

# CSS Styles (WIP)

CS/IT 490 WD, Fall 2013

Last update 2013-09-03

Written by Rachel J. Morris  
Creative Commons Attribution-NonCommercial-ShareAlike  
3.0 Unported License

# Breakdown

- Style Notes
- Colors
- Text
- Dimensions
- Layout
- Miscellaneous

# Style Notes

- As we go through different styles, you'll notice that some styles can be broken up or shoved together.
- For example, you could specify a **background image** and **background color** with the “background:” style, or specify them separately with “background-image” and “background-color”



# Background

- Properties of a Background:
  - Image
  - Position
  - Repeating
  - Size
  - Color

# Background

```
.myItem { background-image: url( "path" ); }
```

Specify a path to a background image, relative to the .css file.

# Background

```
.myItem { background-position: center center; }
```

The position can contain (x,y) coordinates as units, x% y% shifted, or simple values like “left top”, “center center”, “right bottom”.



# Background

```
.myItem { background-position: 20px 40px; }
```

(X,Y) Coordinates:

- This can be in pixels (px)
- This can be in em (unit of font size)
- This can be % of the page
- This can be inches/centimeters (in, cm)

# Background

```
.myItem { background-position: left top; }
```

If you use the shortcut positions:

- Horizontally, you can have:
  - left, center, right
- Vertically, you can have:
  - top, center, bottom



# Background

```
.myItem { background-repeat: no-repeat; }
```

Whether or not a background image repeats.  
You can set this to:

- repeat (both directions)
- repeat-x (repeat horizontally only)
- repeat-y (repeat vertically only)
- no-repeat (no repeating)

# Background

```
.myItem { background-size: 100%; }
```

Allows you to scale the background image.

- Can specify x,y separately, or together. If only one number is given, then  $x = y = \text{value}$ .
- Setting it to “cover” means the image may be cut off but it will scale the full width.
- Setting it to “contain” means the full image will be shown. (Scaled for height to fit)



# Background

```
.myItem { background-color: rgba( 0, 0, 0, 0.5 ); }
```

Allows you to specify the background color.

- Can use hex code: #FF00FF
- Can specify RGB: ( 0-255, 0-255, 0-255 );
- Can specify RGBA ( 255, 255, 255, 0.5 );
  - Note the last number is a float between 0 and 1 to specify transparency/alpha.



# Background

```
.myItem {  
background: url( "img.png" ) #FF00FF no-repeat;  
}
```

Specify multiple background properties all at the same time.

# Border

- Properties of a Border:
  - Border color
  - Border width
  - Border style
  - Border radius
- You can specify these as separate edges (i.e., border-left-color) or all at once (i.e., border-color, border)

# Border color

```
.myItem { border-color: #000000; }
```

Set the border's color. Use hex, RGB, or RGBA. Can specify different edges:

```
.myItem {  
border-left-color: #FF0000;  
border-right-color: #00FF00;  
border-top-color: #0000FF;  
border-bottom-color: #FF00FF;  
}
```

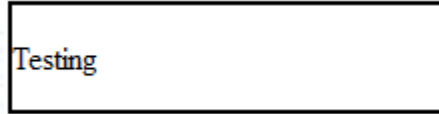




# Border width

```
.myItem { border-width: 2px; }
```

Sets thickness of the border.

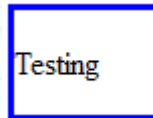


# Border style

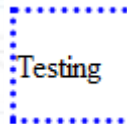
```
.myItem { border-style: solid; }
```

Can choose different border styles:

- Solid



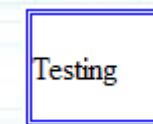
- Dotted



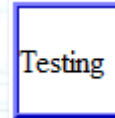
- Dashed



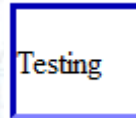
- Double



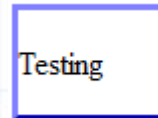
- Ridge



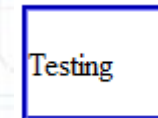
- Inset



- Outset



- Groove



# Border radius

```
.myItem { border-radius: 15px; }
```

Can make the border have rounded corners  
(CSS3 feature)





# Border

```
.myItem { border: solid 2px #000000; }
```

Can specify multiple properties at once!



