Here’s a **detailed class script** for teaching Week 2: **Semantic HTML and Accessibility**:

**Class Title: Semantic HTML and Accessibility**

**Duration:** 1 hour 30 minutes  
**Objective:** Teach the importance of semantic HTML and how to create accessible and structured web pages.

**Class Plan**

**1. Recap of Week 1 (10 mins)**

**Script:**  
“Welcome back, everyone! Before we dive into today’s session, let’s quickly recap what we learned last week. Can anyone remind me what an HTML document structure looks like? What are some common tags we used last week?”

* Review the following points:
  + Basic HTML document structure (<!DOCTYPE html>, <html>, <head>, <body>).
  + Tags like <h1>, <p>, <a>, <div>.
* Ask students for examples of what they created last week.

**Transition:**  
“Great! Now that we’ve refreshed our memory, let’s dive into today’s topic: Semantic HTML and Accessibility.”

**2. Introduction to Semantic HTML (20 mins)**

**Script:**  
“Semantic HTML is all about using tags that have meaningful names to describe their content. For example, instead of using a generic <div> for every section, you can use <header> for the top part, <main> for the main content, and <footer> for the bottom. Why is this important? It helps search engines understand your site better and makes it easier for screen readers to navigate.”

* **Key Points to Cover:**
  + Difference between semantic and non-semantic tags: <div> vs <header>, <section>, <article>.
  + How semantic tags improve SEO and accessibility.
  + Example: Building a basic blog page structure.
* **Live Demo:**  
  “Let’s create a simple blog layout using semantic tags.”
* <!DOCTYPE html>
* <html lang="en">
* <head>
* <meta charset="UTF-8">
* <meta name="viewport" content="width=device-width, initial-scale=1.0">
* <title>My Blog</title>
* </head>
* <body>
* <header>
* <h1>Welcome to My Blog</h1>
* <nav>
* <a href="#home">Home</a>
* <a href="#about">About</a>
* <a href="#contact">Contact</a>
* </nav>
* </header>
* <main>
* <article>
* <h2>First Blog Post</h2>
* <p>This is my first blog post. I hope you enjoy it!</p>
* </article>
* <section>
* <h3>Related Posts</h3>
* <ul>
* <li><a href="#">Post 1</a></li>
* <li><a href="#">Post 2</a></li>
* </ul>
* </section>
* </main>
* <footer>
* <p>&copy; 2025 My Blog</p>
* </footer>
* </body>
* </html>
* Ask students to follow along and build this structure.

**Here’s how you can explain these key points during your class:**

**1. Difference Between Semantic and Non-Semantic Tags**

**What to Say:**

“Let’s start by understanding the difference between semantic and non-semantic tags.

* **Non-Semantic Tags:** These are tags that don’t convey any meaning about their content. The most common example is the <div> tag. It’s basically a generic container used to group content together. It doesn’t tell us—or the browser—what the content inside it is about.

For example:

<div>

<h1>About Us</h1>

<p>We are a company focused on innovation.</p>

</div>

This works, but the browser and search engines have no idea that this content is an 'About Us' section.

* **Semantic Tags:** These tags *do* have meaning and tell us (and search engines) what the content represents. For example, <header> is used for the top section of a page, <section> is for grouping related content, and <article> is for a standalone piece of content like a blog post or news article.

Here’s how we can rewrite the same example with semantic tags:

<section>

<h1>About Us</h1>

<p>We are a company focused on innovation.</p>

</section>

Now the browser, search engines, and even screen readers know that this is a section of related content.”

**Interactive Question:**  
“Can anyone tell me what tag they might use to represent the navigation menu? (Answer: <nav>)

What about a footer at the bottom of the page? (Answer: <footer>)”

**2. How Semantic Tags Improve SEO and Accessibility**

**What to Say:**

“Semantic tags have two huge benefits:

1. **SEO (Search Engine Optimization):**  
   Search engines like Google use your HTML to understand the structure of your webpage. If you use semantic tags, you make it easier for search engines to figure out what’s important. This can improve your site’s ranking in search results.

For example:

* A <header> tag tells Google this section contains the page’s main heading or introductory info.
* <article> tells search engines, ‘This is a self-contained piece of content.’

If you only use <div> everywhere, search engines have to guess, and that can hurt your SEO.

1. **Accessibility:**  
   Semantic tags make your site more accessible to people using assistive technologies, like screen readers.

For instance:

* A screen reader will announce a <nav> tag as ‘navigation,’ so visually impaired users know they’re in the site’s menu.
* Similarly, a <footer> will be announced as ‘footer,’ making it clear this is the bottom section of the page.

