

Practical Application of Cascading Style Sheets for Accessibility and Useability

Summary: Beginners to advanced CSS - through structure, implementation and use - to advanced design techniques, tricks and tips.

Description:

1) The introduction to CSS will cover the components of CSS as well as practical implementation and basic examples of CSS in use. It will cover laying out simple HTML to work with CSS and creation of a simple external style sheet (inline and internal styles will also be touched upon).

2) The practical application of CSS tutorial will cover use of CSS to make web pages accessible to a range of impaired users, without disadvantaging the experience other users: from creating 'standards complaint' code, to optimizing the content for screen-readers. This tutorial includes working with a variety of designs and demonstrate the radical changes that can be achieved by switching styles without touching the HTML.

Audience: Web developers, designers and maintainers. This tutorial assumes a basic knowledge of HTML.

Time: 3 hrs

Introduction to CSS

Cascading Style Sheets:

- Control how your pages look.
- Incorporate style into your web pages.
- Help you to keep style and content separate.

With CSS you can:

- Set colours and fonts for your text including: headings, paragraphs, links, etc.
- Control the location of elements of your page with margin, padding, position, etc.
- Add borders, place images, display backgrounds and a whole lot more!

Creating a style

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

The selector names what the style will apply to.

The property names what aspect of the selector you're going to change.

The value names exactly how the property will change.

```
body { background: #99ccff }
```

The background of the body of the page will be light blue.

```
p { color: #333333 }
```

All text within a p tag will be dark gray.

You can apply more than one property by separating them with a semi colon.

You can use as many spaces as you like between properties to make it easy for yourself to read.

```
h1 { font-size: 1.1em;  
color: #003366;  
font-family: verdana, arial, helvetica, sans-serif; }
```

Note: units are up to you. For example, the font size here might have been 14px or 12pt.

You can apply the same properties to more than one selector by separating them with a comma.

```
h2, h3 { color: #000066 }
```

All text within h2 and h3 tags will now be dark blue.

Class selectors

Let's say that you want certain areas of your text to be red.

```
.red { color: #990000 }
```

Class selectors let you apply styles wherever you like.

```
<p class="red">This text will be red</p>  
<h2>This text will be dark blue but <div class="red">this text will be red</div></h2>
```

If no style has been set for an HTML element, it will inherit the style of the parent element.

```
<p class="red">This text will be red and <b>this text will be bold and still red</b></p>
```

You can limit class selectors to certain HTML elements.

```
p.red { color: #990000 }
```

Only paragraphs with a class of "red" will now appear red.

Headings with a class of "red" will not appear red..

ID selectors work just like class selectors. Web standards dictate that ID selectors only be used once in a page.

```
#content { margin: 20px }  
<div id="content">This container will be surrounded by a margin of 20 pixels.</div>
```

They are useful for controlling large consistent areas, e.g. #page, #navigation and #footer

Contextual selectors let you define styles for elements depending on where those elements are located.

```
#navigation a { color: #ffffff }
```

Links will be white only when they appear within the navigation area of the page.

Pseudo-classes can break up selectors with a colon so that different styles can be applied during different circumstances. They are commonly used for links.

```
a:link { color: #003366; }  
a:visited { color: #990033; }
```

To apply a class selector to a pseudo-class, it must be defined before the colon.

```
a.purple:link { color: #663399; }  
<a class="purple" href="page2.html">This link will be purple</a>
```

Some values can contain a number of style instructions separated by spaces.

```
#header { background: #e5f0f7 url("head.gif") no-repeat top right; }
```

The background colour of the header area is light blue.

The header area of the page will have a background image: "head.gif". (The path is relative to the style sheet, not the HTML page.)

It won't repeat (tile).

The background image will be located in the top right hand corner of the header area.

If you don't want to list all the values in one property, you can break them up.

```
#header { background-color: #e5f0f7;  
background-image: url("head.gif");  
background-image: no-repeat ;  
background-position: top right; }
```

It is sometimes useful to leave comments in your style sheet.

