen readers and assistive technologies can better interpret the page structure
2. **Better SEO**: Search engines understand the content hierarchy and importance
3. **Code maintainability**: Makes HTML more readable and self-documenting
4. **Consistent structure**: Provides a standardized way to organize content
5. **Device compatibility**: Helps with responsive design and content adaptation across different devices

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>HTML Forms</title>
  <link rel="icon" href="edugamez-logo.png" type="image/x-icon">
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <!-- Navigation bar -->
  <div class="navbar-container">
    <nav class="navbar">
      <div class="logo">
        <img src="edugamez-logo.png" alt="EduGamez Logo" width="40" height="40">
        <span>Web Basics</span>
      </div>
      <div class="menu-toggle">
        <div class="bar"></div>
        <div class="bar"></div>
        <div class="bar"></div>
      </div>
      <ul class="nav-menu">
        <li><a href="index.html">Home</a></li>
        <li><a href="form.html" class="active">Form Demo</a></li>
        <li><a href="index.html#books">Books</a></li>
        <li><a href="index.html#hobbies">Hobbies</a></li>
        <li><a href="index.html#schedule">Schedule</a></li>
      </ul>
    </nav>
</div>

<h1>Fill The Forms</h1>

<form action="#" method="post">
    <fieldset>
        <legend>Personal Information</legend>

        <div>
            <label for="name">Name:</label>
            <input type="text" id="name" name="name" required>
        </div>

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

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

    <fieldset>
        <legend>Gender</legend>

        <div>
            <input type="radio" id="male" name="gender" value="male">
            <label for="male">Male</label>
        </div>
    <div>
      <input type="radio" id="female" name="gender" value="female">
      <label for="female">Female</label>
    </div>

    <div>
      <input type="radio" id="other" name="gender" value="other">
      <label for="other">Other</label>
    </div>
  </fieldset>

  <fieldset>
    <legend>Interests</legend>

    <div>
      <input type="checkbox" id="technology" name="interests" value="technology">
      <label for="technology">Technology</label>
    </div>

    <div>
      <input type="checkbox" id="sports" name="interests" value="sports">
      <label for="sports">Sports</label>
    </div>

    <div>
      <input type="checkbox" id="art" name="interests" value="art">
      <label for="art">Art</label>
    </div>

    <div>
      <input type="checkbox" id="music" name="interests" value="music">
```

```
        <label for="music">Music</label>
    </div>
</fieldset>

<fieldset>
    <legend>Country</legend>

    <div>
        <label for="country">Select your country:</label>
        <select id="country" name="country">
            <option value="">--Please choose a country--</option>
            <option value="usa">United States</option>
            <option value="canada">Canada</option>
            <option value="uk">United Kingdom</option>
            <option value="australia">Australia</option>
            <option value="germany">Germany</option>
            <option value="france">France</option>
            <option value="japan">Japan</option>
            <option value="Tanzania">Tanzania</option>
            <option value="other">Other</option>
        </select>
    </div>
</fieldset>

<fieldset>
    <legend>Profile Picture</legend>

    <div>
        <label for="profile-pic">Upload a profile picture:</label>
        <input type="file" id="profile-pic" name="profile-pic" accept="image/*">
    </div>
</fieldset>
```

```
        <div>
            <button type="submit">Submit</button>
        </div>
    </form>

    <footer>
        <p>&copy; 2025 HTML Forms Demo</p>
    </footer>
</body>
</html>
```

What's the difference between GET and POST?

GET Method:

- Data is appended to the URL as query parameters
- Data is visible in the URL (not secure for sensitive information)
- Limited amount of data can be sent (URL length restrictions)
- Requests can be bookmarked
- Good for retrieving data

POST Method:

- Data is sent in the request body, not visible in the URL
- More secure for sensitive information (like passwords)
- Can send larger amounts of data
- Requests cannot be bookmarked
- Good for submitting data that will modify the server's state

In our form, I used the POST method since we're collecting sensitive information like passwords.

What happens if required fields are empty upon submission?

When a form field is marked with the `required` attribute and is left empty upon submission:

1. The browser will prevent the form from being submitted
2. A validation message will appear near the empty required field
3. The form will not be processed until all required fields are filled
4. The focus will be automatically placed on the first empty required field

In our form, the Name, Email, and Password fields are marked as required, so users must fill these in before the form can be submitted.

Css style:

```
/* General styling */
body {
    font-family: 'Arial', sans-serif;
    background-color: #f5f5f5;
    color: #333;
    line-height: 1.6;
    max-width: 1000px;
    margin: 0 auto;
    padding: 0;
}

/* Navigation bar styling */
.navbar-container {
    width: 100%;
    background-color: #2c3e50;
    position: sticky;
    top: 0;
    z-index: 1000;
}

.navbar {
    display: flex;
    justify-content: space-between;
    align-items: center;
    padding: 0 20px;
    height: 70px;
    max-width: 1000px;
    margin: 0 auto;
}

.logo {
    display: flex;
    align-items: center;
}

.logo img {
    margin-right: 10px;
    box-shadow: none;
}

.logo span {
    color: white;
    font-weight: bold;
    font-size: 1.2rem;
}

.nav-menu {
    display: flex;
    list-style: none;
    margin: 0;
    padding: 0;
}

.nav-menu li {
    margin: 0 15px;
}

.nav-menu a {
    color: white;
    text-decoration: none;
    font-weight: 500;
    position: relative;
    padding: 5px 0;
}

.nav-menu a:hover {
    color: #3498db;
}

.nav-menu a.active {
    color: #3498db;
}

.nav-menu a.active::after {
    content: '';
    display: block;
    height: 2px;
    width: 100%;
    background-color: #3498db;
    position: absolute;
    bottom: 0;
    left: 0;
}

.menu-toggle {
    display: none;
    flex-direction: column;
    cursor: pointer;
}

.bar {
    width: 25px;
    height: 3px;
    background-color: white;
    margin: 3px 0;
    transition: 0.4s;
}

/* Content styling */
h1 {
    color: #2c3e50;
    text-align: center;
    margin: 30px 0;
    font-size: 2.5rem;
    padding: 0 20px;
}

h2 {
    color: #3498db;
    margin-top: 25px;
    margin-bottom: 15px;
    border-bottom: 2px solid #eaeaea;
    padding-bottom: 5px;
}

h3, h4, h5, h6 {
    color: #2c3e50;
    margin-top: 20px;
    margin-bottom: 10px;
}

section {
    background-color: white;
    padding: 20px;
    margin-bottom: 25px;
    border-radius: 8px;
    box-shadow: 0 2px 5px rgba(0,0,0,0.1);
}

/* Form styling */
form {
    background-color: white;
    padding: 25px;
    border-radius: 8px;
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
}

/* List styling */
ul, ol {
    padding-left: 25px;
    margin-bottom: 20px;
}

ul li, ol li {
    margin-bottom: 8px;
}

/* Image styling */
figure {
    margin: 20px 0;
    text-align: center;
}

img {
    max-width: 100%;
    height: auto;
    border-radius: 6px;
    box-shadow: 0 3px 10px rgba(0,0,0,0.2);
    transition: transform 0.3s ease;
}

img:hover {
    transform: scale(1.02);
}

figcaption {
    margin-top: 10px;
    color: #7f8c8d;
    font-style: italic;
}

/* Table styling */
table {
    width: 100%;
    border-collapse: collapse;
    margin: 20px 0;
    background-color: white;
}

th, td {
    border: 1px solid #ddd;
    padding: 12px;
    text-align: left;
}

th {
    background-color: #3498db;
    color: white;
    font-weight: bold;
}

tr:nth-child(even) {
    background-color: #f2f2f2;
}

tr:hover {
    background-color: #e3f2fd;
}

/* Link styling */
a {
    color: #3498db;
    text-decoration: none;
    transition: color 0.3s ease;
}
```

