

Slides: <http://bit.ly/slidehtml4>

# HTML/CSS Class 4: Techniques, Tips & Tricks

Alexis Goldstein

@alexisgoldstein

[alexis@alexisgo.com](mailto:alexis@alexisgo.com)

**Review (Last Chance!)**

# Review: Jargon so far

Let's review the terms & jargon we've learned thus far:

- HTML terms:
  - **Tag**
  - **Element**
  - **Attribute**
- CSS terms:
  - **Element Selector**
  - **Class Selector**
  - **Id Selector**
  - **Pseudoclasses**

# A Super Fast Quiz!

```
<html>
```

```
<head>
```

```
  <style>
```

```
    .kittenPics
```

```
    {
```

```
      border: 2px solid pink;
```

```
    }
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  
```

Is this a:

A: ID Selector

B: Element  
Selector

C: Class Selector

?

# A Super Fast Quiz!

```
<html>
```

```
<head>
```

```
  <style>
```

```
    .kittenPics
```

```
    {
```

```
      border: 2px solid pink;
```

```
    }
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  
```

Answer: C! Class Selector!

# A Super Fast Quiz!

```
<html>
<head>
  <style>
    .kittenPics
    {
      border: 2px solid pink;
    }
  </style>
</head>
<body>
  
```

Is this a:  
A: Property  
B: Pseudoclass  
C: Attribute  
?

# A Super Fast Quiz!

```
<html>
<head>
  <style>
    .kittenPics
    {
      border: 2px solid pink;
    }
  </style>
</head>
<body>
  
```

Answer: C,  
Attribute!

# Review: Topics so far

- CSS:
  - **Margin and Padding**
  - **Borders**
  - **Float**
  - **Positioning:**
    - **Static**
    - **Fixed**
    - **Relative**
    - **Absolute**

# A Super Fast Quiz!

```
<html>
<head>
  <style>
    #centerMe
    {
      ??? : ??? ;
      margin: 0px auto;
    }
  </style>
</head>
<body>
  <div id="centerMe">I should be centered on the
page!</div>
```

To position a div in the middle of the page, we can add **margin: 0 auto;** but we also need to specify *one more property!*

Is that property:

A: position

B: width

C: float

# A Super Fast Quiz!

```
<html>
<head>
  <style>
    #centerMe
    {
      ??? : ??? ;
      margin: 0px auto;
    }
  </style>
</head>
<body>
  <div id="centerMe">I should be centered on the
page!</div>
```

Answer:  
B: width

# Float and Clear

# Styling Images with Float

- With CSS float, an element can be pushed to the left or right, allowing other elements to wrap around it. When an element is set to float, text and other content will flow around the floated element.

The **float** property specifies whether or not an element should float. It also specifies which direction it should float (left, right). Example:

```
.alignLeft  
{  
  float: left;  
}
```

# Styling Images with Float

This is most commonly used with images, in order to align them left or right so text flows around an image. It is also useful when working with layouts.

Let's give this a try:

1. [http://www.w3schools.com/css/tryit.asp?filename=trycss\\_float](http://www.w3schools.com/css/tryit.asp?filename=trycss_float)
2. [http://w3schools.com/css/tryit.asp?filename=trycss\\_float\\_elements](http://w3schools.com/css/tryit.asp?filename=trycss_float_elements)

# Styling Images with Float: Lab

**<http://jsfiddle.net/fiddlefiddle/rFjRq/>**

**Completed version: <http://jsfiddle.net/rFjRq/11/>**

# Using clear

The clear property specifies which sides of an element where other floating elements are not allowed.

Best described visually! See this in action: [http://www.w3schools.com/css/tryit.asp?filename=trycss\\_float\\_clear](http://www.w3schools.com/css/tryit.asp?filename=trycss_float_clear)

# Print Stylesheets

- Last week, I created a pretty colorful page.
- What happens if I try and print a page with so many background colors?
- My visitors will curse me!

# Print Stylesheets

- We can create a separate stylesheet *just for printing* by using a new attribute, the **media** attribute.
- It works by adding a second **link** element to your head section:

```
<head>
  <meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
  <title>New Web Project</title>
  <link rel="stylesheet" href="style.css">
  <link rel="stylesheet" media="print" href="print.css">
</head>
```

# Print Stylesheets

- What do we do in the print.css stylesheet?
- For one, we can remove ALL background colors with this one simple line: **background: white;** if we put it in the body selector:

```
body {  
    background: white;  
}
```

# Print Stylesheets

- We may also want to completely remove our menu bar, which is currently in the footer.
- We can do that by leveraging the **display** property:

```
#footer  
{  
  display: none;  
}
```

# Print Stylesheets

- Want more tips and tricks on what to add to your print.css stylesheet?
- Check out the following article on alistapart.com: [http://  
www.alistapart.com/articles/  
goingtoprint/](http://www.alistapart.com/articles/goingtoprint/)

# Liquid vs Fixed

## **Fixed Layout:**

In a Fixed Layout, the columns are set to a specific width: 500 pixels total (by total, I mean if you add up the widths of all the columns), 750 pixels total, 900 pixels total, etc.

If you resize the browser on a fixed layout page, the columns will stay the same size.

## **Liquid Layout:**

In a Liquid Layout, instead of using pixels to set a specific width, the columns change sizes as you adjust the browser size.

One way to do this is with percentages. The left column could be 20% of the page, the middle column 50% and the right column 30%, for example.

**To read more:** <http://green-beast.com/blog/?p=199>

# Further Reading

Samples of just about every layout you can imagine:

- <http://layouts.ironmyers.com/>
- <http://matthewjamestaylor.com/blog/perfect-3-column.htm>

Web Grids - Column-based Layouts:

- [http://webdesign.about.com/od/layout/ss/web\\_grids.htm](http://webdesign.about.com/od/layout/ss/web_grids.htm)

Fixed-width Layouts Versus Liquid Layouts:

- <http://webdesign.about.com/od/layout/i/aa060506.htm>

# Changing our Fixed Layout to Liquid

- Let's assume we have a three-column layout that uses Absolute Positioning in CSS, which we reviewed last week.
- Let's take a version of that file, and convert it to be a liquid layout instead of a fixed layout.

# Changing our Fixed Layout to Liquid

- We'll use this JSFiddle as our starting point: <http://jsfiddle.net/GzwVb/1/>
- Finished file: <http://jsfiddle.net/GzwVb/5/>

# Changing our Fixed Layout to Liquid

- Changing the CSS:
  - Make the header and footer both have a **100% width**

```
#header, #footer
{
    background-color: beige;
    height: 50px;
    width: 100%;
}
```

# Changing our Fixed Layout to Liquid

- Changing the CSS:
  - #left should not be **position:absolute** anymore, and we don't need a value for **top**. Let's change #left instead to have **float:left;**

Before:

```
#left
{
  position: absolute;
  top: 65px;
}
```

After:

```
#left
{
  float: left;
}
```

# Changing our Fixed Layout to Liquid

- Changing the CSS:
  - #right should not be **position:absolute** anymore, and we don't need a value for **top**. Let's change #right instead to have **float:right;**

Before:

```
#right
{
  position: absolute;
  top: 65px;
  left: 300px;
}
```

After:

```
#right
{
  float: right;
}
```

# Changing our Fixed Layout to Liquid

- Changing the CSS:
  - Let's take out the **width, top, left and position: absolute** from the `#middle` id selector.

Before:

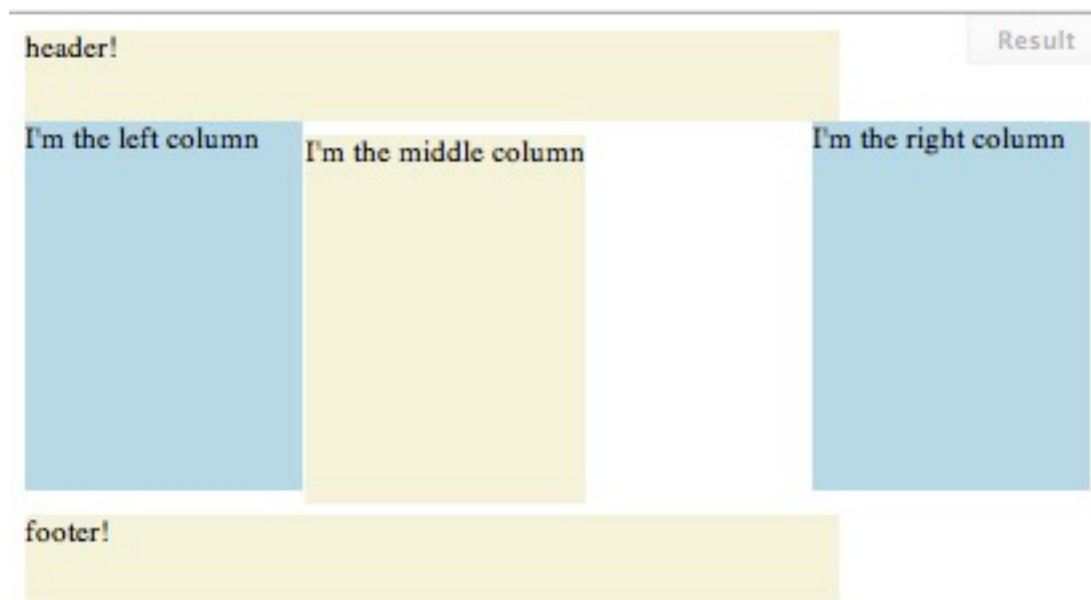
```
#middle
{
  position: absolute;
  top: 65px;
  left: 160px;
  height: 200px;
  background-color: beige;
}
```

After:

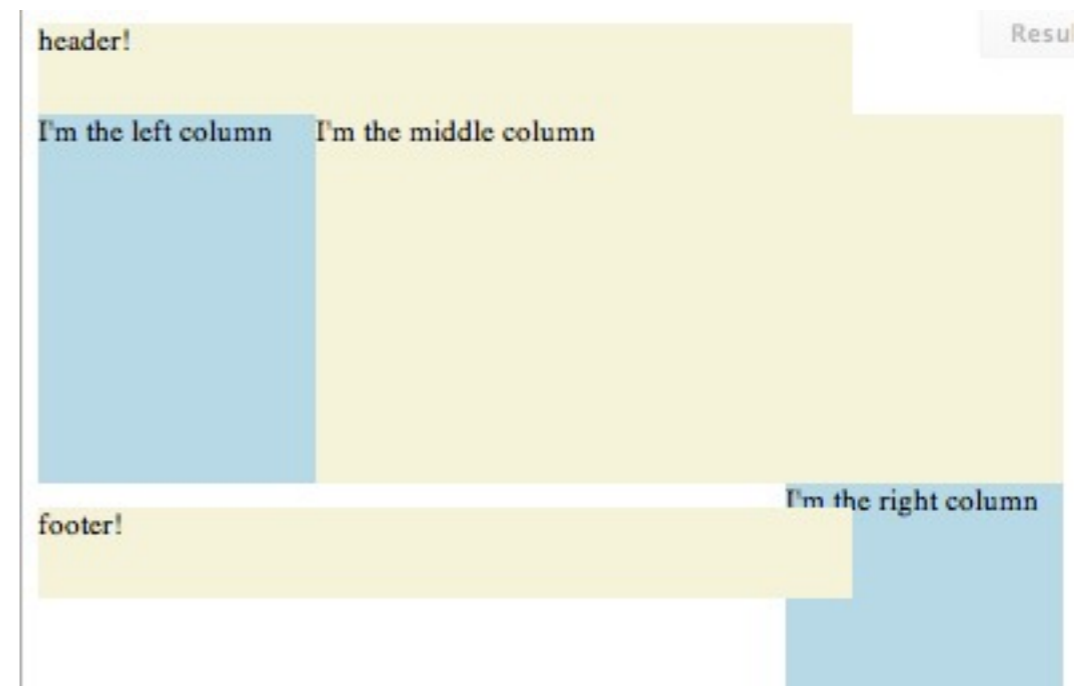
```
#middle
{
  height: 200px;
  background-color: beige;
}
```

# Whoa! What happened!

Before, mangled:



After,  
MORE mangled!:



# Changing our Fixed Layout to Liquid

- Changing the HTML:
  - The middle div no longer goes in the middle!
  - That's because when items are floated, anything you want to fill in space left behind must go ***after*** **the floated elements.**

# Changing our Fixed Layout to Liquid

- The middle div no longer goes in the middle!

Before:

```
<div id="wrapper">
  <div id="left">
    I'm the left column
  </div>

  <div id="middle">
    I'm the middle column
  </div>

  <div id="right">
    I'm the right column
  </div>
</div>
```

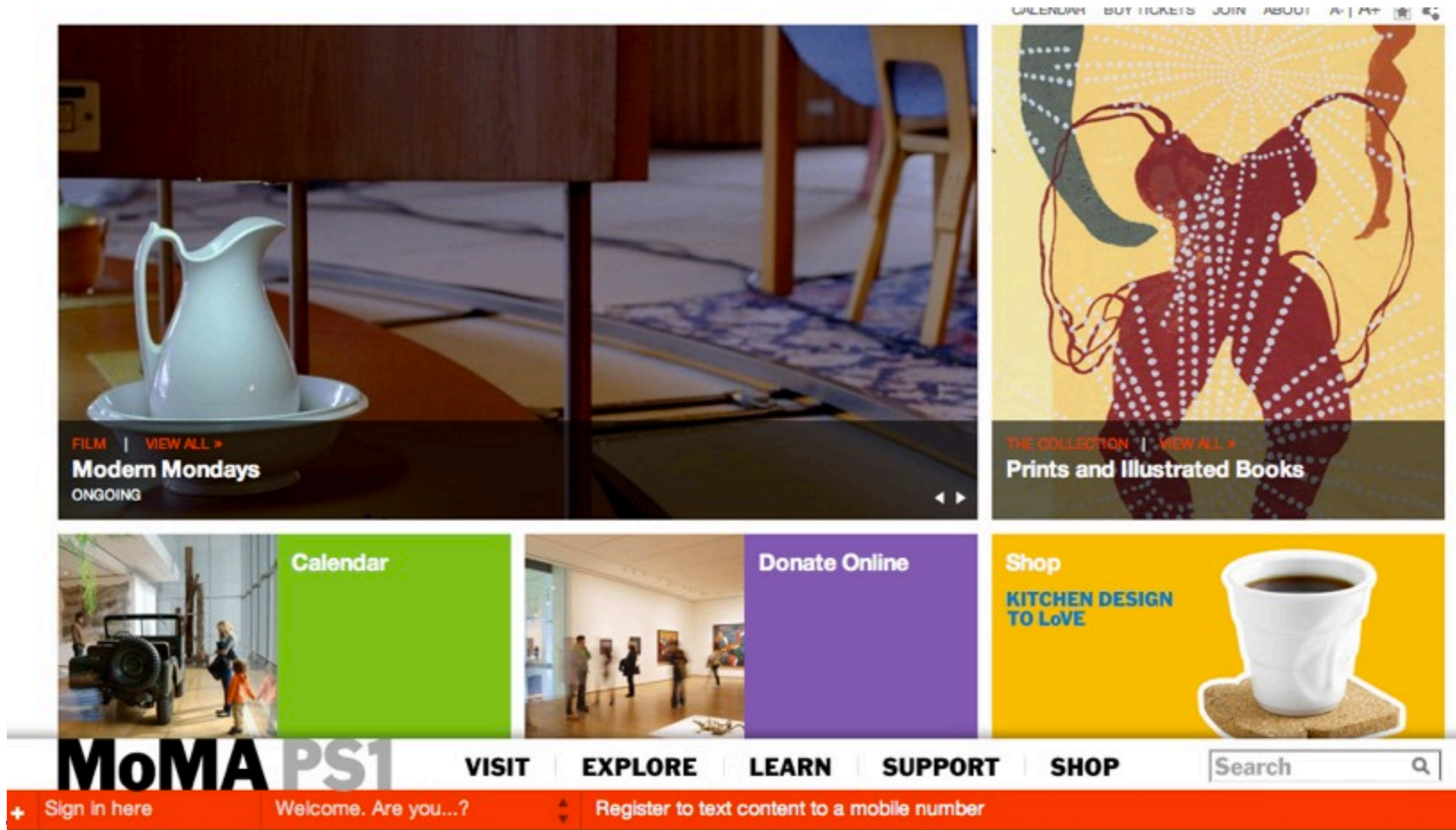
After:

```
<div id="wrapper">
  <div id="left">
    I'm the left column
  </div>

  <div id="right">
    I'm the right column
  </div>

  <div id="middle">
    I'm the middle column
  </div>
</div>
```

# Imitation is the highest form of flattery



# Imitation is the highest form of flattery

- You'll need a few images. You can download them, and the completed file (for reference!) here: <http://bit.ly/html4code>
- We'll be starting with the file `imitationIsFlatteryNoCSS.html`

# Imitation: Step one! Check out the

- The first thing we're going to style is the div whose id is set to red.
- It's almost at the bottom of the page:

- ```
<div id="red"></div>  
  
</body>  
</html>
```

# Styling the red div

- `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">`

```
<html>
  <head>
    <title>NOT THE MOMA</title>
    <style>
      #red
      {
        background-color:red;
        position:fixed;
        bottom:0px;
        left:0px;
        width:100%;
        height:30px;
      }
    </style>
  </head>
```



# Styling the navigation

The next HTML we will style is the HTML that makes up the navigation:

```
<div id="nav">
  <ul>
    <li>
      
    </li>
    <li>VISIT</li>
    <li>EXPLORE</li>
    <li>LEARN</li>
    <li>SUPPORT</li>
    <!-- take away border from the last item-->
    <li style="border:none">
  </ul>
</div>
```

- **MoMA PS1**
- VISIT
- EXPLORE
- LEARN
- SUPPORT
- SHOP

# Styling the navigation: part I

```
#nav
{
  background-color: transparent;
  background-image: url("images/bg.png");
  background-repeat: repeat-x;
  position: fixed;
  bottom: 25px;
  left: -5px;
  width: 100%;
  height: 50px;
  font-size: 20px;
  font-weight: bold;
}
```



# Styling the navigation: part 2

```
#nav ul, li
{
    /*take away the bullets*/
    list-style-type:none;
    display:inline;
    /*space out the items horizontally*/
    padding:0px 10px;

    /* we add this to move it down a little
    * without this, the list starts at the top
    * of the div. We move it 2 pixels from the
    top in the positive direction = move it down*/
    position:relative;
    top:2px;
}
```

**MoMA PS1**

**VISIT EXPLORE LEARN SUPPORT SHOP**

# Styling the navigation: part 3

```
/* just add the border to the items, not the list*/  
#nav li  
{  
  border-right: 1px solid lightgray;  
}
```

**MoMA PS1**

**VISIT**

**EXPLORE**

**LEARN**

**SUPPORT**

**SHOP**

# Styling the navigation: part 4

```
#momaLogo
{
  position:relative;
  /* the top of the div is actually where the
   * gray bleed-out starts
   * we are pushing the MOMA logo a bit down
   * from that
   */
  top:9px;
}
```

**MoMA PS1**

**VISIT**

**EXPLORE**

**LEARN**

**SUPPORT**

**SHOP**

# Imitation is the highest form of flattery

```
<body>  
  <div id="container">  
  
    <div id="bigBox">  
      <div class="invisible"></div>  
      <div class="overlay"></div>  
    </div>  
  
  </div>  
</body>
```