Anything between /* and */ will be ignored.

```
/* This section contains all the styles for tables */  
/* The reason I used margin here and not padding is because... */
```

Applying CSS to your page

There are three different ways of adding CSS to your page, and you can use all of them at once:

- External style sheet(s) linked into your page
- Internal styles inside the <head> tag
- Inline styles within the HTML of the page

Inline styles override internal styles, which override external styles.

External styles

There are two ways of linking external style sheets from the <head> area of your page:

```
<link rel=stylesheet TYPE="text/css" href="/styles/colours.css">  
or  
<style type="text/css"> @import url(/styles/colours.css) </style>
```

The styles in this example are defined in a separate file; 'colours.css' in the 'styles' folder. External styles are my favourite because they facilitate site-wide changes.

Internal styles

Internal styles are defined in the <head> area of your page:

```
<style type="text/css">  
body { background: #99ccff }  
p { color: #333333 }  
</style>
```

Inline styles

Inline styles are applied directly within the code of your page to individual elements.

```
<b style="color: #9933cc">This text will appear bold and purple </b>
```

You might like to start with some basic CSS such as font family and colour.

```
#content { color: #333333; font-size: 0.8em; font-family: verdana, arial, helvetica, sans-serif; }
```

```
h1 { color: #333366; font-size: 1.2em }
```

It's often a good idea to begin with a main style (contained for e.g. within <div id="content"></div>), a set of headings, links etc., before heading on into the tricky stuff.

Why make websites attractive?

- Attractive sites may encourage visitors to your site and once there, want to hang around.
- Making a visual impact on your audience may help you to get your message across.
- Attractiveness can go hand in hand with accessibility and useability. For example, a site that loads slowly will be ugly (if only momentarily), as well as unusable.
- Consistency of layout and design not only makes your site more accessible, it also presents a stronger visual impact.

Why make websites useable?

- Every user should be able to access the content of your site easily and quickly.
- The most useable website is one where your visitors don't have to think about how they're using the site at all – they just use it!
- Creating a useable site means that your visitors will stay in your site longer, and you'll be able to direct them to the information that you want to present.

Why make websites accessible?

- You want as many visitors as possible to be able to absorb the content of your site, regardless of technology or ability.
- Ensuring that your site can be read by assistive technologies also means that search engines will have an easier time indexing your pages.
- Accessibility isn't as much work as you might think. A well designed site can be effortlessly accessible.

Why use CSS?

CSS emphasises a separation of form and content. This makes updating your site quicker and easier and can ensure that your site is easily interpreted by assistive technologies.

How?

Everyone has a different idea of what is attractive, so to some extent, I'm going to have to leave that bit up to you! However I will demonstrate some techniques for attractive/accessible/useable layout, which you can alter with your own graphics, dimensions, fonts and so on.

NOTE: CSS does not work exactly the same way in all browsers (in fact it probably won't) but you should be able to ensure that your site is still accessible, useable, and attractive in any browser.

CSS Techniques

All of the values in these examples may be changed to suit your design.

Examples beginning with `.name` or `#name` are CSS; examples beginning with `<tags>` are HTML.

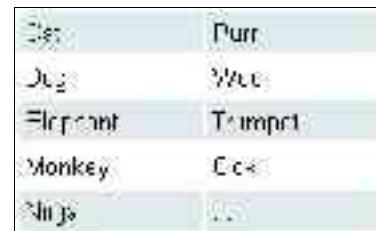
Use small repetitive images to fill space. A background colour should also be set to provide sufficient contrast when images are not displayed.

```
.navigation { background: #660000 url("navigation.gif") top left; }
```

Shade alternate rows in data tables (accessible and attractive).

```
.odd td { background: #ddecec; }
```

```
.even td { background: #ffffff; }
```



Dog	Purr
Dog	Walk
Elephant	Trumpet
Monkey	Cuck
Ninja	AS

Change the appearance of 'focus' links for keyboard users as well as 'active' links. The hover colour has been added here too.

```
a:hover, a:active, a:focus { color: #990000; }
```

Use continuous images (sprites) to create stretchable boxes.