Testing

# Color

```
.myItem { color: #AAFFEE; }
```

The Color property changes text color.

- Hex, RGB, RGBA

# Opacity

- Can set an element to be semi-transparent.
- Note that this will cause all children to also be semi-transparent.
- If you just want a semi-transparent background, set the color with RGBA.



# Opacity

- Can set an element to be semi-transparent.
- Note that this will cause all children to also be semi-transparent.
- If you just want a semi-transparent background, set the color with RGBA.

# Opacity

```
div.one {  
background: rgba( 0, 0, 0, 0.5 );  
color: rgb( 255, 0, 255 );  
}
```



Background is  
semi-transparent but text  
is completely magenta.

```
div.two {  
background: rgb( 0, 0, 0 );  
color: rgb( 255, 0, 255 );  
opacity: 0.5;  
}
```



Text is semi-transparent

# Shadows

- Text Shadow
- Box Shadow



# Text Shadow (CSS3)

```
.myItem { text-shadow: -2px 4px 5px #000000; }
```

- Adds shadowing to text
- Shadow horizontal placement
- Shadow vertical placement
- Blur amount (px)
- Color

Testing

Testing

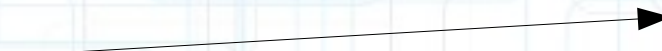
Testing

Testing

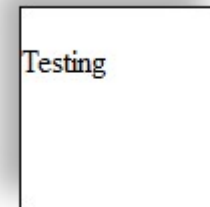
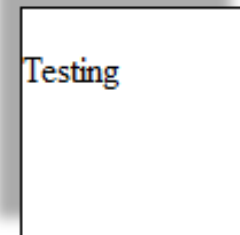
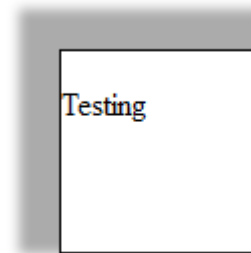
# Box Shadow (CSS3)

```
.myItem { box-shadow: -2px 4px 5px #000000; }
```

- Create a shadow around a box object
- Horizontal Placement
- Vertical Placement
- Blur
- Spread
- Color



Spread = 10px



# Text

- Font Family
- Font Size
- Text Decoration
- Font Style
- Font Weight
- Line Height
- Letter Spacing
- Word Spacing
- Direction
- Text Align
- Text Indent
- Text Outline



# Font Family

```
.myItem { font-family: "arial"; }
```

- Sets the font of the text. Can set to several values in case the web browser doesn't support one chosen.
- Set by family-name or generic-name:
  - “arial” is a family-name
  - “sans-serif” is a generic-name

# Font Family

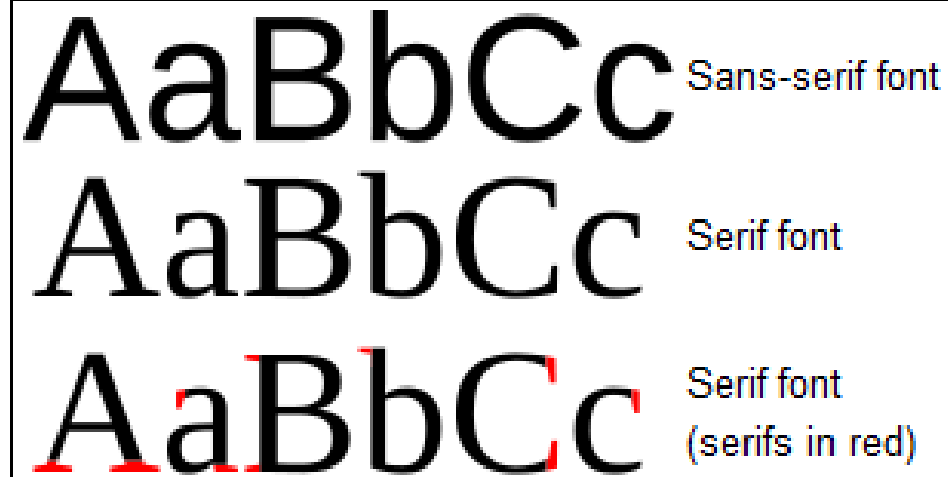
```
.myItem { font-family: "arial", sans-serif; }
```

