

Slides:  
<http://alexisgo.com/GDI/class4.pdf>

# 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 {  
    text-decoration:none;  
}  
  
a:hover {  
    text-decoration:underline;  
}
```

# 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>
  
  width="100" height="200" class="kittenPics">
```

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

# A Super Fast Quiz!

```
<html>
<head>
    <style>
        .kittenPics
        {
            border: 2px solid pink;
        }
    </style>
</head>
<body>
    
    width="100" height="200" class="kittenPics">
```

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-left: auto;** and **margin-right: 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

# 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" href="style2.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-color: white;** if we put it in the body selector:

```
body {  
    background-color: 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;
    /* or... */
    visibility: hidden;
}
```

# 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/>

# Media Queries

- Media queries are an enhancement to the **media** attribute, that allow you to ask questions about the device being used to browse the web
- We can query the device's width, height, orientation, resolution, etc.
  - This is supported in IE 9 and later, and all modern browsers (Safari, Chrome, Firefox, Opera)

# Media Queries

- There are two parts to the query below:
  1. The media type (`screen`), and
  2. The query itself, which is enclosed within parentheses. The query will ask if the value of a particular media feature (`max-device-width`) meets a given target value (`320px`)

```
<link rel="stylesheet"
      media="screen and (max-device-width: 320px)"
      href="mobile.css">
<link rel="stylesheet"
      media="screen and (min-device-width: 321px)"
      href="standard.css">
```

# Media Queries

- You can also use media queries as selectors, creating different set of rules for different devices in the same stylesheet:

```
@media screen and (max-device-width: 480px) {  
    /* one set of styles for phones, iPad */  
}
```

```
@media screen and (min-device-width: 481px) {  
    /* another set of styles for desktop computers */  
}
```

# Orientation

```
@media all and (orientation:portrait) {
```

```
}
```

```
@media all and (orientation:landscape) {
```

```
}
```

# Targeting iPhone 4

- The iPhone 4 and 4S's Retina Display means there are 4 times more pixels than in non-Retina Display devices.
- We can check the device-pixel-ratio for a value of 2 or higher to target Retina Displays:

```
@media only screen and (-webkit-min-device-pixel-ratio: 2) { ... }
```

- The “only” keyword means that the styling will be ignored by any browsers that don't support CSS3
- Read more at: <http://blog.iwalt.com/2010/06/targeting-the-iphone-4-retina-display-with-css3-media-queries.html>

# Media Queries

- Further reading: <http://www.alistapart.com/articles/responsive-web-design/>
- Listing of sites using media queries, with previews of how they look on different devices: <http://mediaqueri.es/>

# **CSS for Page Layout**

# 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: Practice

- Let's add a paragraph and an image to our file from last week.
- We'll float the image left
- The paragraph text should wrap around the floated image

## HTML

```
  
<p>this is some text  
    this is some text  
    this is some text this is some text  
    this is some text this is some text  
    this is some text this is some text  
    this is some text this is some text  
    this is some text  
    this is some text  
    this is some text</p>
```