Create a pretty box image, longer than you think that you will need content for, with a repeating pattern down the side. Set the width of the box to the same width as the image.

```
.box { width: 170px; margin: 10px; background: url("box.gif") bottom left no-repeat; }
```

```
.box h1 { margin: 0px; padding: 20px 16px 1px 16px; background: url("box.gif") top left no-repeat; }
```

You can use anything here to mark the top of the box (I've used `h1`) but it needs to be right at the top, and the only one of its kind within the box. The HTML might be:

```
<div class="box">  
    <h1>This is a neat box</h1>  
    <p>This is the content of the box and it doesn't matter how much there is.</p>  
</div>
```

If you want your navigation menu (or anything else) to stay in the same place on the page and move with the scrollbar when a page is scrolled, use fixed position.

```
#navigation { position: fixed; top: 10px; right: 10px; width: 170px; margin: 10px; }
```

Use 'float' to create boxes within the content area that sit beside each other and move sensibly if the screen is resized.

```
.mycolumn { width: 170px; margin: 10px; float: left; }
```

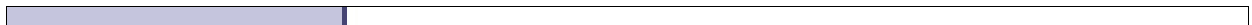
You might need to clear the boxes before including any subsequent content (e.g. a footer area).

```
<div style="clear: both">&nbsp;</div>
```

You can create the illusion of columns of the 'same' height, no matter how much content each column contains. The HTML for the layout uses three styles:

```
<div id="page">
  <div id="navigation"> </div>
  <div id="content"> </div>
</div>
```

Create an image (columns.gif) which can repeat vertically containing the false column effect:



```
#page { background: transparent url("columns.gif") top left; padding: 0px; margin: 0px; width: 100%; }
```

```
#navigation { padding: 10px; width: 150px; }
```

```
#content { padding: 20px 20px 0px 170px; clear: left; }
```

The left hand padding of the content should be a bit wider than width of the false column effect.

You can create an image which appears to sit in the bottom right hand corner and 'overlap' the content area. Note that both backgrounds are set to 'no-repeat'.

```
body { background: url("corner.gif") bottom right no-repeat; }
```

The content area left overlaps the image set as the background of the page. Then include just the area of the image that gets covered by the page area to complete the picture.

```
#content { background: #ffffff url("corner2.gif") bottom right no-repeat; }
```

*Use margin and not padding on the page area as IE treats padding differently to other browsers.



Including a separate style sheet for printing:

```
<link rel="stylesheet" type="text/css" href="main.css">
```

```
<link rel="stylesheet" type="text/css" media="print" href="print.css">
```

Conditional comments: including a separate style sheet for a different browser, e.g. any version of IE earlier than 7.5:

```
<link rel="stylesheet" type="text/css" href="main.css">
```

```
<!--[if lte IE 7.5]>
```

```
<link href="ie.css" rel="stylesheet" type="text/css">
```

```
<![endif]-->
```

Note: you must have the version (7.5) at the end or the 'if' comment will display in the body.

HTML Page Example

```
<html>
<head>
  <title>Android Design</title>
  <link href="style2.css" rel="stylesheet" type="text/css">
</head>

<body>
<div id="page">
  <div id="header">
    
  </div>

  <div id="navigation">
    <ul>
      <li><a href="#">About Us</a>
        <ul><li><a href="#">Our History</a></li></ul>
      </li>
      <li><a href="#">Showcase</a></li>
      <li><a href="#">Customers</a></li>
      <li><a href="#">Services</a></li>
      <li><a href="#">Contact Us</a></li>
    </ul>
  </div>

  <div id="content">
    <h1>Lorem ipsum</h1>
    <p class="pull">All of our designs are guaranteed 100% Ninja proof!</p>
    <p class="blue">Claritas est etiam processus dynamicus, qui sequitur mutationem </p>
    <p>Dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh
      <a href="#">commodo consequat</a>. </p>
    <h2>Claritas est</h2>
    <p>Claritas est etiam processus dynamicus, qui sequitur mutationem .</p>
    <div style="clear:both">&nbsp;</div>
  </div>

  <div id="footer">
    <a href="#">View the CSS</a> <a href="#">Back 1</a> <a href="#">Home</a>
  </div>

</div>

</body>
</html>
```

