

# Using Style Sheets for Consistency

---

Cascading Style Sheets enable you to easily maintain a consistent look across all the pages of a web site. In addition, they extend the power of HTML. For example, style sheets permit specifying point sizes for text, “layering” of objects, full-justification of text, pull-quotes, inverse text, boxes and borders, creating columns (without tables) and much, much more. Styles can be used to redefine basic HTML tags by adding characteristics to them or you can create your own styles. If you have used styles in word processing programs, then you should have a general idea of how Cascading Style Sheets (CSS) work.

One caveat in using style sheets is that not all CSS works with old browsers. Be sure to preview your site in multiple browsers and on multiple platforms (which every good web designer does anyway.)

## Cascading Styles

Styles can be applied in one or all of three ways:

- 1) *Inline* styles are used with an individual tag in the body of a document.
- 2) *Internal* style sheets, usually defined in the header of the document, provide rules that apply to the entire HTML file, but not to other files in the same web site.
- 3) *External* style sheets are text files (usually with a *.css* extension) that can be applied to any document on the web site. Creating a site in which numerous pages access external style sheets facilitates making changes to the entire site quickly. When changes are made to the external style sheets, the changes are immediately seen the next time the pages using those styles are loaded.

### What is “Cascading”?

Documents may include an inline style, internal style sheets, and external style sheets. Each may attempt to redefine an HTML tag or apply a style to object on the page. When multiple styles are used for one object, the style settings will display the characteristics of each of the styles as long as they do not conflict.

So, what happens when they do conflict? There is a “priority level” which determines which styles have the greatest level of specificity. Basically, the style “closest” to the object is applied first and then the level or priority goes to the styles that are the “next closest”. This is where the term “*cascading*” comes from.

<u>Level of precedence</u>	<u>type</u>
1) Highest:	inline style (if there are two inline styles, the one closest gets priority.)
2) Second:	internal style sheet
3) Third:	external style sheet

Another factor influencing specificity is the type of “selector” used. These will be discussed a little later on, but basically “ID” selectors take precedence over “Class” selectors.

### What is Inheritance?

“Inheritance” allows CSS rules to be applied to a large portion of the page and other CSS rules to apply to a “nested” portion of the page. For example, if a section of text that has a style applied making it red, 36 point, and bold contains a section of text that is simply defined to be 24 point, then this nested text will inherit the additional characteristics and appear red, 24 point, and bold.

## Using CSS for Page Layout

When laying out a page, CSS provides an incredible amount of flexibility with a significant decrease in the amount of code needed compared to using HTML Tables for design. The best way to visualize this is to think of a series of nested “boxes” whose height and width define areas of the page. Because of inheritance, each of these nested boxes has the characteristics of their respective containers.

In traditional HTML layout, the browser reads and draws the page from top to bottom. In a sense with CSS, it still reads from top to bottom, but because the boxes can be nested and given specific positioning characteristics, something at the bottom of the code listing could appear at the top of the browser window. These boxes are best defined by using the **DIV** tag.

# Using Style Sheets for Consistency

## The Powerful <div> Tag

The DIV tag is the most powerful item in your CSS Toolbox. It contains no formatting by itself (other than a line break before and after), but is used to identify logical divisions on the page. These might be structural divisions like banner, content, and footer, or they might be applied to text groupings. The power comes when the *ID* or the *Class* attribute is added to a DIV. When this occurs, you can apply CSS rules to entire blocks of code at the same time. IDs or classes applied to DIVs that are used for page layout rely heavily on the *Box* category in the *CSS Rule Definition* window

### Box Category

- Width:** The amount of horizontal space taken up by the “box” and its contents.
- Padding:** Defines “white space” between the boundaries of the box and its contents.
- Margins:** Defines “white space” outside the boundaries of the box and indicates the distance between it and other objects on the page.
- Float:** Used to make contents of a containing block “wrap” around an object.
- Clear:** Used to prevent wrapping.
- Height:** Listed last to discourage its use because if the block contents do not take up the entire vertical space specified, the remainder as defined will be empty.

The most important *Box* attributes are *Width*, *Padding*, and *Margins*. They are required for page layout.

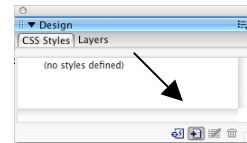
### When Should *ID* Be Used and When Should *Class* Be Used?

The *ID* can be used only once per page, while the *Class* can be used as many times as needed. So the ID should be used to define structural parts of the page that occur only once like the banner or the footer. The opening tag for an *ID* would look like this: `<div id= “footer”>` and for *Class* would look like: `<div id= “subhead”>`

## Creating Style Sheets in Dreamweaver

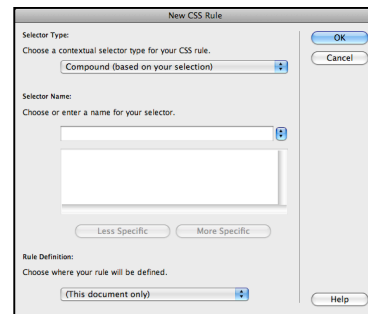
### Creating a new Style Rule

- 1 Pull down *Window to CSS Styles*.
- 2 Click the *New CSS Rule* icon (plus sign) in the lower right.