## CSS

```
#float {  
    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)

# Positioning with CSS

- CSS provides four main ways to position your content:
  - Static
  - Fixed
  - Relative
  - Absolute
- If you use these in combination with the CSS Box model, and the use of a property called **float**, you can layout your websites in myriad ways.

# Positioning with CSS

- `position: fixed;`
- `top: 10px;`
- `right: 10px;`
  
- `left: 10px;`
- `bottom: 10px;`

# Static and Fixed Positioning

## Static Positioning

HTML elements are positioned static by default; there is no need to set them to static. Static positioned elements are positioned according to the normal flow of a page. They ignore anything specified by top, bottom, right or left properties.

## Fixed Positioning

An element with fixed position is positioned relative to the browser window.

It will not move even if the window is scrolled--it will always stay in the same, fixed location on the screen.

See this in action:

- [http://www.w3schools.com/Css/tryit.asp?filename=trycss\\_position\\_fixed](http://www.w3schools.com/Css/tryit.asp?filename=trycss_position_fixed)
- <http://alexisgo.com/teaching/resistor/fixedPosExample.html>

# Fixed Positioning Lab!

For practice with fixed positioning, we're going to build a menubar at the bottom of a new page, similar to the one at <http://moma.org>

**Starting file:** <http://jsfiddle.net/fiddlefiddle/uge2Q/>

**Finished file:** <http://jsfiddle.net/fiddlefiddle/uge2Q/6/>

# Static and Fixed Positioning

## Relative Positioning

A relative positioned element is positioned relative to its normal position. You use the properties top, right, bottom and left to position an element.

For example, `position:relative; left:-20px;` will set an element 20 pixels to the left of its normal position; it subtracts 20 pixels from its normal left position.

See this in action:

- [http://www.w3schools.com/Css/tryit.asp?filename=trycss\\_position\\_relative](http://www.w3schools.com/Css/tryit.asp?filename=trycss_position_relative)

# Absolute Positioning

## Absolute Positioning

The position of an absolutely positioned element is determined by its offset values in the properties: top, right, bottom and left.

But, unlike relative positioning, where the offsets are measured relative to its normal position, an absolutely positioned element is offset from its "container block."

- WTF is a "container block"? It's the first parent element that has a position other than static. If no such element is found, the containing block is <html>.

Absolutely positioned elements can overlap other elements.

Unlike a Fixed element, an absolute element will move as you scroll away from it.

See this in action:

<http://alexisgo.com/teaching/class3/simpleAbsPos.html>

[http://www.w3schools.com/Css/tryit.asp?filename=trycss\\_position\\_absolute](http://www.w3schools.com/Css/tryit.asp?filename=trycss_position_absolute)

# Absolute Positioning

See this in action:

- [http://www.w3schools.com/Css/  
tryit.asp?  
filename=trycss\\_position\\_absolute](http://www.w3schools.com/Css/tryit.asp?filename=trycss_position_absolute)
- [http://alexisgo.com/teaching/class3/  
simpleAbsPos.html](http://alexisgo.com/teaching/class3/simpleAbsPos.html)

# Making our footer stick to the bottom

- We can use the concept of Absolute Positioning to force our footer to be on the bottom of the page.
- Before we do that, we need to add a width to our `#footer` selector:

```
#footer
{
    background-color: black;
    min-height: 20px;
    padding-top: 10px;
    width: 100%;
}
```

# Making our footer stick to the bottom

- We can use the concept of Absolute Positioning to force our footer to be on the bottom of the page.
- Let's set the footer to be absolutely positioned, 0px from the bottom and 0px from the left:

```
#footer
{
    background-color: black;
    min-height: 20px;
    padding-top: 10px;
    width: 100%;

    position: absolute;
    bottom: 0px;
    left: 0px;
}
```

# Absolute Positioning Lab

**Starting file:** <http://jsfiddle.net/fiddlefiddle/s45mk/>

**Finished file:** <http://jsfiddle.net/fiddlefiddle/s45mk/2/>

If you want all the files instead of the JSFiddles, you can find them here:  
<http://bit.ly/GDI3code>

# **Other CSS Concepts**

# CSS Grouping

- If you find yourself repeating the same styles for two different selectors, we can use a technique called **grouping** to save time.
- With grouping, we apply the same style to two selectors at once:

```
h1, h2
{
    margin: 0px;
}
```

commas allow us to **group** selectors.

When a comma separates two selectors, both selectors will get the style in brackets.

# Grouping & Nesting Exercise

- Let's create a simple example that uses:
  - CSS Pseudo-classes
  - CSS Nesting
  - CSS Grouping
- We'll start with this JSFiddle, and modify it: <http://jsfiddle.net/fiddlefiddle/zU6M6/1/>
- A shorter version of the same link:  
<http://bit.ly/CSSgroup>

# Playing with Borders

- Next, let's examine the file in the class4files folder called “borderTricks.html”
- We're going to learn a trick to use borders to make an arrow, for use ultimately in a menuBar, like this:



# Playing with Borders

- borderTricks.html is the final file. You can use it as a cheat sheet if you get stuck following along.
- Let's start with a div element in a new html file. Let's give it an id called "multiColor":

- <html>

```
<head>
  <title>Border Experiments</title>
</head>
<body>

  <div id="multiColor">
    </div>
  </body>
</html>
```

# Playing with Borders

- Now, let's set the style for our `<div id="multiColor">` using an internal stylesheet.