CSS Example 1

```
html { background: #CC9999; }

body {
    margin: 0;
    padding: 0;
    color: #333333;
    font-family: verdana, arial, helvetica, sans-serif;
    font-size: 0.9em;
}

#page {
    margin: 10px;
    background: #ffffff;
}

a:link, a:visited { color: #003366; }

a:hover, a:active, a:focus { color: #990033; }

/* Header */

#header {
    background: #bdddf7 url(pretty-head.gif) top right no-repeat;
    width: 100%;
    height: 92px;
    margin: 0px 0px 10px 0px;
    border: solid 1px #003366;
}

#header img { margin: 8px 10px 0px 10px; }

/* Content */

#content { margin: 20px; }

#content h1 {
    font-size: 1.2em;
    color: #990033;
    display: block;
    border-bottom: 1px solid #990033;
    background: #ffffff url(pretty-heart.gif) top right no-repeat;
}

#content h2 {
    font-size: 1.1em;
    color: #336699;
}

/* Navigation */

#navigation { margin: 20px; }

#navigation ul {
    margin: 0px;
    padding: 10px;
}

#navigation ul ul {
    margin: 0px;
    padding: 0px 0px 0px 16px;
}
```

```

#navigation li {
    list-style-type: none;
    margin: 6px 0px 2px 0px;
}

#navigation a {
    padding: 1px 5px;
}

#navigation a:link, #navigation a:visited {
    text-decoration: none;
}

#navigation a:hover, #navigation a:active, #navigation a:focus {
    text-decoration: none;
}

/* Footer */

#footer {
    border: solid 1px #003366;
    padding: 10px;
    text-align: center;
    background: #bdddff;
}

#footer a {
    border: solid 1px #003366;
    text-decoration: none;
    padding: 1px 4px;
    font-weight: bold;
    margin: 0px 20px;
}

#footer a:link, #footer a:visited {
    background: #e5f0f7;
    color: #336699;
}

#footer a:hover, #footer a:active, #footer a:focus {
    background: #ffffff;
    color: #003366;
}

/* Extra styles */

.pull {
    background: #daf0fe;
    margin: 2px 6px;
    padding: 6px;
    border: solid 1px #003366;
    color: #003366;
    font-weight: bold;
    width: 140px;
}

.blue { color: #336699; }

