



Welcome

What This Course Will Cover

- **Cascading Style Sheets**
 - **Why they are a better way**
 - **The syntax**
 - **Development Tools**
 - **Accessibility Issues**

Week One

- **Focus is on getting your feet wet:**
 - **What are the three common methods for styling your code?**
 - **How do these methods interact?**
 - **Basic styling of fonts**
 - **Introduction to placing elements**

Week Two

- **Creating different layouts**
- **Styling links and lists**
- **Advanced Selectors**
- **Browser Capabilities**
- **Accessibility**

Week Three

- **Box Model**
- **Measurements**
- **Positioning**

Week Four

- **Pseudo classes and elements**
- **Transitions and Transforms**
- **Working on Final Project**

Who is this class for?

- This class is for those who are already familiar with HTML5
- Who have persistence
- Who are willing to do more than I do in these lectures

Who Am I?

- **Ph.D. in Computer Science**
- **Two decades of teaching experience**
- **Emphasis on education for those who are new to tech**

Workload

- **Weekly quizzes – short**
- **Peer-graded assignments**
- **Demonstrate general capabilities to code what we have learned. Students will have similar HTML, but will create a unique look.**

Succeeding in This Class

- In a perfect world you would code with a friend...so use the message boards.
- Never spend more than 15 minutes on one thing that doesn't work. Move on.
- Look things up on your own!
- Practice, practice, practice!

© Colleen van Lent
University of Michigan
School of Information

Unless otherwise noted, this work is licensed under the
CC BY-NC 4.0 license.

CSS3

Cascading Style Sheets

Adding Style to Your Pages

Browser Default Styling

- The same html file may look different when viewed on different browsers.
 - Some tags are supported, some aren't
 - Browsers may have different *default styles*
- In general, default looks are plain.

Cascading Style Sheet

- CSS defined generic rules that can apply to multiple elements

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

```
h1{  
    color:blue;  
}
```

Styled Heading

Adding Style

- As styling tags were phased out of html, styling was done with style attribute

```
<h1 style = "color:blue;">Styled Heading</h1>
```

- Violated separation of content/style

Styled Heading

Rule Syntax

- **Brackets and semicolons are very important**
- **This is where a good editor can make a BIG difference**

```
/* This is how comments are done */
```

Multiple Properties

```
h1 {  
    color:blue;  
    background-color:yellow;  
}
```

Styled Heading

Internal Style Sheet

- Styling is defined within `<head>`
- Rules are defined within `<style>`
- Styles are applied to all elements in that file

```
<head>  
  <meta charset="UTF-8">  
  <title>Title here</title>  
  <style>  
  
  </style>  
</head>
```

- Don't forget to close the style tag!!

External Style Sheet

- You can put rules in an external file (don't use the style tag!!)
- A link to the style sheet is put in the head section.

```
<link rel="stylesheet" href="style.css">
```

- Styles are applied to all elements in all files that links to the style sheet

© Colleen van Lent
University of Michigan
School of Information

Unless otherwise noted, this work is licensed under the
CC BY-NC 4.0 license.

CSS3

The "Cascading" of CSS3

How styles interact

The "Cascading" part of CSS

- **Browser default**
- *External* style sheets
- *Internal* style (in the head section)
- *Inline* style (inside an HTML element)

Rule precedence

- **What if one selector is defined in two external files?**
 - The rules from the most recent file have precedence
- **What if one selector has more than one rule in the same file?**
 - The most recent rule has precedence

```
h1 {  
  color: blue;  
  font-family: Arial; }  
h1 {  
  font-family: Times; }
```


!important

- It is possible to override later rules, use !important

```
h1 {  
  color: blue;  
  font-family: Arial !important;}  
  
h1 {  
  font-family: Times;}
```

Review

- **Why do we want/need to separate content from formatting?**
- **How does this also tie in to external/internal style sheets?**
- **Understand that this is very powerful. See [CSS Zen Garden](#)**

© Colleen van Lent
University of Michigan
School of Information

Unless otherwise noted, this work is licensed under the
CC BY-NC 4.0 license.

Styling Your Text

Styling Your Text

Options

- **Many options for styling your text:**
 - **font (family, style, variant, size)**
 - **color and background**
 - **alignment**
 - **line-height**

font-family

- Font families are styles of text

Examples:

Helvetica, Courier, "Courier New", "Comic Sans MS", cursive, Verdana

font-family

```
h1 {  
  font-family: Arial;  
}
```

Styled Heading

font-family

- **Not all font-families supported by all of the operating systems, so you can provide alternatives.**

```
h1 {  
    font-family: Courier, Impact,  
    Arial;  
}
```

font-family Considerations

- Some fonts are not as user-friendly, use sans-serif when possible.