```
<head>
  <style>
    #multiColor
    {
      width:100px;
      height:100px;
      border-top:orange solid 20px;
      border-bottom: green solid 20px;
      border-left: red solid 20px;
      border-right: yellow solid 20px;

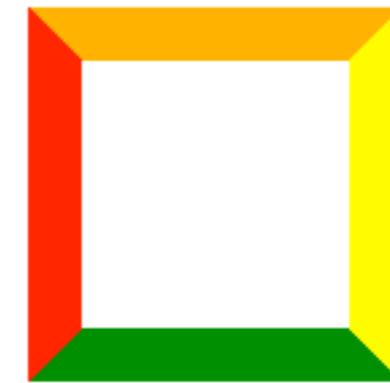
    }
  </style>
</head>
```



# Playing with Borders

- Notice those diagonal edges where the borders change color? We're going to leverage that later.

```
<head>  
  <style>  
    #multiColor  
    {  
      width:100px;  
      height:100px;  
      border-top:orange solid 20px;  
      border-bottom: green solid 20px;  
      border-left: red solid 20px;  
      border-right: yellow solid 20px;  
    }  
  </style>  
</head>
```



# Playing with Borders

- Now, let's create a new div with a similar style, but make it more narrow.

HTML: `<div id="multiColorNarrow"></div>`

CSS: `<style>`

```
#multiColorNarrow
{
    width:1px;
    height:100px;
    border-top:orange solid 20px;
    border-bottom: green solid 20px;
    border-left: red solid 20px;
    border-right: yellow solid 20px;
}
</style>
```



# Playing with Borders

- Let's make a new div that's even smaller

HTML: `<div id="multiColorSmushed">  
 </div>`

CSS: `<style>`

```
#multiColorSmushed
{
  width:1px;
  height:0px;
  border-top:orange solid 20px;
  border-bottom: green solid 20px;
  border-left: red solid 20px;
  border-right: yellow solid 20px;
}
</style>

</head>
```



# Playing with Borders

- Let's see what happens if we remove the top border

HTML:

```
<div id="multiColorArrows">  
  </div>
```

CSS: `<style>`

```
#multiColorArrows  
{  
  width:1px;  
  height:0px;  
  /* remove the top border */  
  border-top:none;  
  border-bottom: green solid 20px;  
  border-left: red solid 20px;  
  border-right: yellow solid 20px;  
}  
</style>
```



# Playing with Borders

- Now let's play with **transparency** to make JUST a green arrow!

**HTML:**    `<div class="greenArrow"> </div>`    `<!-- we'll use a class instead of a div so we can reuse this style more than once -->`

**CSS:**

```
.greenArrow
{
    width:1px;
    height:0px;
    border-top:none;
    border-bottom: green solid 20px;
    border-left: transparent solid 20px;
    border-right: transparent solid 20px;
}
```



We will keep the right and left borders, so as to preserve the bottom green arrow, but make them transparent. This will cause us to see ONLY the bottom green border, which will look like a green arrow!

# Playing with Borders

- Now let's make a green line we can put under our green arrow

HTML:      `<div class="navBar"></div>`

CSS:

```
.navBar {  
    width: 600px;  
    margin: 0 auto;  
    border: green solid 5px;  
}
```



# Playing with Borders

```
<div class="menuBar">  
  About  
  <div class="greenArrow"></div>  
  <div class="navBar"></div>  
</div>
```

About

# Playing with Borders

- Let's add a menu bar with an "about" item

HTML:      <div class="menuBar">About</div>

CSS:

```
.menuBar {  
    background-color: #99C299;  
    margin: 0 auto;  
    width: 600px;  
    color: white;  
    font-weight:bold;  
    padding: 5px 0px;  
}
```



# Playing with Borders

```
<html>
  <head>
    <title>Border Experiments</title>
    <style>
      .greenArrow
      {
        width:1px;
        height:0px;
        border-top:none;
        border-bottom: green solid 20px;
        border-left: transparent solid 20px;
        border-right: transparent solid 20px;

