



# **HTML5 and CSS3**

## **Web Coding for Designers**

### **Lesson 2: Styling with CSS**

Michael Slater, CEO  
Lisa Irwin, Sr. Developer  
[course-support@webvanta.com](mailto:course-support@webvanta.com)  
888.670.6793

**[www.webvanta.com](http://www.webvanta.com)**

# Welcome!

- Four sessions
  - 1: Semantic HTML
  - **2: Styling with CSS**
  - 3: Putting it All Together
  - 4: The Cool New Stuff



# Questions

- Ask questions via chat during the live course
  - Second instructor standing by
- Join the discussion group
- Email us
  - *Please use the discussion group* for non-private questions



Michael Slater  
Cofounder & CEO



Lisa Irwin  
Web Developer

# Goals for This Lesson

- Demystify CSS
- Be able to create simple CSS, modify anything
- Key topics
  - Basic CSS structure and syntax
  - Writing efficient CSS
  - Typography
  - Backgrounds
  - Margins and padding

# Basic CSS Structure

```
h1 {  
  font-size: 30px;  
  font-weight: bold;  
}
```

```
.myclass {  
  color: #900;  
}
```

```
#myID {  
  font-style: italic;  
}
```

- Start with element name, class, or ID
- Repeat as needed
  - property: value;
- No space between numbers and units
- All other spaces ignored

# Coding Style

```
.myclass { font-size: 10px; color: #666; }
```

```
.myclass{font-size:10px;color:#666;}
```

```
.myclass {  
  font-size: 10px;  
  color: #666;  
}
```

```
.myclass  
{  
  font-size:10px;  
  color:#666;  
}
```

- A personal choice
- Trade of compactness vs. readability
- Editors can help with formatting

# Style Multiple Items at Once

```
h1, h2, .myhead, #special {  
  font-size: 72px;  
  font-weight: 700;  
}
```

- List as many items as you want
- Each acts independently
- Minimize amount of code

# Target Elements Using DOM

```
<div class="article">  
  <h2>My Headline</h2>  
  <p class="byline">By Michael Slater</p>  
  <p>This is the text of my article.</p>  
  <p>This is the second paragraph.</p>  
</div>
```

```
.article h2 { color: green; }
```

```
.byline { font-style: italic; }
```

```
.article p { font-family: Georgia, serif }
```



# Using Tag *and* Class or ID

```
div.article h2 { color: green; }
```

```
p.byline { font-style: italic; }
```

```
div.article p { font-family: Georgia,  
serif }
```

- Helps with specificity (see cascade, coming up)
- Improves performance
- Adds clutter to CSS file

# Fancy Selectors

- **Descendent**
  - `.article p` (p tag inside element with class "article")
- **Child**
  - `.article > p` (p tag immediate inside element with class "article")
- **Adjacent**
  - `.article + img` (image at same DOM level as element with class "article")
- **Attribute**
  - `a[href="/"]` (a tag with href pointing to root of site)
- **Pseudo-elements**
  - `.article:first-line` (first line in the element with class "article")
  - `.article:first-child` (first child element inside element with class "article")
  - `p:first-letter` (first letter of a paragraph)
  - `.article:nth-child(6)` (**CSS3** - 6th element inside element with class "article")

# The Cascade

- What if more than one rule applies to the same element, and there are conflicts?
  - Multiple rules in same stylesheet
  - Multiple stylesheets
    - Browser has built-in styles
    - User can provide user stylesheet
    - Author may include multiple stylesheets
  - Inline rules
    - `<p style="font-weight: bold">`

# Who Wins?

- If all else equal, last rule wins
- Inline styles override stylesheets
- More specific rules win
  - Parent elements
  - ID
  - Class
  - Element name
- !important trumps them all

# CSS Reset

- Browsers have built-in stylesheets
  - And they aren't all the same!
- CSS reset is a set of rules that provides a consistent starting point
- Be sure to add back in the things you expect
  - Bold and italic for `<strong>` and `<em>`
  - Bullet lists

# Baseline CSS Styles

- Start with styles for all the common elements
- Put declarations in as few places as possible
  - Don't repeat font declarations all over the place
  - Group together elements that share a declaration

# Typography

- font-size
  - pixels, ems, keywords, relative, percentage
- font-style
  - normal, italic, oblique
- font-weight
  - normal, bold, 100, 200, ... 900, lighter, bolder
- font-variant
  - normal, small-caps

# Finer Points of Typography

- Leading
- Character spacing
- Baselines
- Small caps
- Accented characters
- Typographic marks



# Fonts on the Web

- HTML specifies characters, not how they should be rendered
- Fonts are part of computer on which browser runs
- CSS can request fonts that may be:
  - Defaults built into browser
  - Fonts resident on the computer
  - Fonts loaded from files that are part of website
  - Fonts loaded from a third-party web service

# Font Stacks

- `font-family` defines a list of acceptable alternatives
  - Browser uses the first one it can find
- Start with your first choice
- End with one of the built-in generic fonts
  - serif, sans-serif, cursive, fantasy, monospace
- Keep in mind your visitors may see something different from what you see!