Test

Test

Custom fonts

- To expand beyond "web-safe" fonts use @font-face

```
@font-face {  
    font-family: mySpecialFont;  
    src: url('Colleen.ttf'); }  
  
h1 {  
    font-family: mySpecialFont;}
```

font-style

- **font-style:**
 - **normal**
 - **italic**
 - **oblique**

Normal
Italic
Oblique

font-variant

- **font-variant:**
 - **normal**
 - **small-caps**

```
h1 {  
    font-variant: small-caps;  
}  
<h1>Small caps variation</h1>
```

SMALL CAPS VARIATION

font-size

- This is only the beginning of our discussion on sizes...
- Options
 - xx-small, x-small, small, smaller
 - medium
 - larger, x-large, xx-large, larger
 - Use pixel
 - Use %

color and background-color

- The color attribute is the color of the foreground.
- The background-color is the color of the background

colors

```
h1, span {  
    color:#0000FF;           /* Blue */  
    background-color: #B3B3B3; /* Grey */  
}
```

Colors

Notice the difference for **inline** elements!

text-align

- **Aligning text is simple!**
- **text-align**
 - **left**
 - **right**
 - **center**
 - **justify**

Alignment

Here is a paragraph about alignment. You can use left, right, and justify. You can't tell the difference between left and justify unless you have at least a few lines of text.

Here is a paragraph about alignment. You can use left, right, and justify. You can't tell the difference between left and justify unless you have at least a few lines of text.

Here is a paragraph about alignment. You can use left, right, and justify. You can't tell the difference between left and justify unless you have at least a few lines of text.

line-height

- **As you can guess, doesn't affect font**
- **Adjusts the space between the lines of text**

```
h1 {  
    line-height: 50%;  
}  
  
h1 {  
    line-height: 200%;  
}
```


Review

- **The number of options for styling text can seem overwhelming.**
- **Practice on toy problems!**
- **Design larger projects on paper first!!!**

© Colleen van Lent
University of Michigan
School of Information

Unless otherwise noted, this work is licensed under the
CC BY-NC 4.0 license.

Colors

Using the Best Colors for Your Site

Color Conventions

- Color names (blue, etc.) work, but aren't consistent
- Hexadecimal is common convention
 - #0000FF, #FF0000, #FFFF00
- rgb
 - (0, 0, 1), (1, 0, 0), (1, 1, 0)
- rgba
 - (0, 0, 1, .5)

#0000FF

#FF0000

#FFFF00

rgb(0,0,1)

rgba(0, 0, 1, .5)