### Create internal styles

- 1 Select *This document only* in the *Define In* section.
- 2 Indicate a *Selector Type*, enter a name (for the style being defined) in the *Name Selector* field, and click **OK**.



### Create external styles

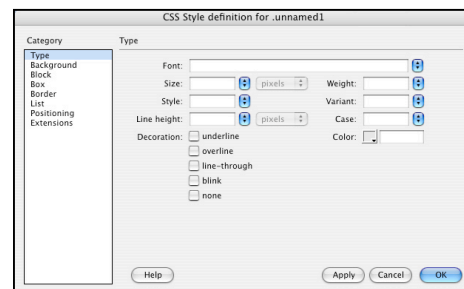
- 1 Select the desired file in the *Rules Definition* section.
- 2 Indicate a *Selector Type*, enter a name for the style in the *Name Selector* field, and click **OK**.

### Create an style in new external style sheet

- 1 Select *New Style Sheet File* in the *Rules Definition* section.
- 2 Indicate a *Selector Type*, enter a name for the style being defined in the *Name* field, and click **OK**.
- 3 Enter an appropriate filename with an extension of *.css* and save in the same folder as your web pages.

### Make Custom Style (class)

- 1 Select *Class* in the *Selector Type* section.
- 2 Select desired choice in *Rules Definition* section.
- 3 Enter a name for the custom style and click **OK**.
- 4 When the *CSS Style Definition* window appears, enter the desired characteristics for the categories you wish to define and click **OK**.



# Using Style Sheets for Consistency

---

## Defining a DIV

- 1 Select the objects to be included in the DIV.
- 2 Pull down *Insert to Layout Objects to Div Tag*.
- 3 In the *Insert Div Tag* window, enter a descriptive name in *ID* or *Class* and click **New CSS Style**.
- 4 In the *New CSS Rule* window, select the characteristics desired for the DIV and click **OK**.

## Absolute Positioning

A fixed (absolute) x,y location for an object can be specified on the screen with using absolute positioning. This style characteristic can be specified by the *Type* attribute in the *Positioning* category or by creating a special kind of DIV called an AP DIV. In the absence of anything else, the values entered in the *Top* and *Left* attributes in the *Positioning* category will affix the upper, left corner of this object to the specified screen location. If the user scrolls the browser, this object will remain in its absolute position relative to the screen and the rest of the contents will appear to move underneath it.

Nested DIVs provide greater control. If the DIV containing the absolutely positioned object is itself contained inside another DIV and if that container DIV has the *Positioning* attribute of *Type* set to *relative*, then the *Top* and *Left* absolute settings will be positioned “relative” to that container DIV’s boundaries rather than the screen.

## Defining an AP DIV to absolutely position content

(In earlier Dreamweaver versions, AP DIV was called a layer)

- 1 Pull down *Insert to Layout Objects to AP Div*. An *ID* named *#apDiv1* appears in the *CSS* window.
- 2 Click *#apDiv1* in the status bar at the bottom of the Dreamweaver window. In the upper left of the Properties window, change the name from *apDiv1* to a more descriptive name.
- 3 In the *design* window, click inside the new AP div box and insert or type the object to be positioned.
- 4 Adjust the size of the AP div box as needed.
- 5 Click the ID symbol (#) in the upper left of the AP div box and the drag to the desired location.

(**Note:** An important attribute when using absolute position is the *z-index*. This allows multiple items to be stacked in an order based on their value (highest number is “on top”>)

## Redefine HTML Tag

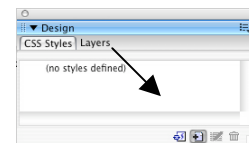
- 1 Select *Tag* in the *Selector Type* section.
- 2 Click the arrow on the right of the *Rules Definition* section.
- 3 Drag to the HTML tag to be redefined in the *Tag* pop-up menu, and click **OK**.
- 4 When the *CSS Style Definition* window appears, add the desired characteristics and click **OK**.

## Define Pseudo Selectors to “style” links

- 1 Select *Compound* in the *Selector Type* section.
- 2 Click the arrow on the right of the *Rules Definition* section. (Four link-related selectors are built into Dreamweaver: *a:active a:hover a:link a:visited*)
- 3 Drag to the desired selector in the *Selector* pop-up menu and then enter the class name before the “a” in the link selector, followed by a space. Click **OK**.
- 4 When the *CSS Style Definition* window appears, enter the characteristics desired and click **OK**.

