

## Web Development: Cascading Style Sheets (CSS)

Cascading Style Sheet (CSS) is a file that is referenced to by HTML files, to control the look of the Web page.

Style sheets can describe to a browser how HTML tags should appear. All browsers have a default way to decorate HTML elements like Headings <h1>, <h2>, <h3>, <h4>, <h5> and Paragraphs <p> for example. CSS can override these defaults and allow you to format the look of any HTML element by referencing its tag, its identification or its class.

Cascading style sheets are an extremely efficient way to format the look of web documents. CSS is easily applied to HTML in Dreamweaver through the CSS Panel or by directly hand coding the .css files.

Styles allow for users to apply major formatting changes extremely quickly through the Styles Dropdown in the Properties Panel in Dreamweaver, or through the CSS Panel, or directly editing the style sheet in the .css file once styles are assigned. If tags and styles are established, alterations to the CSS results in immediate reformatting of any page elements that have been altered. This can make global changes to a page or even a website extremely efficient. For instance, by changing an h1 style to format text as red, all text with an <h1> tag would change to red throughout any webpage that relates to the .css file.

CSS has the ability to format the font, colour, and size of text, the borders, margins and padding, how lists appear, what background graphic should be used and much more.

Dreamweaver provides pre-made templates using CSS to divide content into different sections. These are available through the New Document Dialogue Box.

## THE CSS PANEL

Create an external or internal CSS code when creating new rules through the CSS Panel. Move Inline Styles to existing External Style sheets through dragging and dropping or right clicking in the CSS Panel.

### Sections

#### All – Current:

Shows all styles in effect or just the current one you're working on  
Current mode shows a Summary for the selection and the About area.

#### Summary for Selection:

This area lists the formatting for the style of the selected text.

#### About Area:

Show info button and show the cascade of rules for the selected tag. This can greatly assist in identifying where specificity is overruling in your styles. They are listed in the order in which they are applied in the browser. Tool tips will show the specificity numbers (see Specificity below).

### Rules area:

Shows the External and Internal styles in the document (does not include inline styles).

### Properties

Displays the properties for the selected style. This section has 3 view modes accessible via the buttons on the bottom left of the panel. Category, List, or Show only the Properties that are applied to that style.

## Style types

### Standard tags

Any standard html tag can be stylized. <p>, <h1>, <li>, <a>, <td>

### groups

list multiple types of HTML elements with commas and assign formatting to all simultaneously. h1, h2, h2, h4, h5, h6, ul, ol, p {...}

### .class

Class is defined through the period before a style and in HTML a class is assigned by name alone through the class attribute.

CSS: .classname {property: attribute property: attribute}

HTML: <p class="classname">text</p>

### #identification

id stylized with a # and in html through the id="..." attribute.

## Style Rendering Toolbar

### View > Toolbars > Style Rendering Toolbar

Shows you how your page could appear on various devices based on your CSS code. Most views require specific media related style sheets for that particular device. Read more: <http://www.w3.org/TR/CSS21/media.html>

### View Page without Style

You can also toggle the displaying of Style sheets with the Yin/Yang icon in the toolbar. View a webpage with or without CSS applied.

### Design Time Style Sheets

You can also set it up to display using two different style sheets, one for when you work on the document and another when it's displayed online. This flexibility allows for easier editing offline and the decorated design online. These are called design time style sheets.

## Cascade, Inheritance, Specificity

### Cascade Effect

#### 1 Browser default

The browser's default formatting.

#### 2 External style sheet

`<link href = "main.css"/>`

#### 3 Internal style sheet

Inside the `<head>` tag

#### 4 Inline style

Inside an HTML element. This is not a standard practice as it adds formatting within an HTML body. Using external style sheets and styles within the header (internal) are recommended standards, inline goes against the usefulness of CSS

Basically, the last style applied in the cascade, wins.

This is the order styles are applied and how each one takes precedence over the next. Inline styles overrule internal styles sheets which in turn overrule External style sheets which overrule browser defaults. There are exceptions to the cascade, read on...

