



Week 6: Adding Text and Structure

4010 WEBPAGE DESIGN

Learning Objectives

By the end of this session students should be able to;

- Gain an understanding of the basic XHTML structure
- Gain an understanding of some of the basic that you need to know about text.
- Learn how to structure content
- Gain an understanding of XHTML lists
- Learn how to created number and bulleted lists
- Learn how to format text
- Learning how to copy and existing text into Dreamweaver
- Learn how to import an entire Word document into an HTML page

Understanding Web Page Structure with HTML

- The way your web page is built determines what kind of content it shows and how that content appears in a browser.
- Behind the scenes, there's a special kind of code called **XHTML** that sets up this structure. Think of it like the blueprint for a house—it tells the browser where things go and how they should look.
- Let's explore some basic parts of XHTML and learn simple ways to use them effectively.

Basic HTML Structure Elements (1/2)

1. `<!DOCTYPE html>` Declaration

- Declares the document type.
- Tells the browser you're using HTML5.

2. `<html>`

- The root element of the page.
- Wraps all the content.

3. `<head>` Section

- Contains metadata (info about the page).
- Includes:
 - `<title>`: Sets the page title (shown in browser tab).
 - `<meta>`: Describes page encoding, author, etc.
 - `<link>`: Connects to external stylesheets.
 - `<style>`: Adds internal CSS.
 - `<script>`: Adds JavaScript.

Basic XHTML Structure Elements (2/2)

4. <body> Section

- Holds everything visible on the page.
- Includes:
 - Headings (<h1> to <h6>)
 - Paragraphs (<p>)
 - Images ()
 - Links (<a>)
 - Lists (, ,)
 - Tables (<table>, <tr>, <td>)
 - Forms (<form>, <input>, <textarea>)

Tips for Using XHTML Effectively

- Always **close your tags**: XHTML is stricter than HTML. Example: `</br>` instead of `
`
- Use lowercase for all tags.
- Nest elements properly—don't leave tags hanging.
- Include the **alt** attribute for images to improve accessibility.

Do Exercise Demonstrating HTML Elements on
the handout given

CASCADING STYLE SHEET (CSS)

WEEK 5

Overview

- A **Cascading Style Sheet (CSS)** is a stylesheet language used to control the presentation and layout of HTML elements on a webpage. It allows developers to separate content (HTML) from design (CSS), making websites easier to maintain and more visually appealing.
- **What CSS Does**
 - Controls colors, fonts, and spacing
 - Defines layout (grid, flexbox, positioning)
 - Adds animations and transitions
 - Enables responsive design for different screen sizes

Example CSS

```
body {  
  background-color: #f0f0f0;  
  font-family: Arial, sans-serif;  
  margin: 0;  
  padding: 0;  
}  
  
h1 {  
  color: navy;  
  text-align: center;  
}  
  
p {  
  font-size: 16px;  
  line-height: 1.5;  
}
```

Understanding CSS (1/3)

- A **style sheet** is a collection of these styles that can be applied to a single page in your site, a range of pages or the entire site. CSS lets you control many properties that cannot be controlled with HTML alone, such as specifying font sizes.
- CSS styles work like styles in programs like Word—you define a style once for elements like headings, and any changes to that style automatically update all matching elements across the page.
- CSS lets you manage the design of your whole website from a single file. By linking all pages to that file, you can easily update the site's appearance in one place.

Understanding CSS (2/3)

CSS style sheets are files or blocks of code that define how HTML elements should look. They can be written in three main places that styles can be placed:

1. External Style Sheet

- Stored in a separate .css file and then linked to HTML using `<link>` tag
- Best for large projects and consistent styling across multiple pages.

Example: `<link rel="stylesheet" href="styles.css">`

```
/* styles.css */
```

```
body {  
    background-color: #fff;  
    font-family: 'Segoe UI', sans-serif;  
}
```

Understanding CSS (3/3)

2. Internal Style Sheet

- Written inside a `<style>` tag in the `<head>` of your HTML
- Good for single-page styling.

```
<head>
  <style>
    h1 {
      color: teal;
    }
  </style>
</head>
```

3. Inline Styles

Applied directly to an HTML element using the `style` attribute

Example: `<p style="color: blue; font-size: 16px;">Hello World</p>`

Anatomy of a CSS Rule

- A CSS rule is made up of two main parts:

1. the **selector**
2. and the **declaration block**.

- Here's a breakdown of its anatomy:

```
selector {  
  property: value;  
}
```

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

1. Selector

Identifies which HTML elements the rule applies to.