- Generic Names:
  - serif, sans-serif, cursive, fantasy, monospace
- Web-safe Fonts:
  - Georgia, Times, Arial, Helvetica, Comic Sans (NO), Impact, Tahoma, Verdana, Courier New, Ludica Console
  - More listed here:  
[http://www.w3schools.com/cssref/css\\_websafe\\_fonts.asp](http://www.w3schools.com/cssref/css_websafe_fonts.asp)

# Font Family

```
.myItem { font-family: "arial"; }
```

- Serif vs. Sans-Serif (From Wikipedia)



AaBbCc Sans-serif font

AaBbCc Serif font

AaBbCc Serif font (serifs in red)

The image shows three lines of text within a black-bordered box. The first line displays 'AaBbCc' in a sans-serif font, with the label 'Sans-serif font' to its right. The second line displays 'AaBbCc' in a serif font, with the label 'Serif font' to its right. The third line displays 'AaBbCc' in a serif font, with the labels 'Serif font' and '(serifs in red)' to its right. In this third line, the serifs on the letters 'A', 'a', 'B', 'b', 'C', and 'c' are highlighted in red.



# Font Family

`.myItem { font-family: "arial"; }`

- <http://www.google.com/fonts>
- Has special fonts you can use on your webpage – gives you a .css file to link to and a family name to use.

```
<link href='http://fonts.googleapis.com/css?family=Kavoon' rel='stylesheet' type='text/css'>
```

```
font-family: 'Kavoon', cursive;
```

**Crumpy wizards make toxic brew j**

# Font Size

```
.myItem { font-size: 20px; }
```

- Set the font size
  - Length in pixels (px), cm, ...
  - % - percent of parent's size
  - small, medium, large, ...

# Text Decoration

```
.myItem { text-decoration: underline; }
```

- Various line properties
  - none
  - underline
  - overline
  - line-through



# Font Style

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

- Font styles:
  - normal
  - italic
  - oblique

# Font Weight

```
.myItem { font-weight: bold; }
```

- Adds boldness
  - normal
  - bold
  - older
  - lighter
  - 100 – 900 (thin – thick)

# Line Height

```
.myItem { line-height: 20px; }
```

- Sets the height of the line
  - 2 – will double the font-size
  - 2px – will be 2 pixels high
  - 2% - will be 2% of the current font size

Testing

Testing



# Letter Spacing

```
.myItem { letter-spacing: 5px; }
```

- Sets the spacing between letters
  - Specify either *normal* or a length in pixels/cm
  - Negative values are allowed

Testing

Testing

Testing

# Word Spacing

```
.myItem { word-spacing: 5px; }
```

- Set the spacing between words
  - *normal* or length in px, cm, etc.

One Two Three

normal

One Two Three

5px

OneTwoThree

-5px

# Direction

```
.myItem { direction: 5px; }
```

- Set text-flow direction
  - ltr – Left to right
  - rtl – Right to left

One Two Three

One Two Three



# Text Align

```
.myItem { text-align: center; }
```

- Set horizontal alignment
  - left, right, center, justify

One Two Three

One Two Three

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec a diam lectus. Sed sit amet ipsum mauris. Maecenas congue ligula ac quam viverra nec consectetur ante hendrerit. Donec et mollis dolor.

# Text Indent

```
.myItem { text-indent: 50px; }
```

- Affects amount of indent

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec a diam lectus. Sed sit amet ipsum mauris. Maecenas congue ligula ac quam viverra nec consectetur ante hendrerit. Donec et mollis dolor.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec a diam lectus. Sed sit amet ipsum mauris. Maecenas congue ligula ac quam viverra nec consectetur ante hendrerit. Donec et mollis dolor.