### Inheritance and Specificity

If you tell a `<body>` tag to have blue text, styles used in that `<body>` tag without another style assigned to them will also be blue (this is called inheritance). An element within the `<body>` may have a style that overrides the blue colour with another (specificity).

#### Inheritance

Not all properties are inherited. There is no rule to this, you learn as you use them. For instance, Margins are not inherited by their parents but fonts and font size can be. This is the trickle down effect of elements taking the attributes of parent elements unless otherwise specified.

#### Specificity

When Cascading and Inheritance can't resolve the style specificity takes over. Specificity numbers can be viewed via tooltips in the CSS Panel.

#### Specificity Chart

	<i>Inline</i>	<i>ID</i>	<i>class</i>	<i>Element</i>	<i>Specificity</i>
<code>&lt;p style="..."&gt;</code>	1	0	0	0	1000
<code>h1, h2</code>	0	0	0	2	2
<code>h1.first</code>	0	0	1	1	11
<code>#main</code>	0	1	0	0	100
<code>#main h2</code>	0	1	0	1	101
<code>#main #sidebar p</code>	0	2	0	1	201
<code>h1 + p</code>	0	0	0	2	2
<code>.author</code>	0	0	1	0	10

Higher numbers take precedence over lower ones.

## Multiple External Style Sheets

```
<link rel="stylesheet" type="text/css" href="styles/default.css">
<link rel="stylesheet" type="text/css" href="styles/docs.css">
```

Simply by linking two style sheets you can use more than one level of styling for various pages and purposes.

## Useful HTML tags for defining elements on a page:

Of course the standard body, paragraph, and heading tags can be controlled with CSS. Sometimes areas need to be defined through dividing the document up (header, body, footer) and sometimes specific words or text needs to be emphasized (special text in italic or different colors).

## SPAN tags

Generic inline level tags that can have classes assigned to them. Great for emphasizing certain words within a paragraph for example. The .class Style can be used to format the span tags.

```
.class {propert: attribute}
...<body>
<span class="classname">text</span>
```

## DIV tags

div is short for Division. Use div tags with identification to create structure in a document. By identifying div tags you can assign styles to them using the id style syntax of # before the styles name.

```
#divname {property: attribute property: attribute}
...<body>
<div id="divname">Content for that Section</div>
```

<div> becomes a customizable area using CSS, used specifically to define structure of a document. Common div id tags include: content, header, footer, sidebars, columns, images and so on. The #identity Style can be used to format divisions.

## ATTRIBUTES:

### Box Model

#### Border

All elements have borders. Borders can be coloured and decorated using the border properties.

#### Padding

This is the padding inside an element. Like an inset property. Prevents the content of an element from touching the edges within an elements border

#### Margins

The is like padding except is outside the border of an element. Margins push elements away from each other. Like a text wrap

### Positioning

#### Absolute positioning is good for fixed designs

Relative to the parent container.

Allows for overlapping of elements.

Other elements ignore the Absolute positioned element, making the elements slip under or overlap the absolutely positioned ones.

#### Z-Index

The higher the z-index the closer to the front an element will appear. Elements that overlap using Absolute positioning can thus be controlled using the z-index. Similar to Bring to Front and Send to back in other adobe products.

#### Relative positioning

Relative to the elements own starting position.

Other elements react to the flow. Saving the contents position in the document flow. This allows for nudging of elements from their standard locations.

#### Floats

Work similar to Inline Text wraps when effecting inline code. You can make a picture float to the right of text. Floats remove elements from normal document flow, so can result with elements slipping under each other.

Allows floating elements to float beside each other, create columns with float left on each element. Still limited to float within a container around floating elements.

Container width can help control flow of floating elements.

#### Clear

Clearing an element allows an object to ignore the effects of a floated object.

#### Backgrounds:

```
{background: url(../) no-repeat top left;}
```

## Hiding Text

Block – Display: none (but hides from some screen readers as well)

Box – Margin: -9999 forces text off of page to the left, but a background image would still remain in the element.