# Style the big div on the left

```
#container
```

```
{  
  width:1000px;  
  margin: 0 auto;  
  position:relative;  
  min-height:600px;  
}
```

```
#bigBox
```

```
{  
  width: 650px;  
  height: 300px;  
  background-color: lightblue;  
  position:absolute;  
  left: 0px;  
  top: 0px;  
}
```



**MoMA PS1**

[VISIT](#)

[EXPLORE](#)

[LEARN](#)

[SUPPORT](#)

[SHOP](#)

# Adding a semi-transparent overlay

```
.overlay
```

```
{
```

```
  position: absolute;
```

```
  bottom: 0px;
```

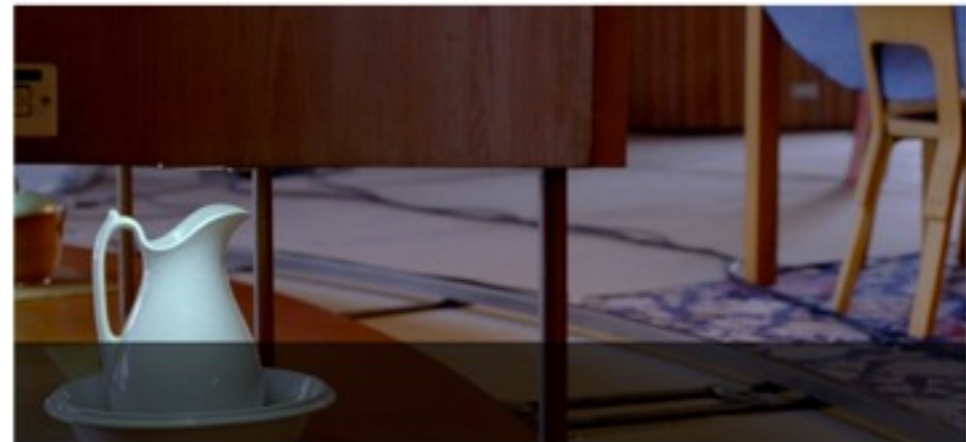
```
  left: 0px;
```

```
  width: 100%;
```

```
  background-color: rgba(0, 0, 0, 0.6);
```

```
  height: 70px;
```

```
}
```



MoMA PS1 VISIT EXPLORE LEARN SUPPORT SHOP

I'm sneaking in some CSS3;  
rgba() is a CSS3 property

```
<div id="bigBox">  
  <div class="invisible"></div>  
  <div class="overlay"></div>  
</div>
```

# Styling the invisible div

```
.invisible
{
  width: 100%;
  height: 300px;
  position: absolute;
  left: 0px;
  top: 0px;

  background-color: transparent;
}
```

# Adding the hover effect

```
.invisible:hover  
{  
  background-color: rgba(0,0,0,0.6);  
}
```

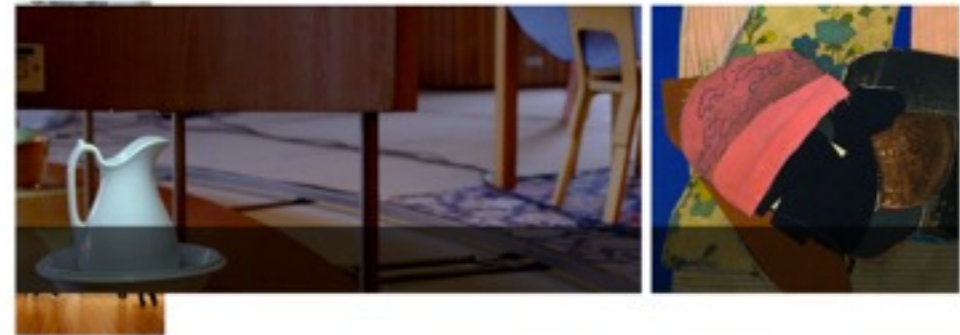
I'm sneaking in some CSS3;  
rgba() is a CSS3 property

# Styling the bottom three boxes

```
<div id="bigBox">  
  <div class="invisible"></div>  
  <div class="overlay"></div>  
</div>  
<div class="bottomBox" id="leftBox"></div>  
  
<div class="bottomBox" id="middleBox"></div>  
  
<div class="bottomBox" id="rightBox"></div>  
</div>
```

End of the div  
id="container"

- 
- 
- 



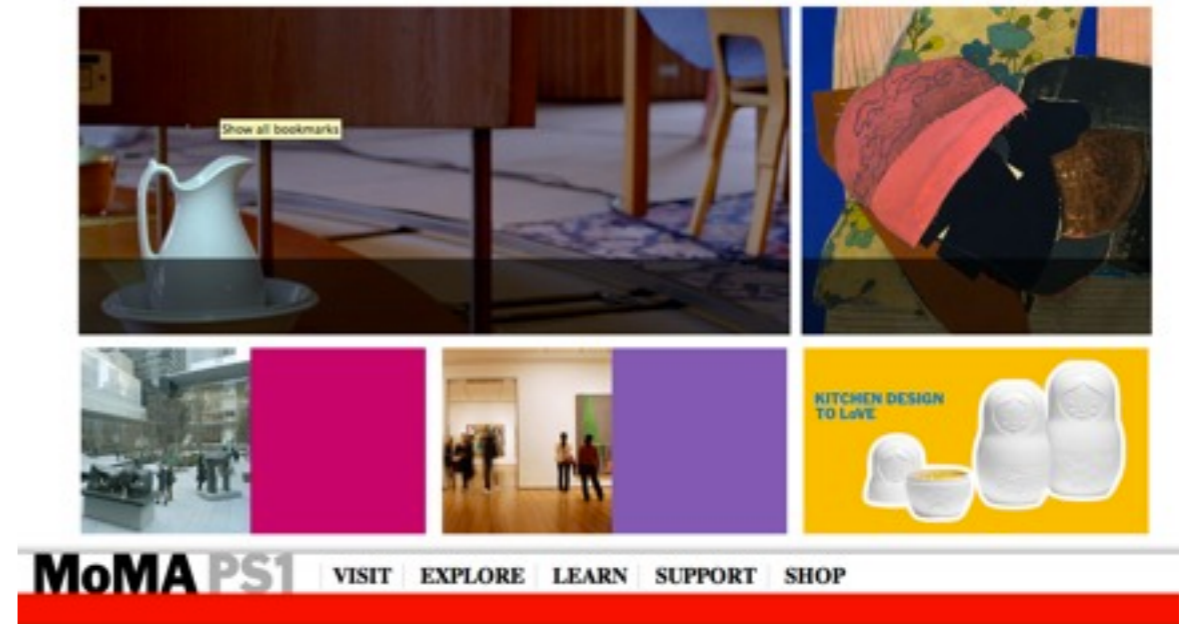
MoMA PS1 VISIT EXPLORE LEARN SUPPORT SHOP

# Styling the bottom 3 boxes: part I

```
.bottomBox
{
  width: 315px;
  height: 170px;
  border: 2px solid transparent;
}
```

# Styling the bottom 3 boxes: part 2

```
#leftBox
{
  background-color:#C06;
  position:absolute;
  top: 310px;
}
#middleBox
{
  background-color:#855CAD;
  position:absolute;
  top: 310px;
  left: 330px;
}
#rightBox
{
  background-color:orange;
  position:absolute;
  top: 310px;
  left: 660px;
}
```



# Changing as we hover!

```
#leftBox:hover
```

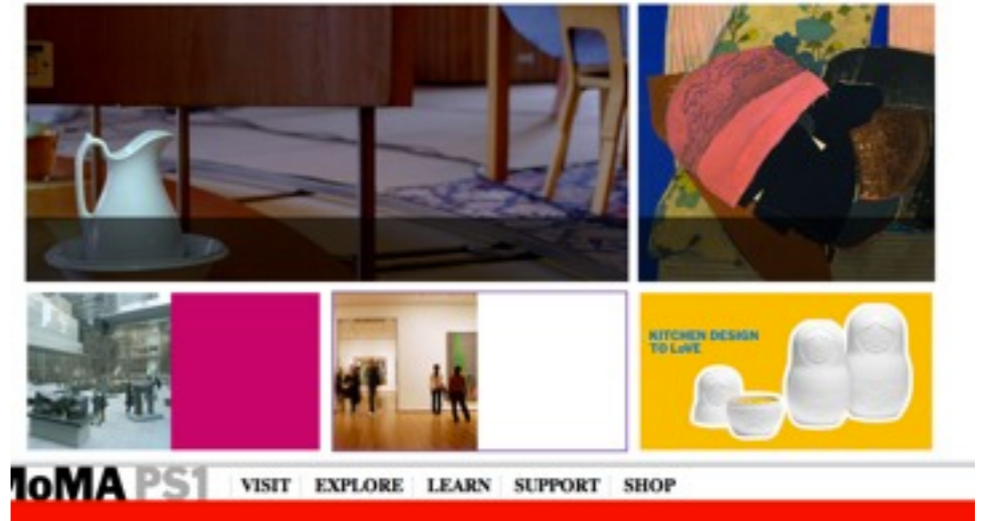
```
{  
  border:2px solid #C06;  
  background-color:white;  
}
```

```
#middleBox:hover
```

```
{  
  border:2px solid #855CAD;  
  background-color:white;  
}
```

```
#rightBox:hover
```

```
{  
  border:2px solid #007AC4;  
}
```



# Finished product

- To find a completed version of this file, see:  
<http://livetotry.com/GDI/codeSamples/imitationIsFlattery.html>

# What is HTML5?

- Formally, HTML5 is the W3C's specification for the next version of HTML.
- Informally, people use “HTML5” to refer to a whole set of new web standards:
  - HTML5
  - CSS3
  - JavaScript



# Where are we now?

- HTML5 is still in "working draft" stage
- Some of the tech is making it into browsers now, but it'll still be a while until the specification is finalized.
- It remains to be seen if all browsers will support all features, and WHEN they will support them.
- Here is a good page summarizing which features are supported by which browser:  
<http://caniuse.com>

# Detecting CSS3 Support

# Detecting support: Modernizr

Modernizr is an open-source JavaScript library that helps you understand what your visitor's browsers do and do not support.