# Dimensions

```
.myItem { width: 100px; height: 50px; }
```

- Set an object's width and height

```
.myItem { min-width: 100px; min-height: 50px; }
```

- Set an object's minimum width/height

```
.myItem { max-width: 100px; max-height: 50px; }
```

- Set an object's maximum width/height



# Dimensions

- Setting the width/height does not take into account padding or the borders – if you add these, your item's dimensions will increase!
- Width can be % (relative to its parent), a length in px, cm, or set to *auto*.

# Dimensions

- Different elements have different *display* types and may not be affected.
- For example, using the *width* style on a span tag `<span>`
  - Unless you change the *display* of the span tag, width won't do anything.



# Types of Display

Elements have different categories they fit in for how they behave:

- none – does not display
- inline – does not have its own line by default (no line-breaks before or after)
- block – does not allow elements next to it.
- There are more types, but these are the main ones.  
[http://www.w3schools.com/cssref/pr\\_class\\_display.asp](http://www.w3schools.com/cssref/pr_class_display.asp)



# Display

```
.myItem { display: block; }
```

- Set the display of an element

Paragraph



- Paragraph wants its own line, but two images sit next to each other if we don't add line breaks (<br/>).

# Display

- Say that we want to center an image on the page.
  - First we need to set the display to block.

Paragraph





# Display

- Then we need to set the margins (we haven't gone over this yet).

Paragraph





# Display

- Without setting display: block, it wouldn't work!

Paragraph

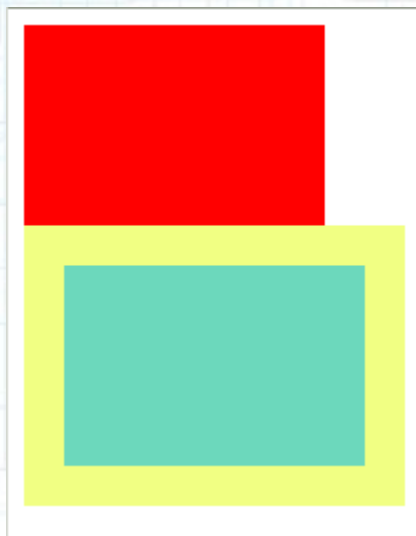


# Layout

- Margins
- Padding
- Float
- Clear
- Position
- Top, Left, Bottom, Right
- Z-Index

# Margins

- Margins affect the area **outside** of an element.
- The margin doesn't affect the element's internal size.



← The yellow here shows the margin



# Margins

- You can set every direction at once, vertical and horizontal separately, or top/right/bottom/left in one.

```
.myItem { margin: 20px; }
```

All directions

```
.myItem { margin: 20px 10px; }
```

Vertical is 20,  
Horizontal is 10

```
.myItem { margin: 2px 3px 4px 5px; }
```

Top is 2, right is  
3, bottom is 4,  
and left is 5

# Margins

- You can also set them separately

```
.myItem { margin-left: 20px; }
```

```
.myItem { margin-right: 20px; }
```

```
.myItem { margin-top: 20px; }
```

```
.myItem { margin-bottom: 20px; }
```



# Margins

- Margins can be:
  - Length (px, cm, etc.)
  - Percent, with respect to parent element
  - Auto, automatically places it.
- If you set **left** and **right** margins to **auto**, and your element has a **block** display, it will be centered on the page!
- I know we all miss the <center> tag, but it's deprecated now so don't use it!