`font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;`

`font-family: Georgia, Times, "Times New Roman", serif;`

`font-family: Monaco, Consolas, "Lucida Console", monospace;`

`font-family: Papyrus, fantasy;`

# Web-Safe Fonts: the Core Set

Arial

Tahoma/Geneva

Trebuchet MS

Verdana

Georgia

Palatino

Times New Roman

Comic Sans

Courier New

Lucida/Monaco

**Impact**

# The Bad Old Days

- Old ways of getting around font limitations
  - Images
  - Image replacement
  - Flash replacement
  - Cufon

# Image Replacement

- HTML has text for SEO, accessibility
- Text hidden for most users, replaced with image

```
<h1 id="logo">My Company Name</h1>
```

```
h1#logo {  
    width: 300px; height: 75px;  
    background: url("/images/logo.png");  
    text-indent: -9999px;  
}
```

# @font-face

- Load a font file, just like an image
- Increases page load time
- Biggest issues are legal, not technical

```
@font-face {  
    font-family: MyFancyFont;  
    url: (/fonts/myfancyfont.eot)  
}
```

# Web Font Services

- One-stop source for web font solutions
  - CSS code, font files (download or link)
  - Licensing
- [Google.com/webfonts](https://google.com/webfonts)
  - Free!
- Commercial Web Font Service Bureaus
  - [www.typekit.com](http://www.typekit.com)
  - [www.webINK.com](http://www.webINK.com)
  - [webfonts.fonts.com](http://webfonts.fonts.com)

# Web Font Considerations

- Small to modest increase in cost
- Increases page load time
  - Limit weights, styles, and character set
  - Browser cache serves pages after first one
- Rendering quality is highly variable
  - Test in variety of browsers and operating systems
  - Look for samples on service bureau sites



# Color Values in CSS

- By Name
- By Hex Value
  - #80FF36 (8 bits = 2 chars each: red, green, blue)
  - #642 (shorthand for #664422)
- By Decimal Value
  - rgb (10, 20, 30)

<b>maroon</b> #800000	<b>red</b> #ff0000	<b>orange</b> #ffa500	<b>yellow</b> #ffff00	<b>olive</b> #808000
<b>purple</b> #800080	<b>fuchsia</b> #ff00ff	<b>white</b> #ffffff	<b>lime</b> #00ff00	<b>green</b> #008000
<b>navy</b> #000080	<b>blue</b> #0000ff	<b>aqua</b> #00ffff	<b>teal</b> #008080	
<b>black</b> #000000	<b>silver</b> #c0c0c0	<b>gray</b> #808080		

<http://sixrevisions.com/css/colors-webdesign/>

#80ff36 = 1000 0000 1111 1111 0011 0110

# Web-Safe Colors

- ***An out-of-date concept!***
- Long ago, many computers limited to 4 bits per color =  $16 \times 16 \times 16 = 4,096$
- Now  $256 \times 256 \times 256 = \mathbf{16,777,216 \text{ colors}}$
- Color fidelity is another matter entirely
  - Largely out of your control
  - sRGB is web's default color space
  - Better/newer browsers respect embedded profiles

# The Fourth Dimension

- RGBA = Red Green Blue **Alpha**
- Alpha = opacity, 0 to 1.0
- Not supported on IE before IE9
  - precede with another css declaration with RGB value
- `rgba(255, 0, 255, 0.5)`
  - maximum red and blue, no green, 50% transparent

# Lists

- Unordered Lists
  - Like this one!
  - Bullet can be anything or nothing
    - Can be nested
- Ordered Lists
  - For things that are in a sequence
  - Number can be displayed or not
    1. First item
    2. Second item

# List Styling

- list-style-type
  - different options for ordered and unordered
- Can use images for bullets
- Need to set left padding or margin