```

CSS Example 2

```
html { background: #CC9999; }

body {
    margin: 0;
    padding: 0;
    color: #333333;
    font-family: verdana, arial, helvetica, sans-serif;
    font-size: 0.9em;
    background: #CC9999 url(android1.gif) bottom right no-repeat;
}

#page {
    margin: 10px 85px 0px 10px;
    background: #ffffff url(android2.gif) bottom right no-repeat;
}

a:link, a:visited { color: #003366; }

a:hover, a:active, a:focus { color: #990033; }

/* Header */

#header {
    background: #bdddf7 url(pretty-head.gif) top right no-repeat;
    width: 100%;
    height: 92px;
    margin: 0px 0px 10px 0px;
    border: solid 1px #003366;
}

#header img { margin: 8px 10px 0px 10px; }

/* Content */

#content {
    margin: 20px 50px 40px 185px;
    background: #ffffff;
}

#content h1 {
    font-size: 1.2em;
    color: #990033;
    display: block;
    border-bottom: 1px solid #990033;
    background: #ffffff url(pretty-heart.gif) top right no-repeat;
}

#content h2 { font-size: 1.1em; color: #336699; }

/* Navigation */

#navigation {
    position: absolute;
    top: 115px;
    left: 15px;
    width: 157px;
    margin: 10px;
    color: #003366;
    background: url(pretty-box.gif) bottom left no-repeat;
}
```

```

#navigation h1 {
    margin: 0px;
    padding: 30px 16px 1px 16px;
    background: url(pretty-box.gif) top left no-repeat;
    font-size: 1em;
}

#navigation ul { margin: 0px; padding: 10px 10px 80px 16px; }

#navigation ul ul { margin: 0px; padding: 0px 0px 0px 16px; }

#navigation li { list-style-type: none; margin: 6px 0px 2px 0px; }

#navigation a { display: block; padding: 1px 5px; }

#navigation a.link, #navigation a.visited {
    text-decoration: none;
    border: solid 1px #daf0fe;
}

#navigation a:hover, #navigation a.active, #navigation a.focus {
    text-decoration: none;
    background-color: #ffffff;
    border: solid 1px #003366;
}

/* Footer */

#footer {
    border: solid 1px #003366;
    padding: 10px;
    text-align: center;
    background: #bdddf7;
}

#footer a {
    border: solid 1px #003366;
    text-decoration: none;
    padding: 1px 4px;
    font-weight: bold;
    margin: 0px 20px;
}

#footer a.link, #footer a.visited { background: #e5f0f7; color: #336699; }

#footer a:hover, #footer a.active, #footer a.focus { background: #ffffff; color: #003366; }

/* Extra styles */

.pull {
    background: #daf0fe;
    margin: 2px 6px;
    padding: 6px;
    border: solid 1px #003366;
    float: right;
    color: #003366;
    font-weight: bold;
    width: 140px;
}

.blue { color: #336699; }

```

Further Information

Contact Information

Feel free to contact me regarding this tutorial.

Email: amanda@dd.com.au

Web: <http://amanda.dd.com.au>

Where to from here?

There is a lot more to learn about CSS! Now that you know how to apply it, the next step is to play.

What are all of the properties that you can apply to selectors, and what values can you give them?

The World Wide Web Consortium is a good place to find out.

<http://www.w3.org/Style/CSS/>

References

12 Lessons for Those Afraid of CSS and Standards –

www.alistapart.com/articles/12lessonsCSSandstandards/

456 Berea St – www.456bereastreet.com/

A CSS Walkthrough – www.christianmontoya.com/2006/10/22/a-css-walkthrough/

A List Apart - CSS Sprites: Image Slicing's Kiss of Death - www.alistapart.com/articles/sprites/

A List Apart - Power To The People: Relative Font Sizes - www.alistapart.com/articles/relafont/

Accessible and Attractive Websites – www.sitepoint.com/article/attractive-websites

Beginners Guide to CSS – archivist.incutio.com/viewlist/css-discuss/33996

Brainjar: CSS Positioning - www.brainjar.com/css/positioning/

Creating Liquid Faux Columns - www.communitymx.com/content/article.cfm?page=1&cid=AFC58

CSS Layout Techniques: for Fun and Profit - www.glish.com/css/

Fast Rollovers Without Preload - wellstyled.com/css-nopreload-rollovers.html

Floatutorial - css.maxdesign.com.au/floatutorial/

Forcing scrollbars - pixel pushing - www.splintered.co.uk/experiments/49/

Mozilla web developer toolbar - addons.mozilla.org/extensions/moreinfo.php?id=60

Position is Everything - www.positioniseverything.net/index.php

The Man in Blue – footerStickAlt – www.themaninblue.com/experiment/footerStickAlt/

The minimalist backlash – www.christianmontoya.com/2006/09/25/the-mimimalist-backlash/

Validate your markup – validator.w3.org/

Zen Garden – www.csszengarden.com/