Non-semantic tags like <div> don’t give this context, which can make it harder for users with disabilities to navigate your site.”

**Demo:**  
“Let’s see a quick example. Imagine we’re building a blog page:

Non-Semantic Version:

<div>

<h1>My Blog</h1>

</div>

<div>

<h2>Latest Posts</h2>

</div>

Semantic Version:

<header>

<h1>My Blog</h1>

</header>

<section>

<h2>Latest Posts</h2>

</section>

See how much clearer the second version is? It’s more organized, better for SEO, and accessible to everyone.”

**Wrap-Up Question for Students:**  
“Why do you think it’s a bad idea to use <div> everywhere instead of semantic tags? What benefits do semantic tags provide?”

**3. Working with Forms (30 mins)**

**Script:**  
“Forms are a crucial part of any website. They allow users to interact with your site, whether they’re signing up, sending a message, or making a purchase. Let’s look at how to create an accessible and semantic form.”

* **Key Points to Cover:**
  + Tags: <form>, <label>, <input>, <textarea>, <select>, <button>.
  + Attributes: action, method, placeholder, required, type.
  + Importance of <label> for accessibility.

**Here’s a detailed script you can follow to teach the key points:**

**Introduction to Forms**

* **Start by explaining why forms are important:  
  "Forms are a vital part of web development because they allow users to interact with a website, such as submitting information, making searches, or logging in. They are the main way data flows from users to the server."**

**Tags Overview**

1. **<form>** 
   * **What it does: The container for all form elements. It defines where and how the data will be submitted.**
   * **Example to show:**

**<form action="/submit" method="post">**

**<!-- Form elements go here -->**

**</form>**

* + **Explain the action and method attributes:** 
    - **action: Specifies the URL where the form data should be sent.  
      Example: "submit.php" or "https://example.com/signup".**
    - **method: Defines how the data is sent (e.g., GET for URLs, POST for secure data submission).**

1. **<label>** 
   * **Why it’s important:  
     "Labels improve accessibility by helping screen readers identify input fields. Clicking a label also focuses its associated input field, which is great for usability."**
   * **How to use:**

**<label for="email">Email:</label>**

**<input type="email" id="email" name="email">**

* + - **for attribute connects the label to the id of the input field.**

1. **<input>** 
   * **What it does: Defines various input fields like text, email, passwords, etc.**
   * **Attributes to highlight:** 
     + **type: Specifies the type of input (e.g., text, email, password).  
       Example:**

**<input type="text" placeholder="Enter your name">**

* + - **placeholder: Displays a hint or example inside the input field.**
    - **required: Ensures the field must be filled before the form is submitted.**
  + **Interactive Idea: Ask students to add multiple input fields (e.g., name, email, and password).**

1. **<textarea>** 
   * **What it does: Provides a multi-line text box for longer inputs.**
   * **Example to show:**

**<label for="message">Message:</label>**

**<textarea id="message" name="message" placeholder="Write your message here..." required></textarea>**

1. **<select>** 
   * **What it does: Creates a dropdown menu for selecting options.**
   * **Example to show:**

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

**<select id="gender" name="gender">**

**<option value="male">Male</option>**

**<option value="female">Female</option>**

**<option value="other">Other</option>**

**</select>**

* + **Explain that each <option> represents one choice.**

1. **<button>** 
   * **What it does: Used to submit a form or trigger actions.**
   * **Example to show:**

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

* + **Explain that the type attribute can be submit, button, or reset.**

**Interactive Activity**

* **Task for students:  
  "Create a basic sign-up form with the following: name, email, password, and a message box. Include labels, placeholders, and at least one required field."**

**Wrap-Up on Accessibility**

* **"Always use <label> for input fields. This improves accessibility, especially for users with disabilities who rely on screen readers."**
* **Emphasize the use of placeholder and required attributes to guide and validate user input.**

**This structure should help you deliver a clear and engaging class on forms!**

**Live Demo:**  
“Let’s create a contact form with proper structure and accessibility features.”

<form action="/submit" method="POST">

<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>

<label for="message">Message:</label>

<textarea id="message" name="message" required></textarea>

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

</form>

* Emphasize the role of for attribute in <label> for linking it to the corresponding input field.
* Discuss basic validation using attributes like required, type, and pattern.