## Use an existing (external) style sheet

- 1 Click *Attach Style Sheet* icon in the lower right of the *CSS Styles* window.
- 2 Select *Link* (to attach to an external) or *Import* (to add sheet to document.)
- 3 Browse to find style sheet (with *.css* extension), highlight it, and click **OK**.
- 4 Select media type. (**Note:** Multiple style sheets can be attached.)



## Converting internal styles to external style sheet

- 1 In the *CSS* window, highlight all of the rules in the list and then click the “down arrow” (on the right of the *CSS* window’s title bar) and select **Move CSS Rules**.
- 2 In the *Move to External Style Sheet* window, click **A new style sheet...** and then click **OK**.
- 3 Enter the desired name **layout.css** in both the *Save As* and *URL* fields and verify the *Save* location is inside the site folder. You can now use these styles with other web page files.

# Using Style Sheets for Consistency

---

## Defining Style Sheets for Different Purposes

Since a single Dreamweaver file can have multiple style sheet files attached, for good organization it is recommended to use specific external style sheets for different functions. For example, use one style sheet for layout rules, use another for text formatting rules, and possibly another for image formatting.

## Creating a print Style Sheet

Since style sheets can define output to various media and since a Dreamweaver file can have multiple style sheets attached, you can create pages that automatically look one way when displayed in a browser and a different way when printed. This is accomplished by having a “screen” style sheet and “print” style sheet. Both have rules with identical names but with different characteristics. So, when the page is viewed in the browser, the screen rules apply and when the page is printed, the print rules apply. A key attribute is *Display* in the *Block* category. Setting the value to *none* means objects with that style won’t appear.

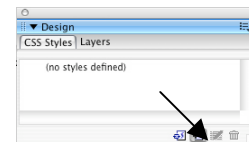
- 1 Duplicate the style sheet for each media desired by copying, pasting, and renaming. A good naming suggestion is to include the media name as part of the new filename. For example if your original style sheet is *layout.css*, then name your “print duplicate” *layoutprint.css*
- 2 Take a look at your document’s code and find the following line of code in the *head* section of your document: `<link href="xxyy.css" rel="stylesheet" type="text/css">`
- 3 Click after *rel="stylesheet"* to add **media="screen"**. The line will now look like this:  
`<link href="xxyy.css" rel="stylesheet" media="screen" type="text/css">`
- 4 In the *CSS* window, click the **Attach File Sheet** icon (chain link).
- 5 Browse to find the file you duplicated above in step 1.
- 6 Click the arrow to the right of *Media* and drag to select **Print**. Then click **OK**.
- 7 Make desired changes to both the print and screen style sheets.
- 8 Save your file.

## Applying styles

- 1 If the *Properties* window is not visible, pull down *Window to Properties*.
- 2 Select object to apply style to.
- 3 Click the desired style in the *Style* popup menu.

## Editing Styles

- 1 Highlight the style to be edited in the *CSS Styles* window and click the *Edit Styles* icon
- 2 Make the appropriate changes in the *CSS Rule Definition* window.
- 3 When finished editing, click **OK**.



## Using “Float” to Position Objects

Text and other objects can “wrap” around another element by specifying a left or right value for the *Float* property (in the *Box* category) of the element being wrapped. A *float* value of *left*, means the element will float to the left of the surrounding elements

- 1 Click the *New CSS Style* icon in the lower right of the *CSS Styles* window.
- 2 Create a *Class* with an appropriate name and click **OK**.
- 3 In the *Box* category set a *Width* for the element (and other values as desired) and then set the *Float* property to *Left* or *Right* as desired.

## Creating Styles with Visible Borders

Styles allow you to add borders of various widths on one, two, three, or all sides of an element.

- 1 Click the *New CSS Style* icon in the lower right of the *CSS Styles* window.
- 2 Create a *Class* with an appropriate name and click **OK**.
- 3 In the *CSS Rule Definition* window enter settings appropriate to the text or image being formatted.
- 4 Click the *Border* category. Specify a *Style* (solid, dashed, etc.), the *Width* (thickness of the line), and the *Color* (of the line) for the *Top*, *Bottom*, *Left*, and/or *Right* borders

# Using Style Sheets for Consistency

---

## Centering Objects

Objects governed by a DIV *class* or *ID* can be centered within their respective containers by setting a width for the object in the *Box* category and setting both the left and right margins (also in the *Box* category) to *auto*. *Width* establishes space for the object. Since margins, are “outside” the box’s width, they define the distance between the box and its container. The *auto* setting tells the browser to provide whatever space is left after the box width, so using *auto* for both the left and right value splits the space “left over” between the two sides resulting in a centered object.