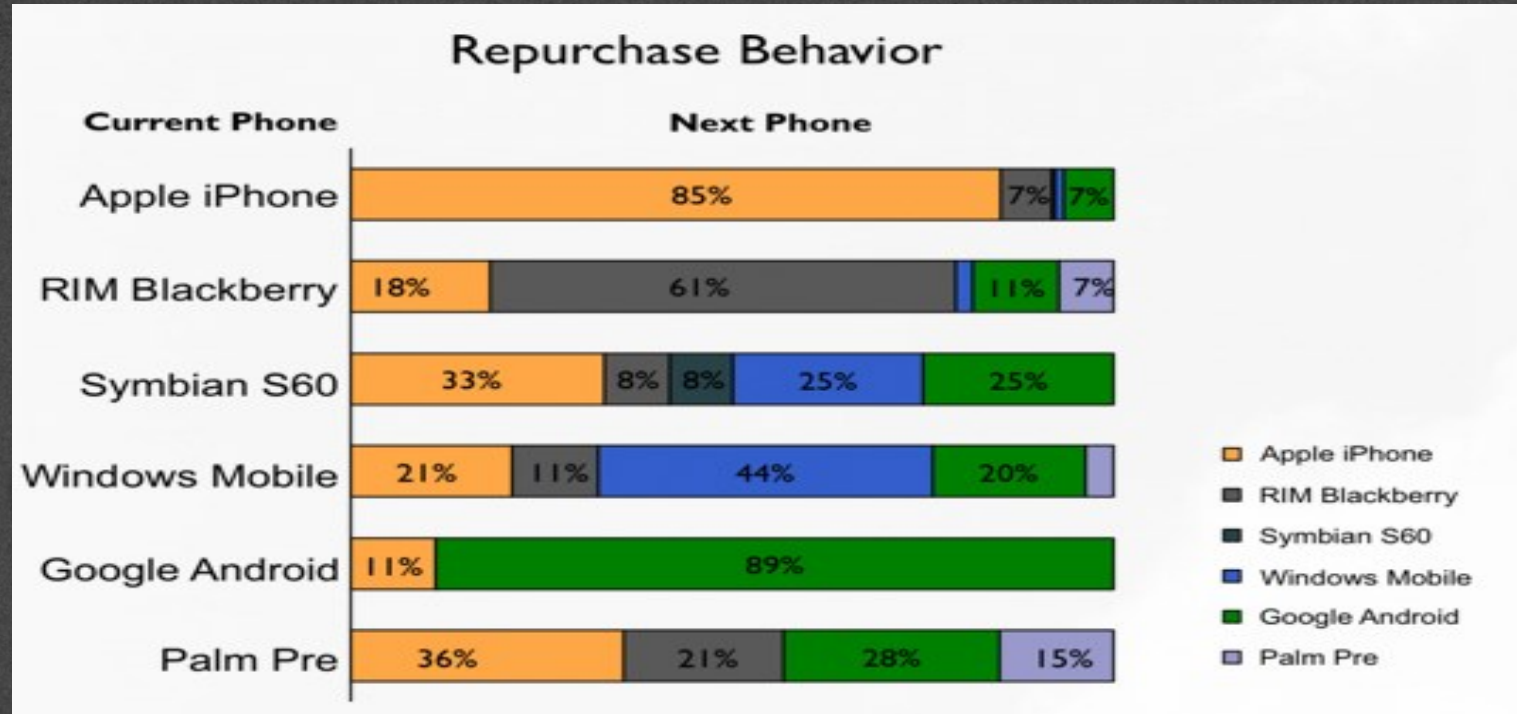
Accessibility

- **Appropriate use of color is critical to web accessibility**
- **Many more people are visually impaired or color blind than are legally blind**

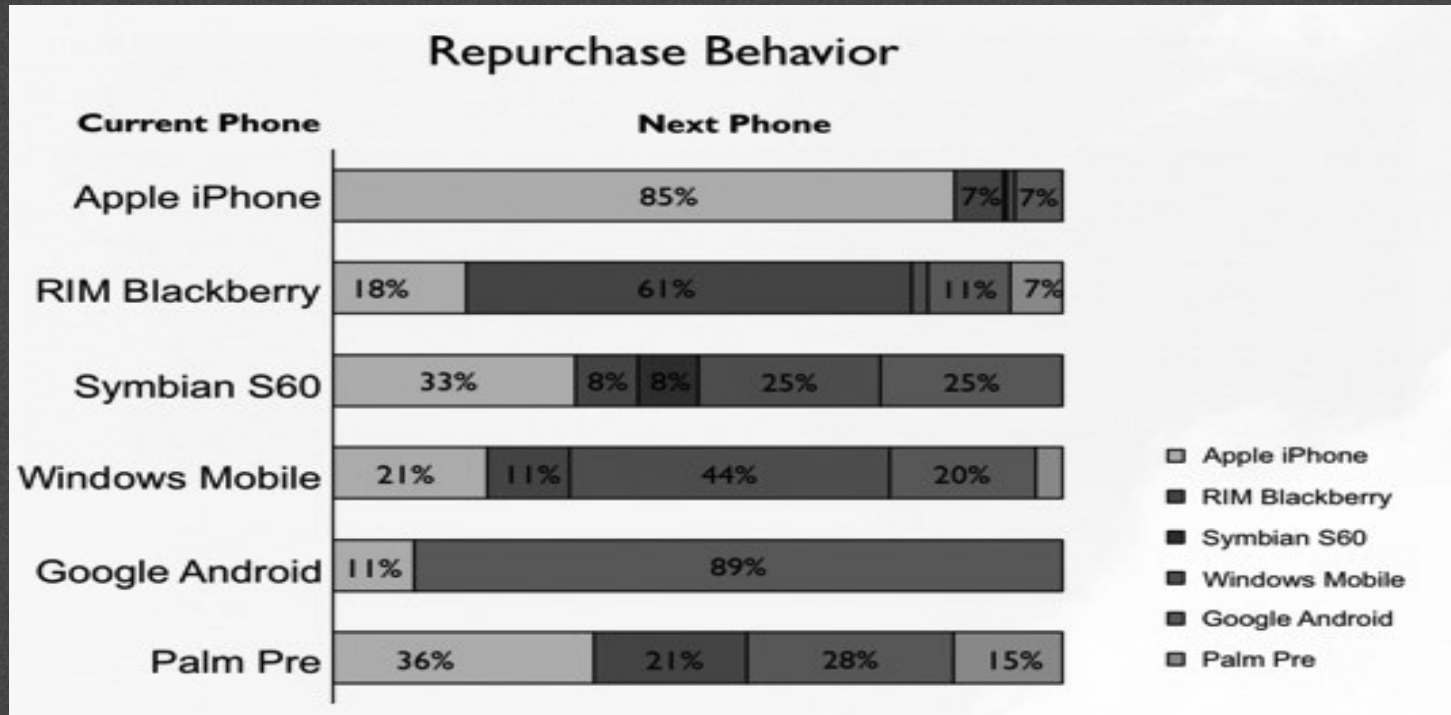
What is color contrast?