# Margins

- Centering an element:

```
div { width: 150px; height: 100px; }
```

Set the CSS

```
div.one { background: #ff0000; }
```

```
div.two { background: #00ff00; margin: 5px auto; }
```

---

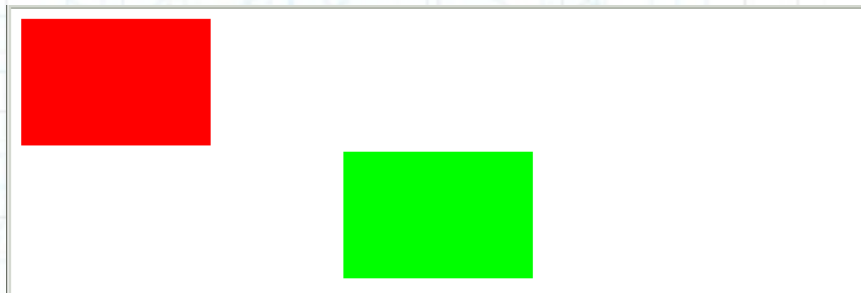
```
<body>
```

```
  <div class="one"></div>
```

```
  <div class="two"></div>
```

```
</body>
```

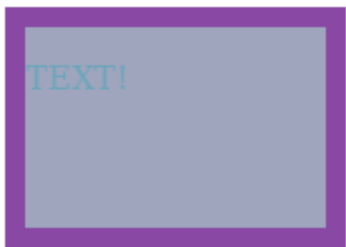
---



Result

# Padding

- Padding affects the *inside* of your element, and adds spacing.
- This *will* affect the width of your element, so beware on pages with very exact positioning!



The border is the padding here.

No padding



Padding – box is bigger, text doesn't stick to the corner



# Clear

- You can use the Clear style to enforce no elements are neighboring in a certain direction...
  - clear: left;
  - clear: right;
  - clear: both;
  - clear: none;



# Float

- Float allows you to stick an object to the *left* or *right* side.
- This also makes them stack against each other horizontally.
- This can cause rendering problems on your webpage! This is why “clearfix” exists!

# Float

- Floats are really useful:

```
.three { float: left; background: #a0abff; }  
.four { float: left; background: #ffc4a0; }  
.five { float: right; background: #ffa0f3; }
```



# Float

- But can cause problems:

## **My Webpage**

### **Navigation:**

- ITEM1
- ITEM2
- ITEM3

### **Stuff:**

I like turtles

Gee this page is swell. But the navigation takes up a lot of vertical space. Let's make those List Items float left!



# Float

- But can cause problems:

```
ul li { float: left; }
```

```
<h2>Navigation:</h2>  
<ul>  
  <li><p>ITEM1</p></li>  
  <li><p>ITEM2</p></li>  
  <li><p>ITEM3</p></li>  
</ul>
```

Now I should have a nice,  
horizontal nav bar!

# Float

- But can cause problems:

## My Webpage

### Navigation:

- ITEM1ITEM2ITEM3

### Stuff:

I like turtles

What the heck?!

Our floating List Items are contained within an Unordered List, but mess up the markup position coming afterwards!

# Float Fix #1

- Fix #1: create a “clear” class:

```
<style type="text/css">
  ul li { float: left; }

  .clear { clear: both; }
</style>

<h2>Navigation:</h2>
<ul>
  <li><p>ITEM1</p></li>
  <li><p>ITEM2</p></li>
  <li><p>ITEM3</p></li>

  <div class="clear"></div>
</ul>
```

Add an element with the “clear” class after any floating items.

## My Webpage

### Navigation:

- ITEM1ITEM2ITEM3

### Stuff:

I like turtles

Now it's OK



# Float Fix #2

- Fix #2: Google “clearfix”



APRIL 21, 2011

## A new micro clearfix hack

The clearfix hack is a popular way to contain floats without resorting to using presentational markup. This article presents an update to the clearfix method that further reduces the amount of CSS required.