Examples: p targets all <p> tags

.className targets elements with a specific class

#idName targets an element with a specific ID.

2. Declaration Block

Contains one or more declarations, enclosed in {}.

Each declaration includes:

Property: The style attribute you want to change (e.g., color, font-size).

Value: The setting for that property (e.g., blue, 16px).

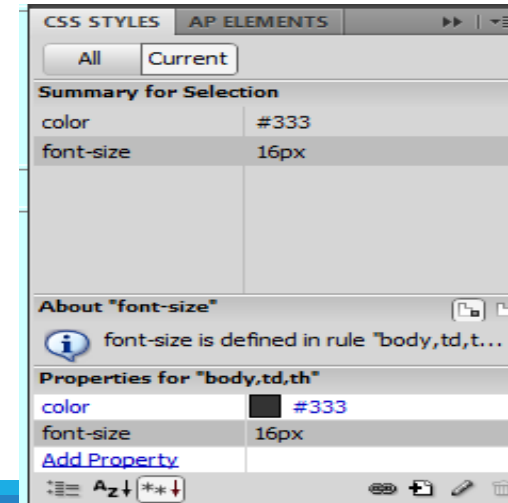
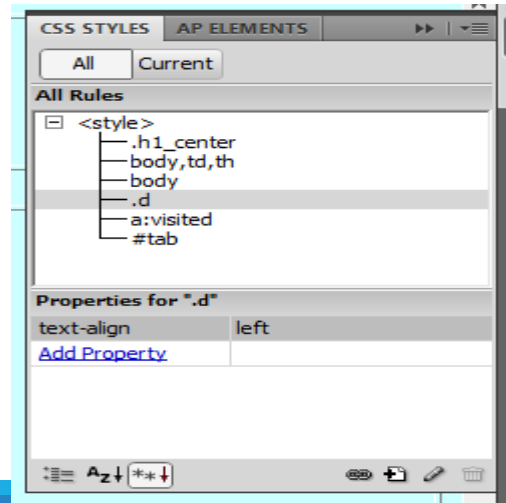
Understanding Selector Types

- A **selector** tells CSS **which HTML element** you want to style.
- **Common Selector Types**

Selector Type	What It Targets	Example CSS Rule	What It Does
Universal (*)	All elements	<code>* { margin: 0; }</code>	Removes margin from everything
Type (tag)	Specific HTML tags	<code>p { color: blue; }</code>	Makes all <code><p></code> text blue
Class (.name)	Elements with a class	<code>.box { border: 1px solid; }</code>	Styles anything with class “box”
ID (#name)	One unique element by ID	<code>#header { font-size: 24px; }</code>	Styles the element with ID “header”
Group (a, p)	Multiple elements together	<code>h1, h2 { color: red; }</code>	Makes both <code><h1></code> and <code><h2></code> red
Descendant	Elements inside another	<code>div p { color: green; }</code>	Styles <code><p></code> inside <code><div></code> only
Child (>)	Direct children only	<code>ul > li { list-style: none; }</code>	Styles only direct <code></code> in <code></code>
Sibling (+)	Next element after another	<code>h1 + p { margin-top: 0; }</code>	Styles <code><p></code> right after <code><h1></code>
Attribute	Elements with certain attributes	<code>input[type="text"] { }</code>	Styles text input fields only

The CSS Style Panel

- The CSS Styles Panel in Dreamweaver is your main control center for working with CSS. It gives you a complete overview of all the styles used on the page through the **All** tab, and lets you zoom in on the specific styles applied to the selected element using the **Current** tab.
- You can also use this panel to make new style rules, change or delete existing ones, and link external CSS files to your page.