      }
      .navBar
      {
        width: 600px;
        margin: 0 auto;
        border: green solid 5px;
      }
      .menuBar
      {
        background-color: #99C299;
        margin: 0 auto;
        width: 600px;
        color: white;
        padding: 5px 0px;
      }
    </style>
  </head>
```

# Playing with Borders

```
<body>
  <div class="menuBar">About
    <div class="greenArrow">
    </div>
    <div class="navBar"></div>
  </div>
</body>
</html>
```

# Playing with Borders

- Add it all together!

HTML:

```
<div class="menuBar">About
  <div class="greenArrow"></div>
  <div class="navBar"></div>
</div>
```

We have all the CSS we need already defined!



# Using CSS in Tumblr

# Another HTML tag: **blockquote**

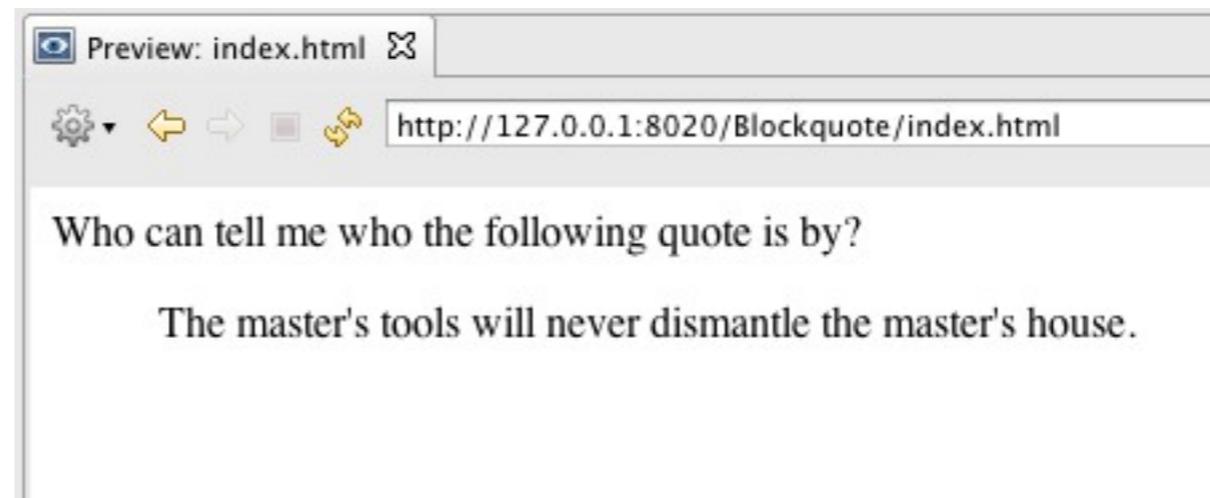
- One HTML tag we have not covered is **blockquote**
  - Blockquote's default style is to have the text indented over by one tab:

```
<body>
```

Who can tell me who the following quote is by?

```
<blockquote>The master's tools will never  
dismantle the master's house.</blockquote>
```