With Modernizr, you can provide different CSS styling for browsers that do not support new CSS3 features, or use JavaScript to fall back gracefully if the visitor's browser does not support the new video element.

# Detecting support: Modernizr

You can download Modernizr here: <http://modernizr.com>.

You then want to include it in your page's <head> section:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>My Beautiful Sample Page</title>
  <script src="modernizr-1.7.min.js"></
script>
</head>
```

# Detecting support: Modernizr

Next, add the class “no-js” to the <html> element:

```
<html class="no-js">
```

When Modernizr runs, if your browser has JavaScript enabled, it will replace that class with the class “js”.

```
<html class="js">
```

Modernizr will then add classes for **every** feature it detects, prefixing them with “no-” if the browser doesn’t support it.

# Detecting support: Modernizr

If you are using Safari 5, which supports almost **everything** in HTML5/CSS3 currently, your `<html>` element will look something like this:

```
<html class="js canvas canvastext geolocation rgba hsla multiplebgs  
borderimage borderradius boxshadow opacity cssanimations csscolumns  
cssgradients cssreflections csstransforms csstransforms3d csstransitions video  
audio localStorage sessionStorage webworkers applicationcache fontface">
```

If you are using IE 8, which supports almost **nothing** in HTML5/CSS3 currently, your `<html>` element will look something like this:

```
<html class="js no-canvas no-canvas text no-geolocation no-rgba no-hsla no-  
multiplebgs no-borderimage ... you get the idea >
```

# A Modernizr CSS Example

```
.csscolumns ol {  
  -moz-column-count: 2;  
  -webkit-columns: 2;  
  -o-columns: 2;  
  columns: 2;  
}
```

```
.no-csscolumns ol {  
  float: left;  
  margin: 0 0 20px;  
}
```

```
.no-csscolumns ol li {  
  float: left;  
  width: 180px;  
}
```

If the browser supports CSS columns, the `.csscolumns` style is applied.

If the browser *doesn't* support CSS columns, as determined by the “no-csscolumns” class added by Modernizr, the `.no-csscolumns` style is applied. Instead of using CSS columns, we float our list items and apply some margins and widths to get a similar result.

# Detecting support: Modernizr

To learn more about how to use Modernizr, see:

- <http://www.alistapart.com/articles/taking-advantage-of-html5-and-css3-with-modernizr/>
- <http://www.modernizr.com/docs/>
- <http://diveintohtml5.org/detect.html>

# CSS3 Effects

# New face of fonts: @font-face

## **Old Way:**

- font-family: Helvetica, Verdana, Arial, sans-serif;
  - Have fallback fonts in case your visitors did not have your favorite font installed.
- Create an image with a specific font, to ensure it looks the way you want

# New face of fonts: @font-face

## New Way:

With CSS3, instead of relying on fonts everyone has installed, or using a specific font in an image, you can instruct the browser to **download the font** if the person viewing your site is missing the font:

```
@font-face
{
  font-family: "Bitstream Vera Serif Bold";
  src:
    url("http://developer.mozilla.org/@api/deki/files/2934/
=VeraSeBd.ttf");
}

body {
  font-family: "Bitstream Vera Serif Bold", serif
}
```

- NOTE that this will only make the font available to the **browser**, not to the rest of the computer.

# Other font solutions

Here's a great article on how @font-face compares to Google's font API and TypeKit:

<http://www.red-team-design.com/google-font-api-and-typekit-solutions-vs-font-face>

<http://www.google.com/webfonts>

# CSS3: Hands-on practice

- The CSS3 properties I will be reviewing are all illustrated in this page: <http://alexisgo.com/resistor/css3effects/effects.html>
- Instead of looking at the final solution in class, I suggest you leave it as an exercise for later.
- For now, please follow along at this JSFiddle:
  - <http://jsfiddle.net/7JCWN/1/>
  - I've created a skeleton and filled in some of the more complicated examples, but left most for you to fill in!

# Rounded Corners

- border-radius (Safari and Chrome)
- -moz-border-radius (Firefox)

## Border-Radius

```
h2
{
  background-color:pink;
  padding: 5px 10px;
  width:400px;

  /* firefox */
  -moz-border-radius: 20px;

  /* safari and chrome*/
  -webkit-border-radius: 20px;

  border-radius: 20px;
}
```

# Browser Prefixes

- Why do we use...?
  - border-radius for Safari and Chrome
  - -moz-border-radius (Firefox)
  - -webkit-border-radius
  - -o-border-radius
  - -ms-border-radius
- The CSS3 (and HTML5) specs are still in **draft** format.
- While the names and parameters of the new CSS properties are not *likely* to change, there is no guarantee that they won't.
  - In order to add support for these new features

# Browser Prefixes

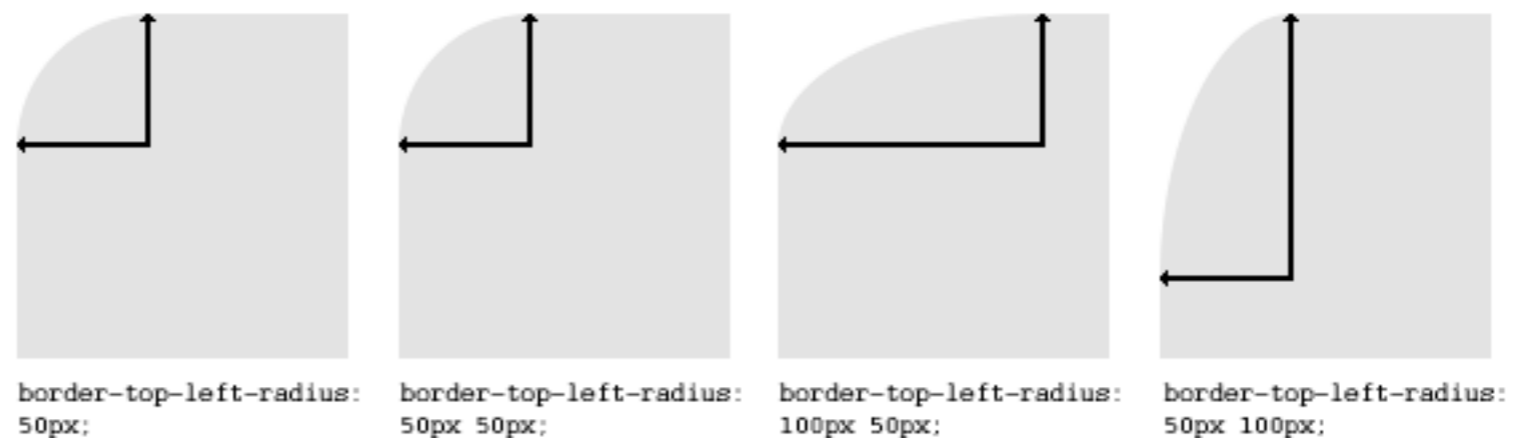
Why do we use

- border-radius for Safari and Chrome
- -moz-border-radius (Firefox)
- Using the browser prefixes ensures that the functionality will work, even if the w3c changes the standard.
- Firefox uses the **-moz** prefix
- Chrome and Safari use the **-webkit** prefix
- Opera (out of scope for today) uses the **-o** prefix
- In cases where the spec has been mostly finalized, you can simply use the property name. We have done this above, with **border-radius**

# Unevenly Rounded Corners

To vary the rounding, you can specify a different radius on each corner:

- `border-radius` (Safari): `0 20 100 50px;`
- `-moz-border-radius` (Firefox): `0 20 100 50px;`



- You can also specify a different horizontal and vertical radius:

{

`border-top-right-radius: 160px 10px;`

`border-bottom-left-radius: 160px 10px;`

`border-bottom-right-radius: 160px 20px;`

}

# Drop Shadows

```
#drop
```

```
{
```

```
/* firefox */
```

```
-moz-box-shadow: black 0px 5px 5px;
```

vertical  
offset

horizontal  
offset

blur radius

```
/* safari and chrome*/
```

```
-webkit-box-shadow: black 0px 5px 5px;
```

```
/* fallback */
```

```
box-shadow: black 0px 5px 5px;
```

```
}
```


**Drop Shadow**

# Inset Shadows

```
#inset
{
  /* firefox */
  -moz-box-shadow: inset black 0px 5px 5px;