**Hands-On Task:**  
Ask students to add fields for phone numbers and dropdowns for selecting topics of interest.

**4. Accessibility Basics (20 mins)**

**Script:**  
“Web accessibility is about making sure everyone can use your website, including people with disabilities. This is not just good practice but also a requirement in many countries. Let’s look at some ways to make your site more accessible.”

* **Key Points to Cover:**
  + Adding alt attributes to <img> tags for screen readers.
  + Using ARIA roles (role="button", aria-label, aria-hidden) for dynamic elements.
  + Ensuring proper keyboard navigation (e.g., using tabindex).

**Here’s a structured script you can follow to teach these key accessibility points:**

**Introduction to Web Accessibility**

* **Start by emphasizing the importance of accessibility:  
  "Web accessibility ensures that everyone, including people with disabilities, can use and interact with your website. It’s not just about compliance—it’s about creating inclusive experiences."**

**1. Adding alt Attributes to <img> Tags**

* **Why it’s important:  
  "The alt attribute provides alternative text for images, which screen readers use to describe the image to visually impaired users. It also helps if the image fails to load."**
* **How to write good alt text:  
  "Describe the purpose of the image, not just what it looks like. For example, if it’s a decorative image, you can leave the alt attribute empty (alt="")."**
* **Example to show:**

**<img src="profile.jpg" alt="Profile picture of John Doe">**

* **Interactive Idea: Show examples of good vs. bad alt text, like:** 
  + **Bad: alt="picture"**
  + **Good: alt="A young woman smiling and holding a laptop"**

**2. Using ARIA Roles**

* **What is ARIA?  
  "ARIA (Accessible Rich Internet Applications) is a set of attributes you can add to HTML to make dynamic and interactive elements more accessible to screen readers."**
* **Common ARIA roles and attributes:**
  + **role="button":  
    "Use this when an element looks or behaves like a button but isn’t a <button>."  
    Example:**

**<div role="button" tabindex="0">Click Me</div>**

* + **aria-label:  
    "Provides a label for an element when there’s no visible text."  
    Example:**

**<button aria-label="Close menu">X</button>**

* + **aria-hidden="true":  
    "Hides elements from screen readers, like icons that are purely decorative."  
    Example:**
  + **<i class="icon" aria-hidden="true"></i>**
* **Best Practice:  
  "Use ARIA only when necessary. For example, if you can use a semantic tag like <button>, there’s no need for role="button"."**

**3. Ensuring Proper Keyboard Navigation**

* **Why keyboard navigation matters:  
  "Some users rely solely on keyboards or assistive technologies to navigate websites. Ensuring proper navigation makes your site usable for everyone."**
* **Using tabindex:  
  "The tabindex attribute controls the order in which elements receive focus when users press the Tab key."**
  + **tabindex="0": Adds an element to the normal tab order.**
  + **tabindex="-1": Removes an element from the tab order.**
  + **Example:**
  + **<div tabindex="0">Focusable Div</div>**
* **Interactive Tip:  
  "Ask students to try tabbing through a form or webpage. Then, show how tabindex can fix focus issues."**

**Interactive Activity**

* **"Let’s create an accessible navigation bar together. It should include:** 
  + **Images with meaningful alt attributes.**
  + **A dynamic menu icon with role="button" and aria-label.**
  + **Proper keyboard navigation using tabindex."**

**Conclusion**

* **Wrap up by reinforcing the importance of accessibility:  
  "Accessibility isn’t optional—it’s a fundamental part of web development. The more inclusive we make our websites, the better experience we create for everyone."**

**This script should help you deliver a detailed and practical class on accessibility!**

**Demo:**  
“Here’s how we can make our contact form more accessible.”

<form action="/submit" method="POST" aria-labelledby="contactForm">

    <h2 id="contactForm">Contact Us</h2>

    <label for="name">Name:</label>

    <input type="text" id="name" name="name" required aria-required="true">

    <label for="email">Email:</label>

    <input type="email" id="email" name="email" required aria-required="true">

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

  </form>

**Activity:**  
Ask students to test their form using only the keyboard (tab navigation).

**5. Q&A and Assignment (10 mins)**

**Q&A:**

* Open the floor for questions and clarifications.
* Encourage students to share their thoughts on semantic HTML and accessibility.

**Assignment:**  
“Your homework is to create a personal portfolio page using semantic HTML and an accessible contact form. Be creative! Make sure your form has at least three fields and is fully keyboard-navigable.”

**Final Notes**

* **Encouragement:** Emphasize the long-term benefits of good semantic and accessible practices.
* **Tools Recommendation:** Suggest browser extensions like Lighthouse for testing accessibility.