- You may not intuitively know when something has poor contrast
- There are tools that quantify the contrast between text and its background
 - [WAVE Web Accessibility Evaluation Tools](#)
 - [WebAIM: Contrast Checker](#)

Don't use color alone to convey meaning



Test in gray scale ...



Review

- **Use web safe colors and use an accepted convention**
- **Test your site using a contrast checker**
- **Avoid using color to convey meaning**

© Colleen van Lent
University of Michigan
School of Information

Unless otherwise noted, this work is licensed under the
CC BY-NC 4.0 license.

Display and Visibility

Display is Key to Layout

- **Every element is a box**
- **Display affects the layout of neighboring elements**

Common Values

- **inline:** sits next to other elements
 - takes up “just enough” width and height
- **block:** forces line break
 - default: take up all horizontal width and “just enough” height
 - rules can set height and width

Common Values

- **inline-block:**
 - same as inline, but accepts height and width
- **none: removed from page**
 - **Still in DOM, but not visual (even to SRs)**

Complementary Properties

- **float**
 - Reposition elements to the right or left.
 - Elements are aware of one another and will not overlap.
 - Values: left, right
- **clear**
 - Used to keep floating elements away
 - Values: left, right, both

Element Overflow

- **What happens when you set a height/width and the content doesn't fit any longer?**
- **Use overflow to determine access**

Overflow

- **visible:** Can cause text to show up “on top” of other text
- **hidden:** Hides anything that goes beyond bounding box
 - This can cause problems since if the user increases font size, they may not be able to see content
- **scroll:** Gives horizontal and vertical scrollbars
- **auto:** Adds scrollbars as needed

Other Display Values

- **New display properties are available, but not always supported:**
 - **Table**
 - **Grid**
 - **Flexbox**

display:table

- Sometimes you want to have table-like layout without using table structure, use **display:table** along with **display:table-cell** for elements.

Visibility

- Specifies whether or not element is visible
- Options include:
 - visible
 - hidden
 - collapse (only for table elements)
- Unlike `display:none` a hidden element is still part of the **DOM** and still takes up space

Review

- **Display is just one tool for positioning our elements on the page**
- **Early design will make the coding easier**
- **Utilize tools to see the different options**

© Colleen van Lent
University of Michigan
School of Information

Unless otherwise noted, this work is licensed under the
CC BY-NC 4.0 license.

Homework One

Adding Simple Style

Homework One

- Your final project will be a site that you create.
- But for the next few weeks you will style a site that I have begun for you.
- The goal is to practice a few different concepts each week.

Requirements

1. Copy ("fork") the [code that I have for you](#).
2. Modify all three HTML files to successfully link to the style.css file.
3. Create a css file named style.css and add rules.
4. Host and share your site.
5. Peer review sites.

Example

- Starting site
- Example ending site

Selectors

- **body**
- **header**
- **nav**
- **main**
- **footer**
- **li**
- **h1**
- **p**

Properties

- **font-family**
- **font-size**
- **background-color**
- **display**
- **width**
- **text-align**
- **color**
- **line-height**

Example

- <https://replit.com/@WD4E-CSS/ProjectStarter>

Share Your Site

- If you are using Replit you can share your site easily.
- If not, you can utilize some of the optional videos to learn about other ways to host your site.

Share Your Site

- If you are using Replit you can share your site easily.
- If not, you can utilize some of the optional videos to learn about other ways to host your site.

Peer Reviews

- You will need to do peer reviews to receive credit for this assignment.
- The expectation is that everyone will participate in this assignment.
- If you are not a paid learner, consider putting the link to your site in a forum.

Remember!!

- Learning a new skill such as programming is primarily about practice, practice.
- Use w3schools and other tutorials as a reference.
- Use Inspect Element to check each element.
- Have some fun with it – we tend to be easy graders. 😊

© Colleen van Lent
University of Michigan
School of Information

Unless otherwise noted, this work is licensed under the
CC BY-NC 4.0 license.