- 1 Pull down *Insert* to *Layout Objects* to *Div Tag*, enter a style name in either *class* or *ID* and click the **New CSS Style** button.
- 2 In the *New CSS Rule* window, verify that type, name, and *Define In* place are correct and click **OK**.
- 4 Click the *Box* category. Specify a *Width* (of the box) and set the *Left* and *Right Margins* to **auto**.
- 5 Add other desired elements to the style as desired.

## Forcing objects so bottom of screen

If your page contains a “footer” you probably want that footer to always be at the bottom of the page. Since the browser reads the tags starting at the top and going down, it’s possible to end up with some strange positioning of the footer. For example if the “last” DIV defined prior to the footer takes up less vertical space than other areas and a *Float* value has been set, the footer might “slide” into that area and no longer be at the very bottom. The *Clear* property can help to make that happen. *Clear* specifies sides of an element that are not affected by other elements. The element with the *Clear* property moves below elements on the specified side. To assure something will be placed at the bottom, the value *Clear* (in the *Box* category) should be set to *Both*.

- 1 Click the *New CSS Style* icon in the lower right of the *CSS Styles* window.
- 2 Create a *Class* with an appropriate name and click **OK**.
- 3 Enter desired values in the various categories.
- 4 In the *Box* category, specify a width and then set *Clear* to *Both*.

## Creating pull quotes

A pull quote is a direct quote from the surrounding text used to call attention to information of interest. The text is usually a larger font and bold and is “set off” from surrounding text with extra “white space” and sometimes a top and bottom border. Depending upon the design, the surrounding text either wraps around the pull quote or is interrupted before the pull quote and continued afterwards.

To create a pull quote:

- 1 Click at the location in the text where you want the pull quote to be.
- 2 Insert a new div with a new *Class* style. (A good name for the class is *pullQuote*.)
- 3 The *CSS Rule Definition* window will open up. In the *Type* category, set the font, size, weight, color, line height, etc. as desired.
- 4 The settings in the *Box* category are critical:
  - Width** – Set this to the number of pixels you want the box to occupy on the screen.
  - Margin** – Top, bottom, right, and left settings define the “white space” around the pull quote.
  - Float** – Specifies whether the pull quote will be position to the left or the right of surroundings.
  - Padding** – Specifies white space between the contents of the pull quote and the boundaries of the box. This setting is necessary if *Borders* are defined.
- 5 If visible lines are desired, select the *Border* category and set the *Style*, *Width* (of the line), and *Color*. (**Note:** Usually for pull quotes, lines are used above and below the contents. In that case, the *Top* and *Bottom* settings should be identical for all.)

## Creating inverse text

- 1 Create a new style in the *CSS* window with an appropriate *Class* name.
- 2 In the *Type* category in the *CSS Rules Definition* window, set the font, size, weight, etc. as desired.
- 3 Set the *Color* to the same color as the background on the screen where the inverse text is to appear.
- 4 In the *Background* category, set a *Background* color that contrasts to the color set above. (**Note:** Usually the background color for inverse text is set to the same color as the surrounding text.)

# Using Style Sheets for Consistency

---

## Creating Style Sheets with text

### Creating and using an inline style

- 1 Within the opening tag, add the attribute *style* followed by an equal sign.
- 2 The style characteristics to be applied follow (inside quotation marks.) A semi-colon is used to separate multiple characteristics. **Example:** `<H1 style="font-size:101px;color:#9933CC">`

### Creating an internal style sheet

- 1 Internal style sheets are placed in the `<head>` section of the HTML document.
- 2 They begin with `<style>` and end with `</style>` and are contained within an HTML comment tag: `<!-- -->`.
- 3 The tag or custom style name is followed by the description, which is inside curly brackets.

#### Example:

```
<style>
<!--
  h1 {font-size:101px;font-style:#italic}           Styles applied to an HTML tag

  h1, h2, h3                                       Styles applied to multiple HTML tags
  {font-family:Arial;
  color:#9933CC
  }

  .bodytext                                       Custom tag created with multiple style characteristics
  {font-family: Arial;
  font-size: 10px;
  color: #012F61;
  text-align: justify
  }
-->
</style>
```

### Using internal styles

The *bodytext* definition shown above could be applied to an HTML tag as it occurs. The custom tag would appear as part of another tag. **Example:** `<P class="bodytext">`

### Creating an external style sheet

External style sheets are plain text files that are saved with a `.css` extension. They can be created with an ordinary text processor. The styles are defined just as they are in the internal style sheet example above except only the styles appear. The opening and closing HTML style tags and the HTML comment tags are not included.

### Using an external style sheet

The style sheet to be used is specified in the `<head>` section of the HTML document with the following code: `<link rel="stylesheet" href="FETCstyles.css" type="text/css" media="screen">`

Then the styles are applied the same as with internal styles.

### CSS Properties

There are too many to attempt to list. Your best bet is to search the web for *Cascading Style Sheets* and study the many options available.