  /* safari and chrome*/
  -webkit-box-shadow: inset black 0px 5px 5px;

  box-shadow: inset black 0px 5px 5px;
}
```

A horizontal rounded rectangle with a light pink fill and a dark shadow, containing the text "Inset Shadow" in a bold, black, serif font.

**Inset Shadow**

# Text Shadows

`text-shadow: 0 2px 5px black;`

The diagram shows the CSS property `text-shadow: 0 2px 5px black;` with four rounded rectangular boxes highlighting the values: `0` (light blue), `2px` (light blue), `5px` (light purple), and `black` (light green). A green arrow points from the `black` box to the text "shadow color". Two blue arrows point from the `0` and `2px` boxes to the text "x and y coordinates of the text shadow". A purple arrow points from the `5px` box to the text "blur radius of the text shadow".

x and y coordinates of the text shadow

blur radius of the text shadow

shadow color

`text-shadow: 0 2px 5px black;`

**Text Shadow**

# Color

Before, we had three ways to define colors on websites:

1. Color Name (color: blue);
2. Hexadecimal Value (color: #CCC);
3. rgb [color: rgb(255, 255, 255) or color: rgb(90%, 80%, 90%)]

CSS3 has introduced two new ways:

## 1. **rgba**

- The **a** stands for **alpha** (the level of transparency).

## 2. **hsl and hsla**

- HSL = Hue, Saturation and Lightness

# Color: rgba

- rgba = Red, Green, Blue, Alpha
- **Example:**
  - background-color: rgba(255, 255, 255, 0.5);

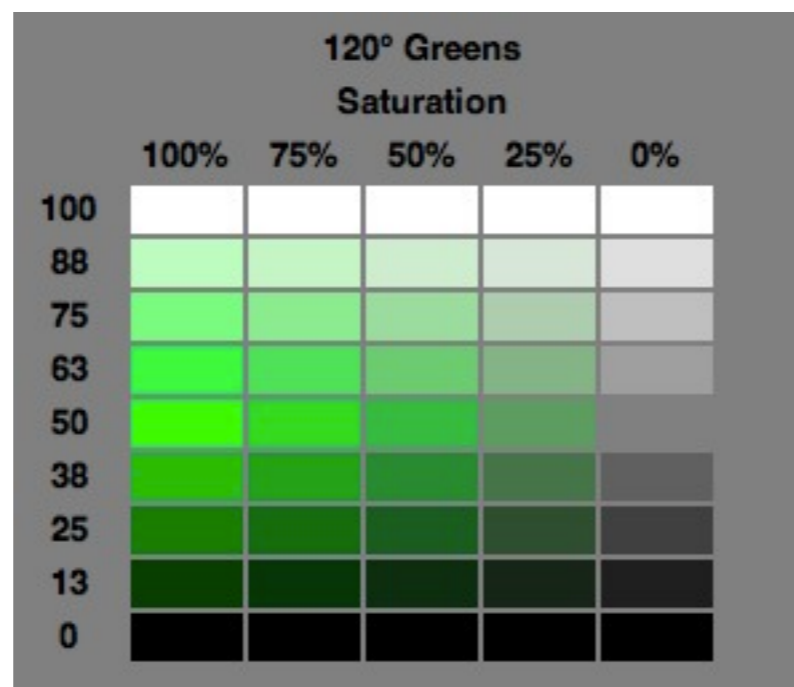
**rgb: old way, no alpha**

**rgba: new way, with alpha**

# Color: hsl and hsla

- HSL = Hue, Saturation and Lightness
- HSLA = Hue, Saturation, Lightness **and Alpha**
- **Syntax:**
  - **hsl( hue--in degrees from 0-359, saturation--in % from 0-100%, lightness--in % from 0-100%)**
  - **hsla( hue--in degrees from 0-359, saturation--in % from 0-100%, lightness--in % from 0-100%, alpha--from 0.0-1.0)**

**Lightness:**



# Color: hsl and hsla

- **Example:**

- `background-color: hsl(260, 50%, 75%);`



- <http://www.w3.org/TR/css3-color/#hsl-examples>

# Animations

# CSS Transforms and Transitions

- We can create animations by leveraging the new CSS Transform and Transition properties.
- Transforms allow us to manipulate our elements.
- Transitions allow us to specify over what time duration these changes should happen: effectively animating the changes.

# CSS Transforms

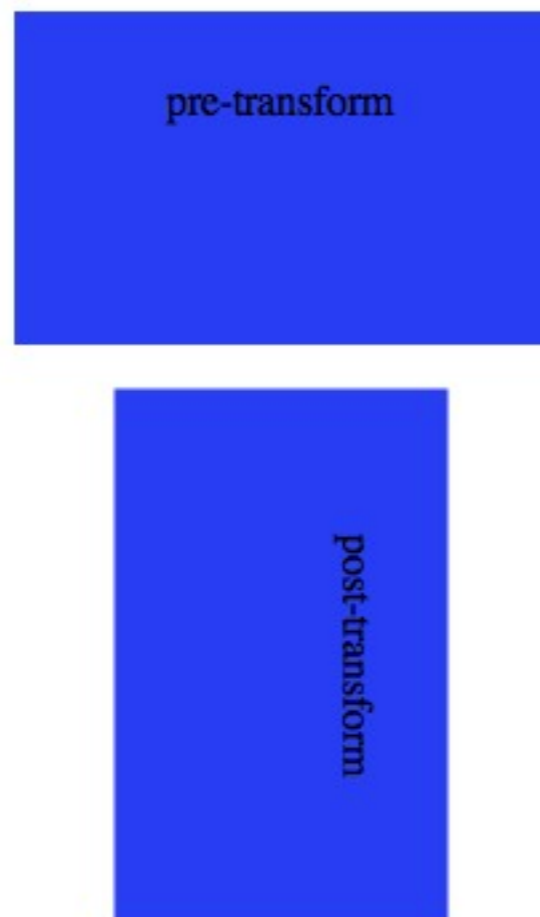
- You can use CSS transforms to rotate or scale elements on your page.
- We used to need JavaScript in order to do stuff like this!
- Our options: **rotate**, **scale**, **skew** and **translate**.

# CSS Transforms

- Our JSFiddle, starting file: <http://jsfiddle.net/8etSs/1/>
- finished file: <http://jsfiddle.net/fiddlefiddle/8etSs/18/>

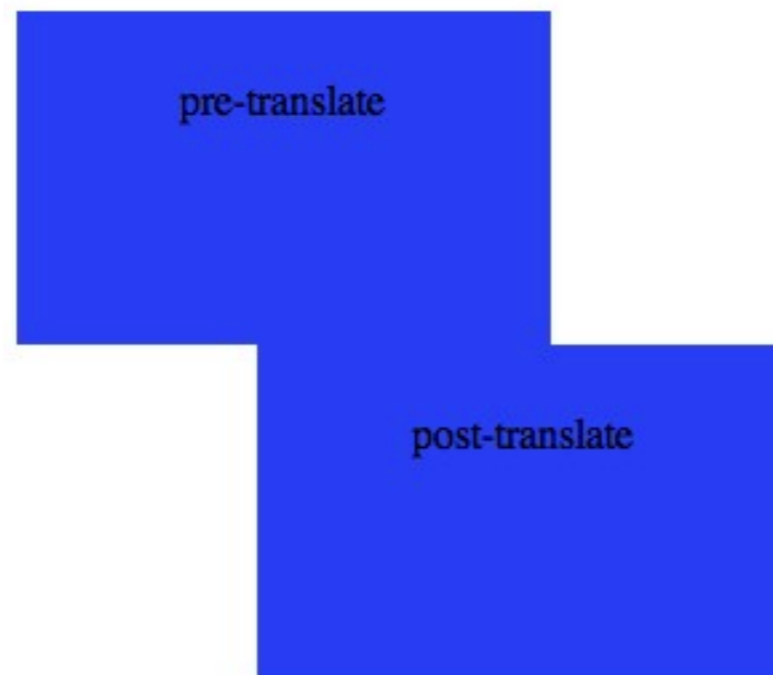
# CSS Transforms: Rotate

- `-webkit-transform: rotate(90deg);`
- `-moz-transform: rotate(90deg);`
- This will rotate your element 90 degrees clockwise



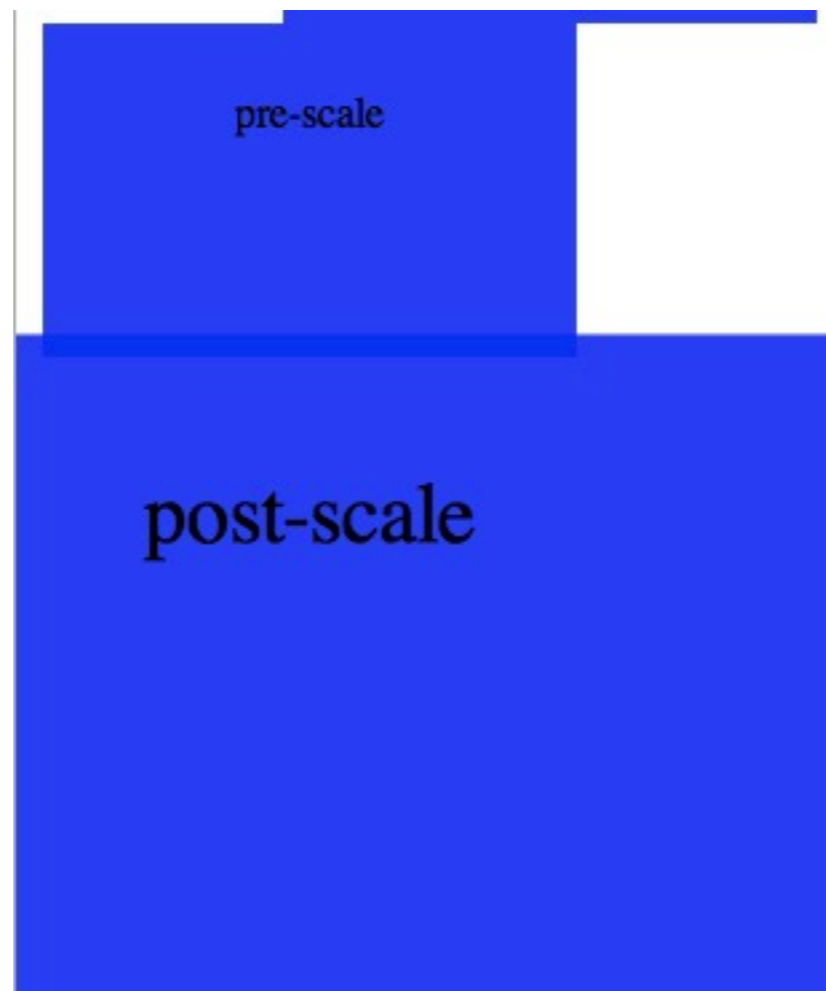
# CSS Transforms: Translate

- `-webkit-transform: translateX(90px);`
- `-moz-transform: translateX(90px);`
- This will move your element over 90px to the right (along the x-axis)



# CSS Transforms: Scale

- `-webkit-transform: scale(2.0);`
- `-moz-transform: scale(2.0);`



# CSS Transforms: Scale

- We can also scale only the vertical or the horizontal by specifying two values
- For example, this code will double the width, but keep the height the same:
  - `-webkit-transform: scale(2.0, 1.0);`
  - `-moz-transform: scale(2.0, 1.0);`
- This code will keep the width the same, but shrink the height to 1/10th of its original size:
  - `-webkit-transform: scale(1.0, 0.1);`
  - `-moz-transform: scale(1.0, 0.1);`

# CSS Transforms: Example

- `/* make a picture 1.25 times its normal size*/`
  - `-webkit-transform: scale(1.25);`
  - `-moz-transform: scale(1.25);`
  - `-o-transform: scale(1.25);`

# CSS Transforms: the origin

- By default, all the transforms occur from the center of the element.
- If you'd like the origin of the element to be somewhere other than the center, you can use the transform-origin property.
- Example:
  - **-webkit-transform-origin: 0 0;**
  - **-moz-transform-origin: 0 0;**
  - **-op-transform-origin: 0 0;**
  - **transform-origin: 0 0;**

# CSS Transforms: another example

- This JSFiddle example uses two divs to build a circle with a shadow underneath.
- The example uses a combination of CSS3 effects to create the shadow: A radial gradient and two transforms, a scale and a translateY.
- It also uses two more basic, CSS2 properties to position the shadow behind the circle along the z-axis: position and z-index
- Starting file: <http://jsfiddle.net/fiddlefiddle/patYu/2/>
- Ending file: <http://jsfiddle.net/fiddlefiddle/patYu/4/>



# CSS Transforms: another example

I wrote up a blog entry to describe the shadow trick we did in class more clearly. You can find that here:

<http://www.fuckyeahhtml5.com/2011/06/tweak-your-shadows-with-css3-gradients-and-transforms/>



# CSS Transitions

Right now, all of these Transforms happen *instantly*.

Usually, we want Transforms to happen over time, over at least one second, for example.

We can make that happen by combining our Transforms with **Transitions**.

# CSS Transitions: No JavaScript!

- We can also leverage CSS **pseudo-classes** to use CSS Transitions.
- In the sample page: <http://alexisgo.com/resistor/css3effects/transform.html> a combination of Transitions and Transforms are used to both **fade in** and **scale up** the paintings as you hover your mouse over them.

# CSS Transitions: No JavaScript!

- Sample page: <http://alexisgo.com/resistor/css3effects/transform.html>