## Disc

- First item
- Second item
- Third item

## Square

- First item
- Second item
- Third item

## Circle

- First item
- Second item
- Third item

## None - no bullets

First item  
Second item  
Third item

## Decimal

1. First item
2. Second item
3. Third item

## Decimal with leading zeros

01. First item
02. Second item
03. Third item

## Lowercase ascii letters

- a. First item
- b. Second item
- c. Third item

## Uppercase ascii letters

- A. First item
- B. Second item
- C. Third item

## Lowercase roman numerals

- i. First item
- ii. Second item
- iii. Third item

## Uppercase roman numerals

- I. First item
- II. Second item
- III. Third item

# Backgrounds

- Any element can have a background
  - Solid color
  - Image, or cropped part of an image
  - Image slice repeated horizontally or vertically
  - Image tile (typically repeated both ways)
- Many separate CSS properties
  - With a shorthand form

# Background Properties

- background-color
- background-image
  - url(filename.jpg)
- background-repeat
  - repeat, repeat-x, repeat-y, no-repeat
- background-attachment
  - scroll, fixed
- background-position
  - %, pixels, or top/center/bottom, left/center/right

# Tiling Backgrounds

- If left and right edges match, and top and bottom edges match, a tile of any size can be seamlessly repeated forever

```
background-image: url('paper-tile.jpg');  
background-repeat: repeat-y;
```

```
background: url('paper-tile.jpg') repeat-y;
```



# Repeating Stripes

- Narrow (even only 1px-wide) stripe can be repeated indefinitely
  - Horizontally or vertically

```
background: #fff url('gradient.jpg') repeat-x;
```

# Links

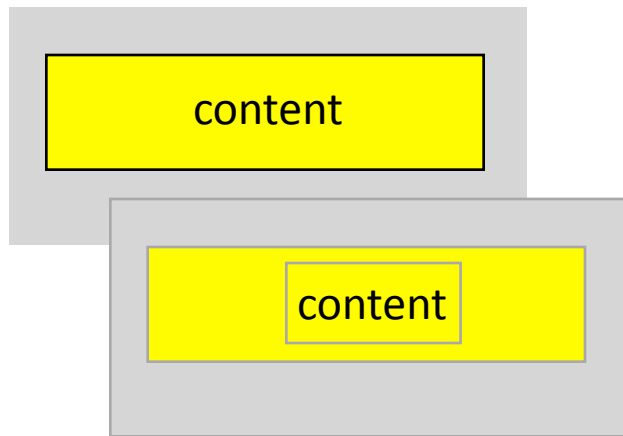
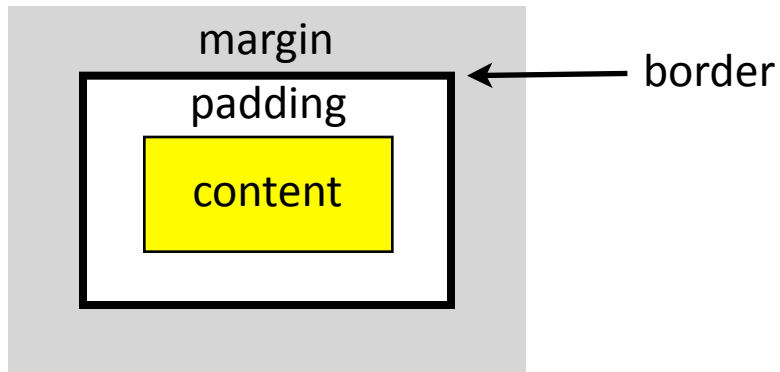
- Links have multiple states, each of which can be styled separately
  - hover
  - active
  - visited

```
a:link {  
    text-decoration: none;  
    color: blue;  
}  
a:visited {  
    color: green;  
}  
a:hover {  
    text-decoration: underline;  
}  
a:active {  
    text-decoration: underline;  
}
```

# Document Flow

- Browser simply takes HTML elements one at a time, and adds them to the page based on CSS styling
- Block-level elements have implicit line break before and after
- Inline elements are, well, inline
- Unless "positioned", everything is simply rendered in a single column with no overlaps

# Margins and Padding



- width specifies ***content width***
- each side can be spec'd in clockwise order:
  - top, right, bottom, left
- touching top and bottom margins collapse
  - only largest survives

# Homework

1. Start with one of the HTML documents you created as homework for session 1.
2. Create CSS to change the appearance.
3. Check it in different browsers.
4. How different can you make it look just by changing the CSS?
5. What can you do with background images?

# If This Isn't Working for You

- Let us know before the next lesson and we'll refund all your fees
- **And if it is...**
  - Help spread the word so we can develop more courses
    - Share via your favorite social media networks
    - Email a friend or a mailing list
    - Write something on your blog
  - It's not too late for your friends to join
    - [www.webvanta.com/courses](http://www.webvanta.com/courses)
    - Click the share links on each of the course pages