```
a:hover {
   color: #2980b9;
   text-decoration: underline;
}

/* Horizontal rule styling */
hr {
   border: 0;
   height: 1px;
   background-image: linear-gradient(to right, rgba(0, 0, 0, 0), rgba(0, 0, 0, 0.75), rgba(0, 0, 0, 0));
   margin: 25px 0;
}

fieldset {
   margin-bottom: 20px;
   border: 1px solid #ddd;
   border-radius: 5px;
   padding: 15px;
}

legend {
   color: #2980b9;
   font-weight: bold;
   padding: 0 10px;
}

/* Form controls */
div {
   margin-bottom: 12px;
}

label {
   display: inline-block;
   width: 180px;
   margin-bottom: 5px;
}

input[type="text"],
input[type="email"],
input[type="password"],
select {
   width: 100%;
   padding: 10px;
   border: 1px solid #ddd;
   border-radius: 4px;
   box-sizing: border-box;
   font-size: 16px;
}

input[type="radio"],
input[type="checkbox"] {
   margin-right: 10px;
}

/* Button styling */
button {
   background-color: #3498db;
   color: white;
   padding: 12px 20px;
   border: none;
   border-radius: 4px;
   cursor: pointer;
   font-size: 16px;
   width: 100%;
   transition: background-color 0.3s ease;
}

button:hover {
   background-color: #2980b9;
}

/* Radio buttons and checkboxes styling */
.gender-option,
.interest-option {
   display: flex;
   align-items: center;
   margin-bottom: 8px;
}

/* Footer styling */
footer {
   text-align: center;
   margin-top: 30px;
   color: #7f8c8d;
   font-size: 14px;
}

/* Custom class selectors */
.required-field::after {
   content: " *";
   color: red;
}

.form-control {
   margin-bottom: 15px;
}

/* Custom ID selector */
#profile-pic {
   border: 1px dashed #3498db;
   padding: 10px;
   background-color: #f8f9fa;
}
```

```
/* Responsive styling */
@media (max-width: 768px) {
    body {
        padding: 0;
    }

    .menu-toggle {
        display: flex;
    }

    .nav-menu {
        position: fixed;
        left: -100%;
        top: 70px;
        flex-direction: column;
        background-color: #2c3e50;
        width: 100%;
        text-align: center;
        transition: 0.3s;
        box-shadow: 0 10px 10px rgba(0, 0, 0, 0.1);
        padding: 10px 0;
    }

    .nav-menu.active {
        left: 0;
    }

    .nav-menu li {
        margin: 15px 0;
    }

    /* Make table responsive */
    table {
        display: block;
        overflow-x: auto;
        white-space: nowrap;
    }

    /* Adjust image size */
    img {
        max-width: 100%;
        height: auto;
    }

    /* Adjust section padding */
    section {
        padding: 15px;
        margin: 15px;
    }

    h1 {
        font-size: 2rem;
    }

    h2 {
        font-size: 1.5rem;
    }
}

@media (max-width: 600px) {
    label {
        width: 100%;
        display: block;
    }

    fieldset {
        padding: 10px;
    }

    h1 {
        font-size: 1.8rem;
    }

    .logo span {
        font-size: 1rem;
    }

    section {
        margin: 10px;
        padding: 10px;
    }
}
```

Inline CSS:

- Applied directly to HTML elements using the `style` attribute
- Example: `<p style="color: blue; font-size: 16px;">Text</p>`
- Pros: Immediately applies to a specific element
- Cons: Mixes content with presentation, difficult to maintain, needs to be repeated for similar elements

- **Internal CSS**:

- Defined within a `<style>` tag in the `<head>` section of an HTML document
- Example:

```
<head>
  <style>
    p { color: blue; font-size: 16px; }
  </style>
</head>
```

Pros: No need for external files, applies to all matching elements on the page

  - Cons: Only applies to the current page, increases page size

2. **External CSS** (what we've implemented):
    - Defined in a separate .css file and linked to HTML using `<link>` tag
    - Example: `<link rel="stylesheet" href="style.css">`
    - Pros: Separates content from presentation, can be cached by browsers, reusable across multiple pages
    - Cons: Additional HTTP request to load the CSS file

How conflicting styles are resolved (CSS Specificity):

CSS uses a specificity hierarchy to determine which style rules apply when there are conflicts:

1. **Priority order** (from highest to lowest):
    - !important declarations
    - Inline styles
    - ID selectors (#example)
    - Class selectors (.example), attribute selectors, and pseudo-classes
    - Element selectors (p, div, etc.) and pseudo-elements
2. **Specificity calculation**:
    - More specific selectors override less specific ones
    - When specificity is equal, the last declared rule wins (cascade)
3. **Example of specificity in action**:

```
p { color: blue; }             /* Specificity: 0,0,0,1 */
.text { color: red; }          /* Specificity: 0,0,1,0 */
#unique { color: green; }      /* Specificity: 0,1,0,0 */
p.text#unique { color: purple; } /* Specificity: 0,1,1,1 */
```

The element with ID "unique" and class "text" would be purple because the last selector is most specific.

1. What is responsive design?

Responsive design is an approach to web design that makes your web pages render well on all devices and screen sizes by automatically adapting the layout. Rather than creating separate websites for different devices, responsive design uses CSS techniques like fluid grids, flexible images, and media queries to adjust the design and content to fit the device's screen.

Key principles of responsive design include:

- Fluid layouts that use relative units like percentages instead of fixed pixels
- Media queries that apply different styles based on device characteristics
- Flexible images and media that scale appropriately
- Mobile-first approach that prioritizes designing for smaller screens first

2. Why is mobile-first a best practice?

Mobile-first is considered a best practice for several important reasons:

1. **Progressive Enhancement**: Starting with a basic design for mobile and then enhancing it for larger screens is more efficient than trying to shrink a complex desktop design.
2. **Performance Benefits**: Mobile users often have slower connections, so designing with mobile constraints in mind forces you to prioritize what's truly important. This creates faster, more efficient websites for all users.
3. **Growing Mobile Usage**: Mobile devices now account for more than half of all web traffic globally. Designing for these users first ensures they get the best possible experience.
4. **Better Focus on Content**: Mobile-first forces you to focus on essential content and functionality, leading to cleaner, more focused designs.
5. **Future-Proofing**: As new devices with different screen sizes emerge, mobile-first responsive designs are more adaptable to these changes.
6. **SEO Advantages**: Google uses mobile-first indexing, meaning it primarily looks at the mobile version of a site for ranking. Having a strong mobile experience improves your search ranking.

Why use Bootstrap?

Bootstrap offers several advantages:

- Responsive design out of the box
- Consistent UI components
- Comprehensive documentation
- Grid system for easy layouts
- Extensive pre-built components
- Community support
- Cross-browser compatibility
- Time-saving development

Difference between container, container-fluid, and row:

1. **Container**:
    - Has a fixed width that varies at different breakpoints
    - Centers content on the page with padding on both sides
2. **Container-fluid**:
    - Spans the entire width of the viewport (100% width)
    - Has padding on both sides
    - Useful when you want full-width content
3. **Row**:
    - Used inside containers to create horizontal groups of columns
    - Has negative margins to offset the padding of containers
    - Ensures proper alignment of columns

- Rows must be placed inside containers, and columns must be placed inside rows

DOM vs. HTML: The HTML file is the static markup of a webpage. The DOM (Document Object Model) is a live, in-memory representation of the HTML structure as objects that JavaScript can interact with and manipulate.

- **DevTools Node Changes:** When you add or delete nodes in the browser's DevTools:

- **Addition:** The browser instantly renders the new element on the page.
- **Deletion:** The browser instantly removes the element from the rendered page. These changes are **temporary and only reflected in your current browser view**. They do not modify the original HTML file on the server. Refreshing the page will reload the HTML file and revert to its original state.