```
.paintings img
{
    padding:10px;
    vertical-align:middle;
    opacity:0.5;
    -webkit-transition : all 1.0s;
    -moz-transition : all 1.0s;
    -o-transition : all 1.0s;
}
```

Normal opacity of the painting is 50%

This means **all** transitions will happen over 1 second

```
.paintings img:hover
{
```

When you hover over a painting image, the opacity changes to 100%,

```
/*make the picture opaque*/
opacity:1.0;
```

```
/* make the picture 1.25 times its normal size*/
-webkit-transform: scale(1.25);
-moz-transform: scale(1.25);
-o-transform: scale(1.25);
```

```
}
```

# CSS Transitions: No JavaScript!

- Let's try adding something similar to the `#transition` selector in our JSFiddle!
- **<http://jsfiddle.net/8etSs/1/>**

# CSS Transitions

## Making things animate!

- With the combination of HTML, CSS and a little bit of JavaScript, we can animate our HTML elements.
- Here is a no-JavaScript example: [http://w3schools.com/css3/tryit.asp?filename=trycss3\\_transition1](http://w3schools.com/css3/tryit.asp?filename=trycss3_transition1)
- Current support for CSS3 Transitions:
  - Safari 3.1+
  - Mobile safari on iPhone if you have iOS 2.0+
  - Firefox 3.7+ Your best bet is to get 4.0, their current beta
  - Opera 10.5x

# CSS Transitions

- Let's build a simple div that fades out when you click on it. (Final page: <http://alexisgo.com/teaching/resistor/transitions.html>)

- HTML:

- ```
<body>
  <div id="overlay" onclick="divClicked()">
    Hi
  </div>
</body>
```

- CSS:

- ```
<style>
  #overlay
  {
    background-color: rgba(40%, 50%, 40%, 0.8);
    width: 300px;
    height: 50px;
    opacity: 1;
  }
  #overlay.invisible
  {
    -webkit-transition: opacity 1.0s; /*change the opacity over
the course of 1 second */
    opacity: 0;
  }
</style>
```

# CSS Transitions

- Let's build a simple div that fades out when you click on it.

- HTML:

```
<body>
  <div id="overlay" onclick="divClicked()">
    Hi
  </div>
</body>
```

- CSS:

```
<style>
  #overlay
  {
    background-color: rgba(40%, 50%, 40%, 0.8);
    width: 300px;
    height: 50px;
    opacity: 1;
  }
  #overlay.invisible
  {
    -webkit-transition: opacity 1.0s; /*change the opacity over
the course of 1 second */
    opacity: 0;
  }
</style>
```

class="invisible" doesn't appear anywhere in the HTML.  
How does this class get added?



# CSS Transitions

- Let's build a simple div that fades out when you click on it.

- `<style>`

```
#overlay
{
    background-color: rgba(40%, 50%, 40%, 0.8);
    width: 300px;
    height: 50px;
    opacity: 1;
}
```

```
#overlay.invisible
```

```
{
    -webkit-transition: opacity 1.0s; /*change the opacity over
the course of 1 second */
    opacity: 0;
}
```

`</style>`

- JavaScript:

- ```
<script type="text/javascript">
function divClicked()
{
    var div = document.getElementById('overlay');
    div.className = 'invisible';
}
</script>
```

class="invisible" doesn't appear  
anywhere in the HTML.

How does this class get added?

# CSS Transitions

- More on CSS3 Transitions:
- <http://css3.bradshawenterprises.com/>
- <http://samuli.hakoniemi.net/css3-transitions-are-we-there-yet/>
- Final version of transform and transitions JSFiddle: <http://jsfiddle.net/8etSs/>

# Further Reading

## General Web Development Tutorials:

- <http://www.webmonkey.com/tutorials/>
- <http://www.webmonkey.com/cheat-sheets/>
- [http://www.webmonkey.com/2010/02/color\\_charts/](http://www.webmonkey.com/2010/02/color_charts/)
- <http://htmldog.com/guides/>

## Positioning with CSS:


- The Official CSS Guide: [http://www.w3schools.com/Css/css\\_positioning.asp](http://www.w3schools.com/Css/css_positioning.asp)
- CSS Positioning in 10 Steps: <http://www.barelyfitz.com/screencast/html-training/css/positioning/>
- <http://www.brainjar.com/css/positioning/>

# Further Exercises

# Building a menubar

- We will practice using the following CSS and HTML concepts to build a navigation bar:
  - HTML div element
  - Using CSS to style an HTML list element
  - Using tricks with CSS borders to make an arrow pointing to our current page
  - Use CSS background-color, margin, and padding to make it look nice
  - Leverage CSS pseudo-classes to give our links some interactivity
  - Practice CSS nesting to target only the ul and lis inside a given div

# Building a menubar



hi hey hlo

- hi
- hey
- hlo

Finished code: [http://alexisgo.com/teaching/  
codesamples/lists.html](http://alexisgo.com/teaching/codesamples/lists.html)

# Building a fixed menubar

Need to put my logo here



- hi
- hey
- hlo

text

text

text

text

text

text

[Finished product: http://alexisgo.com/teaching/codesamples/fixedMenu.html](http://alexisgo.com/teaching/codesamples/fixedMenu.html)

# Single Column Sample

[http://ebooks.adelaide.edu.au/w/woolf/virginia/w91r/  
chapter4.html](http://ebooks.adelaide.edu.au/w/woolf/virginia/w91r/chapter4.html)

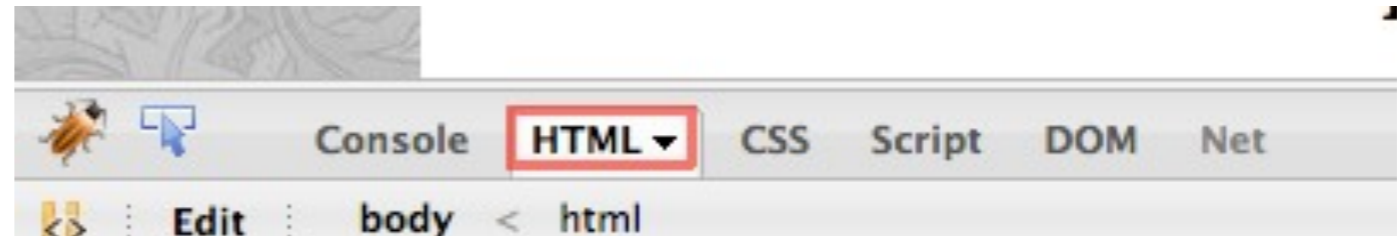
(a shorter version of the same link is: [http://bit.ly/v\\_woolf](http://bit.ly/v_woolf))

# Examining a Single Column Layout

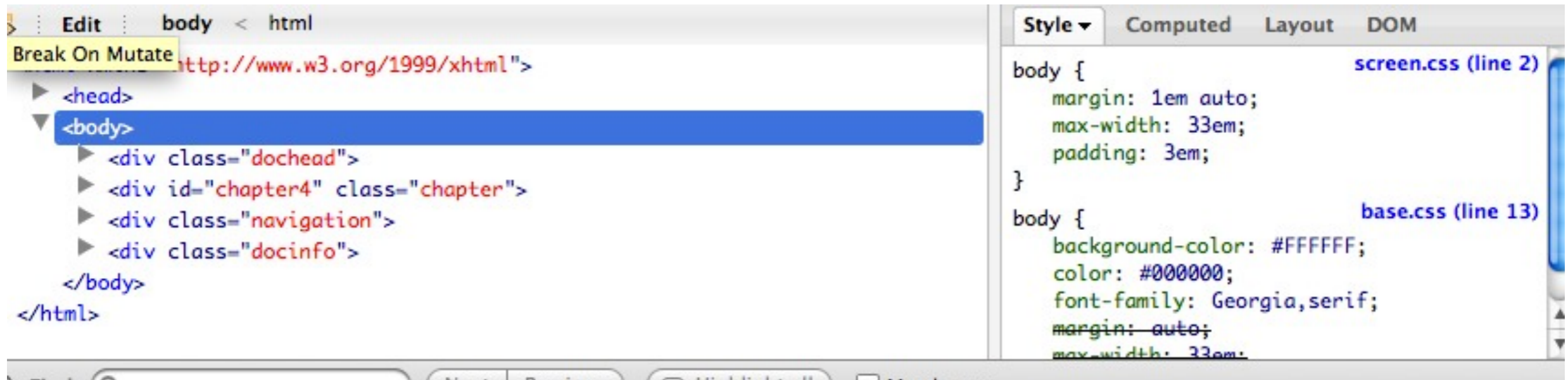
Go to [http://bit.ly/v\\_woolf](http://bit.ly/v_woolf)

With Firebug open:

1. Click the HTML tab
2. Click on `<body>`

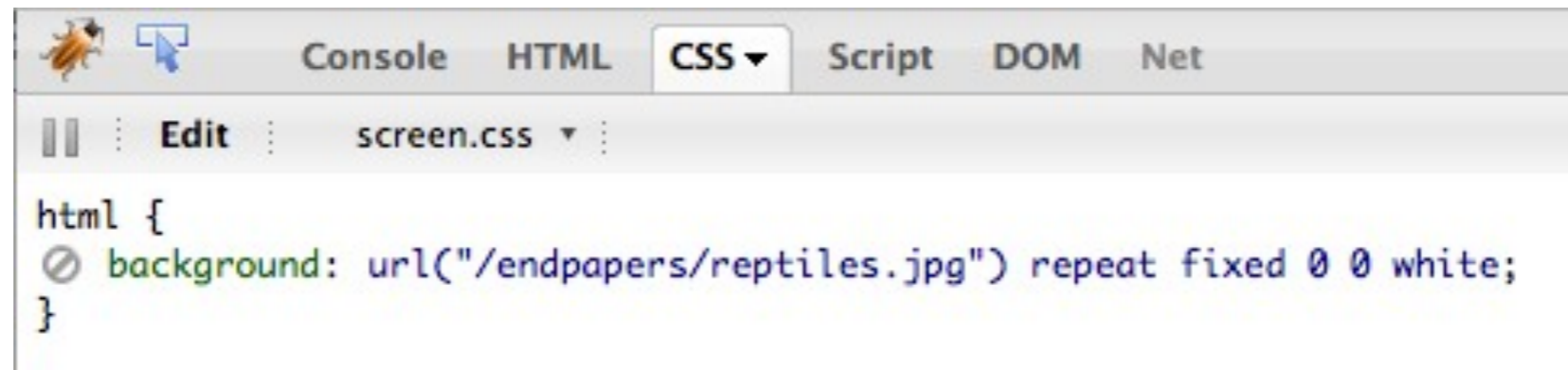


3. Note the Style details that come up in the right-hand window. Firebug should display two sets of styles for the **body** element: one set in the file `base.css` and another set of styles in the file `screen.css`:



# Examining a Single Column Layout

- Remove the **background** style from the **html** selector in `screen.css`:



```
html {  
  background: url("/endpapers/reptiles.jpg") repeat fixed 0 0 white;  
}
```

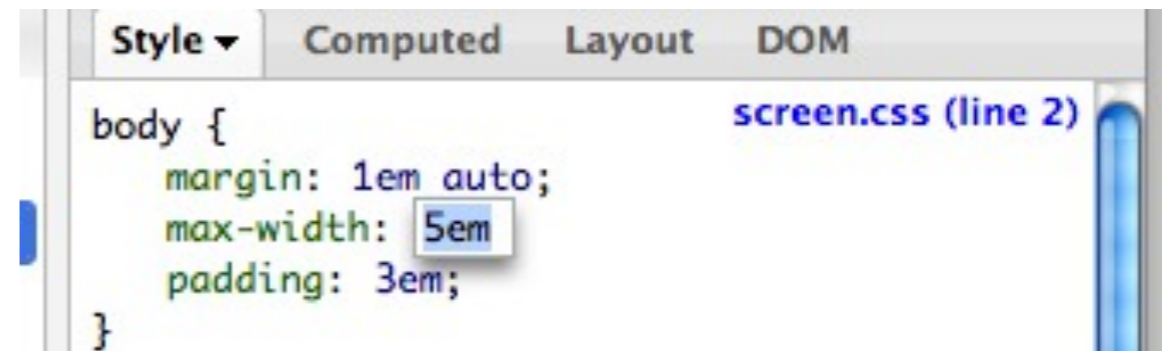
- Remove the **padding: 3em;** from the **body** selector in `screen.css`:



```
body {  
  margin: 1em auto;  
  max-width: 33em;  
  padding: 3em;  
}
```

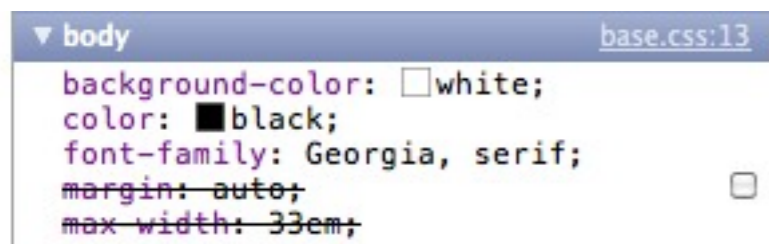
# Examining a Single Column Layout

- Change the **max-width** style from the **body** selector in screen.css file from **33em** to **5em**:

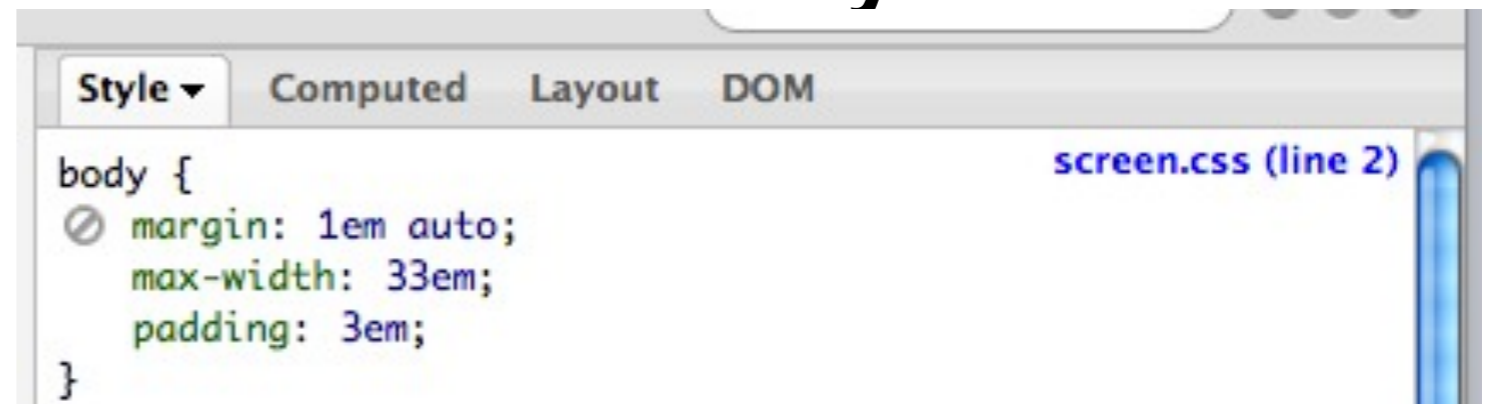


```
body {  
  margin: 1em auto;  
  max-width: 5em;  
  padding: 3em;  
}
```

- Remove the **margin: 1em auto;** from the **body** selector in screen.css, and the **margin: auto** from the **body** selector in base.css:



```
body {  
  background-color: white;  
  color: black;  
  font-family: Georgia, serif;  
  margin: auto;  
  max-width: 33em;  
}
```



```
body {  
  margin: 1em auto;  
  max-width: 33em;  
  padding: 3em;  
}
```

# Building a Single Column Layout

- 1) Create a new HTML file
- 2) Add four divs inside the body tags:

```
<div class="wrap">  
  <div class="header">  
    <!-- banner image goes here -->  
  </div>  
  <div class="mainContent">  
    <!-- main content -->  
  </div>  
  <div class="footer">  
    <!-- footer -->  
  </div>  
</div>
```

3) Create a new CSS file, and copy/paste from the HTML, then edit into proper CSS

4) Make sure to add a link to your stylesheer `<link rel="stylesheet" type="text/css" href="stylesheet.css" />`

# Building a Single Column Layout

You should now have this shell in your CSS file:

```
.wrap  
{  
}
```

```
.header  
{  
}
```

```
.mainContent  
{  
}
```

```
.footer  
{  
}
```

# Building a Single Column Layout

Add the following to the **.wrap** selector in your CSS file:

```
.wrap {  
  background-color:#ffffff;  
  margin:0 auto;  
  width:700px;  
}
```

Create a new CSS selector for the **html** and **body** elements:

```
html, body {  
  background-color: #5d83b1;  
  font-family: Garamond, Helvetica, Arial, sans-serif;  
  min-height: 100%;  
}
```

Add some text after your "wrap" div:     **<div class="wrap">hi**

# Building a Single Column Layout

We will finish the rest of the page by copy-pasting bit-by-bit from the sample files:

- oneCol.html
- oneCol.css

These files are available at: [http://livetotry.com/GDI/class %204.zip](http://livetotry.com/GDI/class%204.zip)

Once you've unzipped the file, navigate to the folder: **class 4/  
one col layout/**

# Sample Two-Column Layout

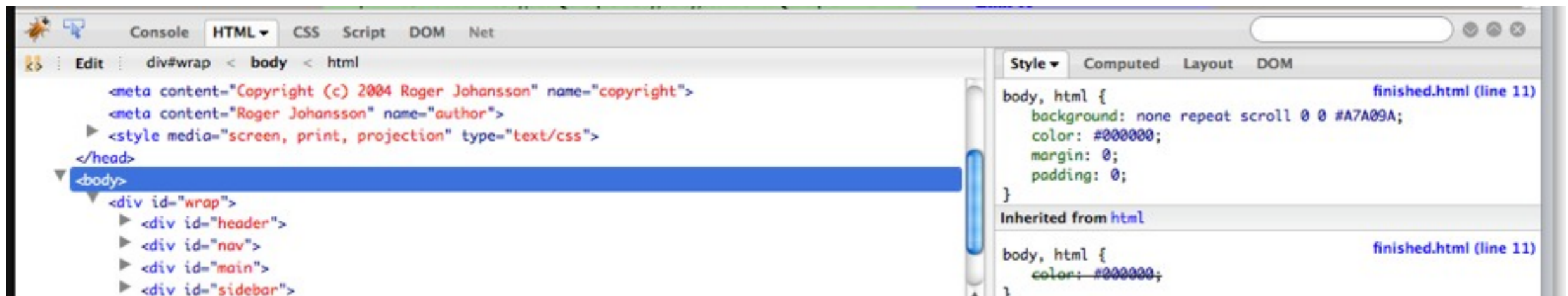
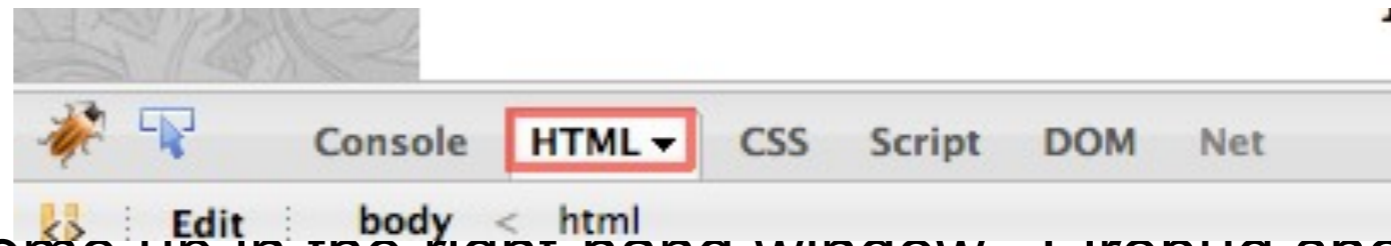
- [http://www.456bereastreet.com/lab/developing\\_with\\_web\\_standards/csslayout/2-col/finished.html](http://www.456bereastreet.com/lab/developing_with_web_standards/csslayout/2-col/finished.html)
  - shorter version of the same link: [http://bit.ly/two\\_col](http://bit.ly/two_col)

# Examining a Two-Column Layout

Go to [http://bit.ly/two\\_col](http://bit.ly/two_col)

With Firebug open:

1. Click the HTML tab
2. Click on <body>
3. Note the Style details that come up in the right-hand window. Firebug should display the styles for the **body** and **html** elements.
4. Note that, unlike in the Single-column example page, the styles here come from the HTML page itself (finished.html). This author has chosen to put the styles inline in the HTML page, instead of creating a separate CSS file. There is more than one way to do it!



# Examining a Two-Column Layout

Go to [http://bit.ly/two\\_col](http://bit.ly/two_col)

- Expand the **<body>** element, and then the **<div id="wrap">** element.
- You'll see five divs inside: header, nav, main, sidebar and footer.
- Mouse over each div, to see what it controls on the page.
- Click on **<div id="main">** in order to reveal the CSS that is styling this element.
- Remove the **width: 480px;** and see what happens
- Remove the **padding: 10px;** and see what happens
- Remove the **background: none repeat scroll 0 0 #99CC99** selector & values, and see what happens.

# Sample Three-Column Layouts

- <http://store.apple.com/us> (all fixed)
- <http://www.amnesty.org/en/who-we-are> (middle column is liquid)
- <http://www.sparkfun.com/commerce/news.php?id=448> (all fixed width columns)

# Examining a Three-Column Layout

Go to <http://www.amnesty.org/en/who-we-are>

This is a much more complicated page, so we're going to have to dig around a bit to find how the main content is styled.

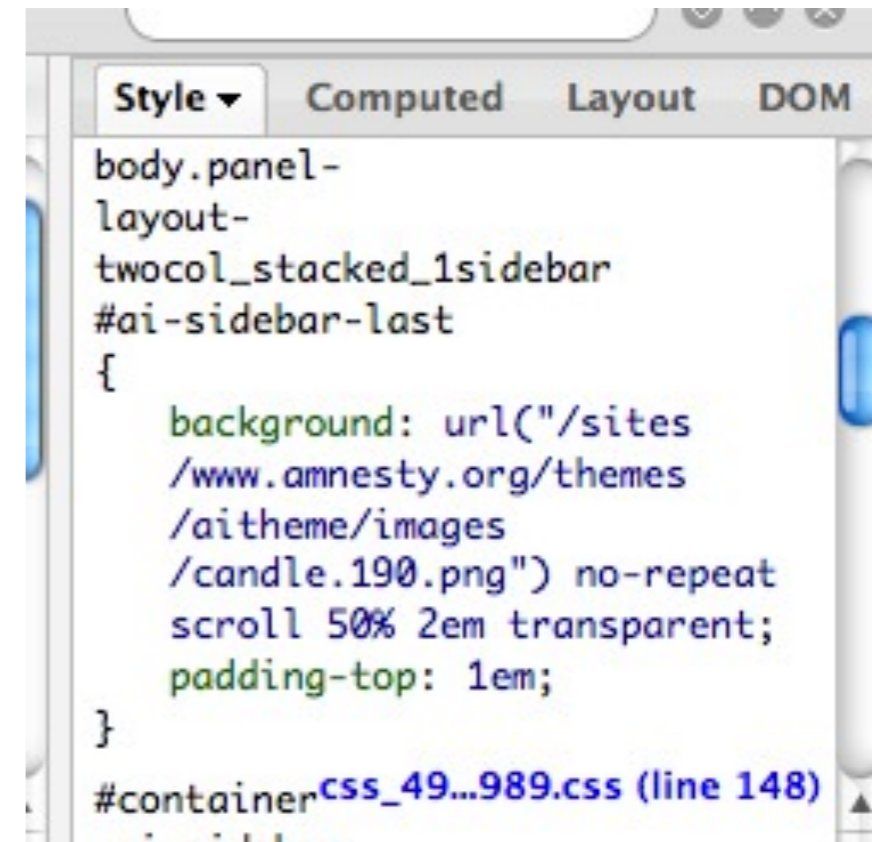
With Firebug open:

1. Click the HTML tab
2. Expand the **body** element
3. Expand the first **div** element you see (it's called **div id="page"**)
4. Now we see some stuff that's relevant: more divs called **header**, **container**, **footer-wrapper**
5. Expand **div id="container"** and then expand the one div it has inside it as well (it's called **div class="panel-display-blah..."**)
6. Mouse over **div id="ai-sidebar-first"**. That's the left column.
7. Mouse over **div id="ai-sidebar-last"**. You can see this is the right column.
8. Mouse over **div id="ai-sidebar-main"**. This is the middle column.

# Examining a Three-Column Layout

- We're going to examine the styles in **div id=ai-sidebar-last** so click on it.
- Scroll down a bit in the style window in Firebug until you see a **background** property.
- Remove that background property, and see what happens.
- Scroll down in the style window until you see this:

```
#ai-sidebar-last
{
  float: left;
  width: 180px;
}
```
- Try removing float: left, and see what happens to the page.



The screenshot shows the Firebug developer tool's 'Style' window. The 'Style' tab is selected, and the list of styles for the element '#ai-sidebar-last' is visible. The following CSS rules are shown:

```
body.panel-
layout-
twocol_stacked_1sidebar
#ai-sidebar-last
{
  background: url("/sites
/www.amnesty.org/themes
/aitheme/images
/candle.190.png") no-repeat
scroll 50% 2em transparent;
padding-top: 1em;
}
```

Below the rules, the source file is identified as '#container css\_49...989.css (line 148)'.

# Examining a Three-Column Layout

To reiterate, peeking into the guts of more complex pages requires:

- patience
- digging
- finding the right divs (usually) that hold the content you care about
- removing styles until you find the style(s) that controls the effect you like and you want to copy

# Thank You!

Thank you for being a great class!

We hope you build some amazing sites and please share your creations with us.

We want to know your feedback so we can make the class better each time. Watch your email for a link to an anonymous survey about the class.

You can always reach us via the [Meetup group](#), or the [Girl Develop IT](#) website.

Thank you!