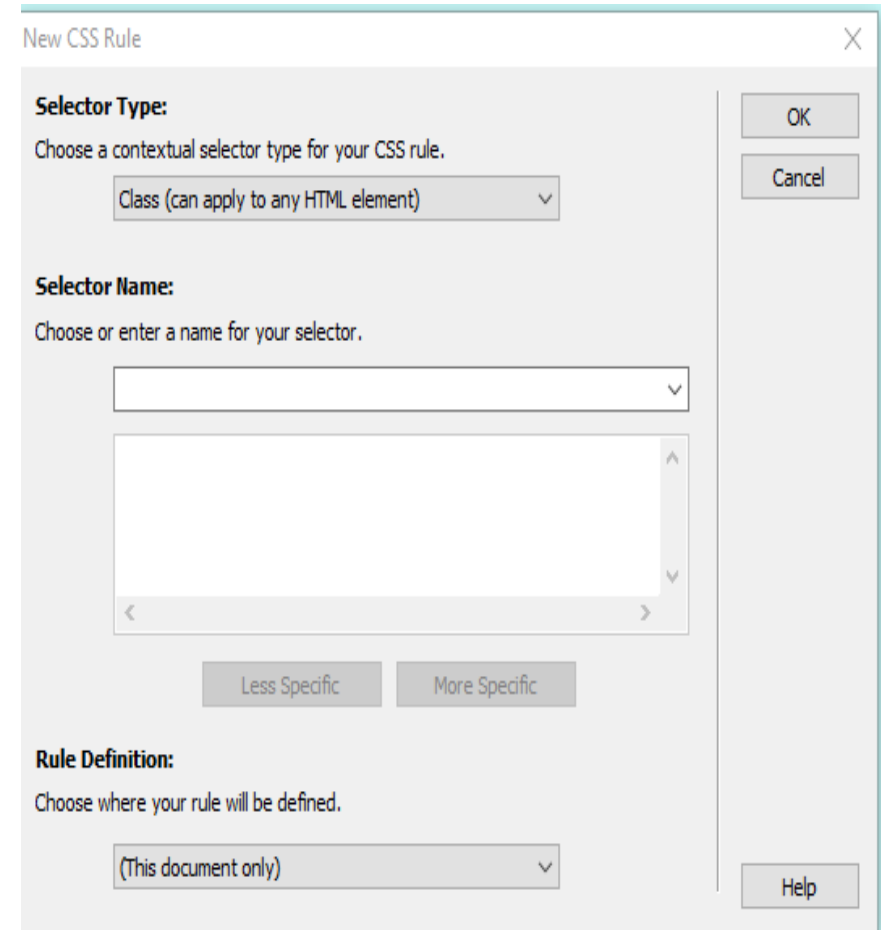
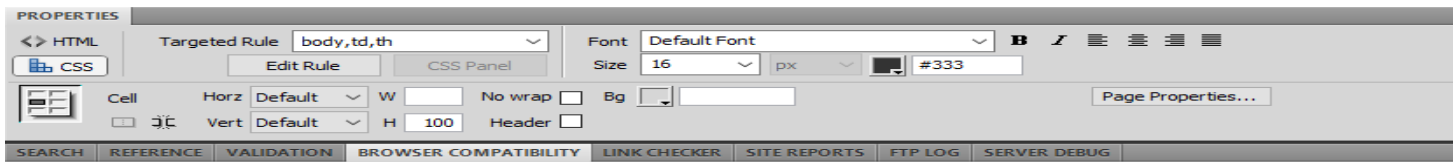


The New CSS Rule Dialogue Box

- No matter how you create a CSS rule in Dreamweaver, the **New CSS Rule dialog box** will appear. For a quick class style on selected text, you might use the **Properties Inspector**. But for more complex rules—like styling images in a sidebar—the **CSS Styles Panel** is the better choice.

Components of the New CSS Rule Dialog Box in Dreamweaver

1. **Selector Type:** Choose whether the rule targets a tag (e.g., p), class (e.g., .highlight), ID (e.g., #sidebar), or compound selector.
2. **Selector Name:** Enter the name of the tag, class, or ID you want to style.
3. **Rule Definition Location:** Decide where to save the rule—either in an external style sheet or embedded in the current HTML file.
4. **CSS Properties Editor:** After creating the rule, this section lets you define styles like font, color, margin, padding, borders, etc.



How to Create CSS Rules with the CSS Styles Panel

1. Open the Panel

- Go to Window > CSS Styles to open the CSS Styles Panel.

2. Click “New CSS Rule”

- At the bottom of the panel, click the + icon to start a new rule.