<http://nicolasgallagher.com/micro-clearfix-hack/>

```
/**
 * For modern browsers
 * 1. The space content is one way to avoid an Opera bug where
 *    contenteditable attribute is included anywhere else in the page
 *    Otherwise it causes space to appear at the top and bottom of elements
 *    that are clearfixed.
 * 2. The use of `table` rather than `block` is only necessary if the
 *    element is not a table
 * `:before` to contain the top-margins of child elements.
 */
.cf:before,
.cf:after {
  content: " "; /* 1 */
  display: table; /* 2 */
}

.cf:after {
  clear: both;
}
```

Put the code into your  
CSS file

# Float Fix #2

- Fix #2: Google “clearfix”

```
.cf:before,  
.cf:after {  
    content: " "; /* 1 */  
    display: table; /* 2 */  
}  
  
.cf:after {  
    clear: both;  
}
```

Add the Clearfix CSS

```
<!-- Fix 2 -->  
<ul class="cf navList">  
    <li><p>ITEM1</p></li>  
    <li><p>ITEM2</p></li>  
    <li><p>ITEM3</p></li>  
</ul>
```

Add the “cf” class to the  
**Container Element**

## My Webpage

### Navigation:

- ITEM1ITEM2ITEM3

### Stuff:

I like turtles



# Position

- Elements can be positioned according to different rules:
  - Static (default)
  - Absolute
  - Fixed
  - Relative



# Position

- If you need to review in the future, this tutorial is really helpful:
- <http://www.barelyfitz.com/screencast/html-training/css/positioning/>

# Position

```
.myItem { position: static; }
```

- Static Positioning
  - Elements are rendered in order of the markup.

# Position

```
.myItem { position: absolute; }
```

- Absolute Positioning

- Elements are rendered at your specified coordinates relative to any non-static parents.
- Specify left, right, top, or bottom to position.



# Position

```
.myItem { position: fixed; }
```

- Fixed Positioning
  - Position is fixed at the same position, no matter the scrolling.
  - Is with respect to the browser window.

# Position

```
.myItem { position: relative; }
```

- Relative Positioning

- The element is positioned relative to its default position.
- Can specify left, right, top, or bottom.
- So new position would be  $x_{\text{default}} + \text{left}$

# Top, Left, Bottom, Right

- Specify locations for non-static elements.

```
.myItem { top: 5px; }
```

```
.myItem { bottom: 5px; }
```

```
.myItem { left: 5px; }
```

```
.myItem { right: 5px; }
```



# Z-Index

- Z-index allows you to specify which elements are ahead of or behind others.
- It is the “Z-axis”.
  - But this is *index*, so you only specify an ordering number.
  - The higher the number, the more on-top it is.

```
.myItem { z-index: 3; }
```

# List Style

- You can remove or changed the bulleted list appearance with list-style.
  - list-style-type
    - Circle, square, katakana
    - View full list here:  
[http://www.w3schools.com/cssref/pr\\_list-style-type.asp](http://www.w3schools.com/cssref/pr_list-style-type.asp)
  - list-style-position
    - inside or outside, based on the margin
  - list-style-image
    - Specify a url for the list bullets.



# List Style

- Can set multiple items with just list-style

```
.myItem { list-style: url( "bullet" ) inside; }
```



# Cursor

```
.myItem { cursor: pointer; }
```

- Can change what the mouse looks like when it hovers over the element.
  - Auto (default)
  - Crosshair
  - Pointer
  - Wait
  - See full list here:

[http://www.w3schools.com/cssref/pr\\_class\\_cursor.asp](http://www.w3schools.com/cssref/pr_class_cursor.asp)

# Overflow

```
.myItem { overflow: hidden; }
```

- When you have a container element that is a fixed size, text within may overflow.
- You can specify whether to allow it to scroll horizontally, vertically, both, or just hide any overflowing data.



# Overflow

```
.myItem { overflow: hidden; }
```

- There is overflow, overflow-x, and overflow-y.
- They can have the values:
  - visible (not clipped, renders outside the box)
  - hidden (clipped)
  - scroll
  - auto (only add scrollbar on overflow)



# Overflow

```
.myItem { overflow: hidden; }
```

```
.myItem { overflow-x: visible; }
```

```
.myItem { overflow-y: auto; }
```

# References

CSS Reference, W3Schools.com  
<http://www.w3schools.com/cssref/>

Sans-serif, Wikipedia.org  
<http://en.wikipedia.org/wiki/Sans-serif>