3. Use the New CSS Rule Dialog Box

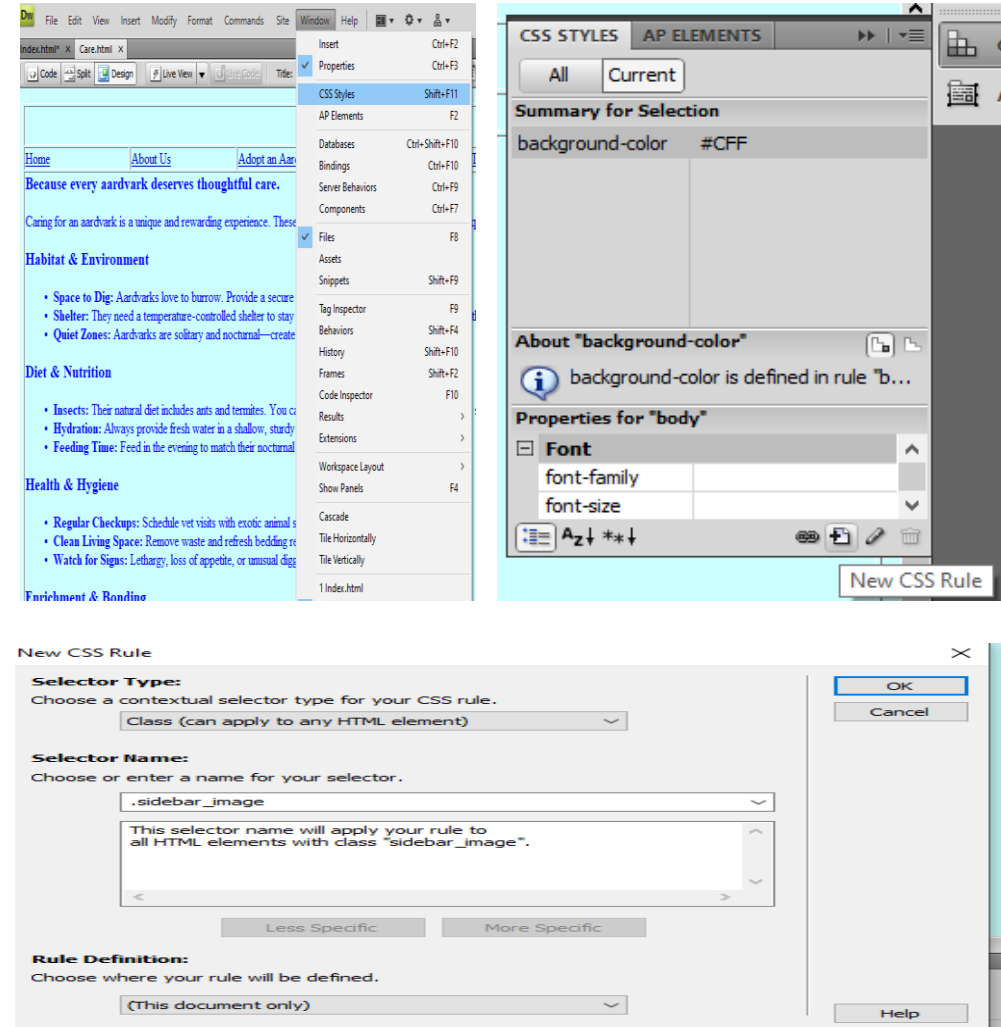
- Choose a selector type:** Tag, Class, ID, or Compound.
- Name your selector:** For example, .sidebar-image.
- Choose where to save the rule:** External stylesheet or embedded in the page.
- Click **OK** to continue.

4. Define Style Properties

- In the next window, set your styles:
 - Font, color, background
 - Margins, padding, borders
 - Layout and positioning

5. Apply the Rule to HTML

- Add the class or ID to your HTML element:



Why Use the CSS Styles Panel?

- Easy to create and manage styles visually
- Keeps your CSS organized
- Great for both simple and advanced styling

Creating Rules with the Properties Inspector (1/2)

- The **Properties Inspector** is split into *HTML* and *CSS* modes. Use CSS mode for styling and HTML mode for structural tweaks. You can switch between them depending on your task.
1. **Open the Properties Inspector**
 - Make sure the **Properties Inspector** is visible by going to **Window > Properties**.
 - Switch to the CSS mode by clicking the "CSS" button if you're not already in it.
 2. **Select the Element You Want to Style**
 - Click on any text or HTML element in Design View.
 - The **Properties Inspector** will show the current styling and allow you to modify it.
 3. **Create or Edit a CSS Rule**
 - Use the "Targeted Rule" dropdown to select an existing rule or choose "New CSS Rule."
 - Click "**Edit Rule**" to open the CSS Rule Definition dialog box.
 - Define properties like *font*, *color*, *margins*, and *alignment*.
 4. **Apply Styles and Preview**
 - After defining your rule, apply it to the selected element.
 - Use Live View or Preview in Browser to see the changes in action.

Creating Rules with the Properties Inspector (2/2)

5. Advanced Styling with External Style sheets

- You can link external style sheets and manage rules globally.
- This is ideal for consistent styling across multiple pages.

6. Styling Specific Elements Like Images

- You can use the Properties Inspector to assign classes and style images.
- For example, center an image or adjust its margins using CSS rules.

Thank you 😊