```
</body>
```



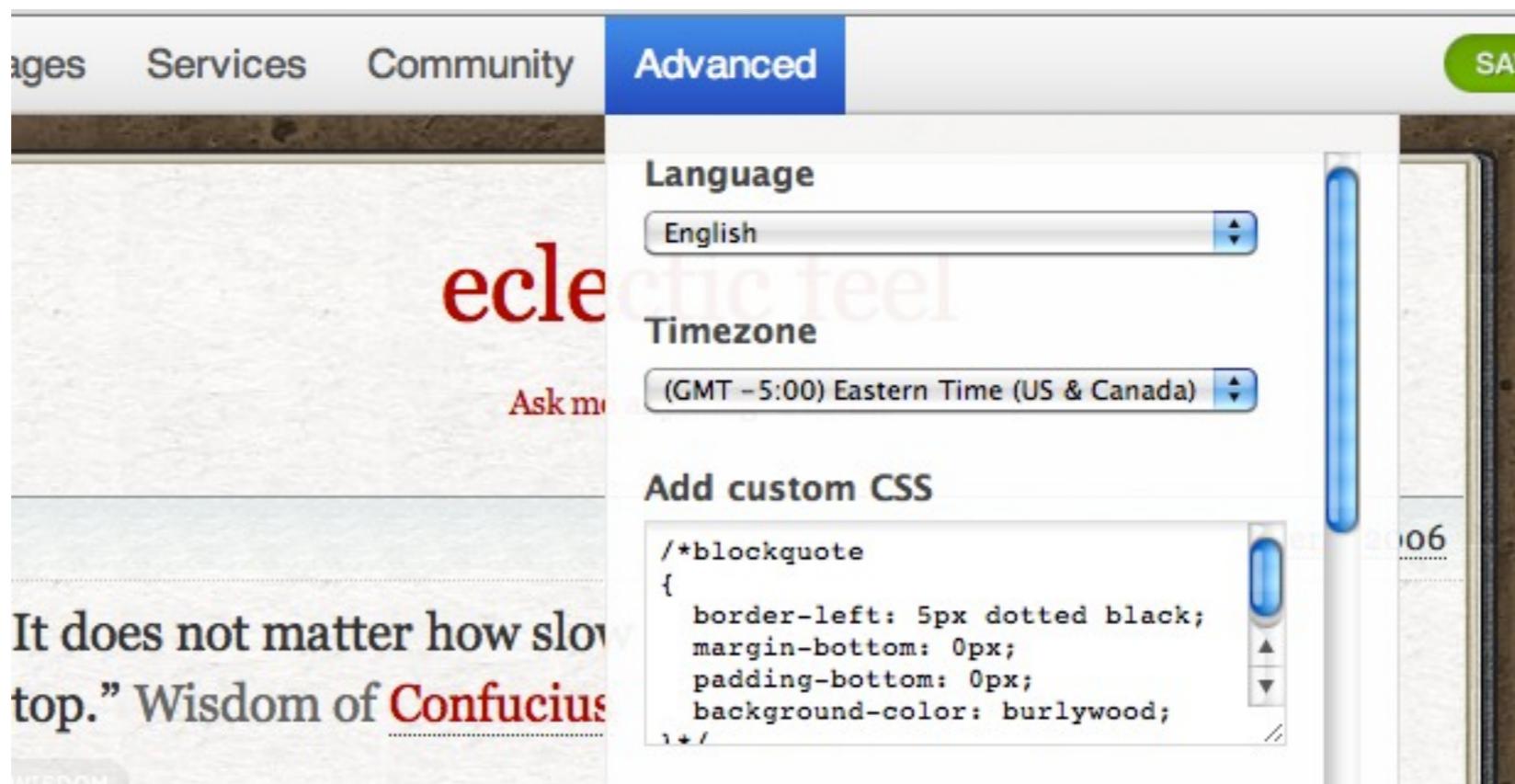
# Changing default styles in Tumblr

- To change a CSS element selector's styling in Tumblr, you can go to:
  - Account > Preferences > Customize your Blog
  - Then go to Advanced > Add custom CSS



# Changing default styles in Tumblr

- For example, if you don't like the way the **blockquote** elements on your tumblr are styled, you can add a new **blockquote** element selector in this field, and it will override the default.



# Changing default styles on Wordpress.org

- You can only change the stylesheet of a wordpress theme if you are using wordpress.org or hosting wordpress on your own site.
- If you are using the free wordpress.com, you have to pay \$15/year to be able to customize the CSS.

# CSS Grouping & Nesting

# 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>

# Fixed Layout

# Fixed Layout with Absolute Positioning

- Let's create a three-column layout that uses Absolute Positioning in CSS
- Try and follow along with the slides.
- If you get stuck, the completed code is at: <http://jsfiddle.net/Carbh/>

# Fixed Layout with Absolute Positioning

- Let's create a three-column layout that uses Absolute Positioning in CSS
- We'll begin by adding a header, wrapper and footer div to the HTML:

```
<body>
  <div id="header">header!</div>
  <div id="wrapper">

  </div>
  <div id="footer">footer!</div>
</body>
```

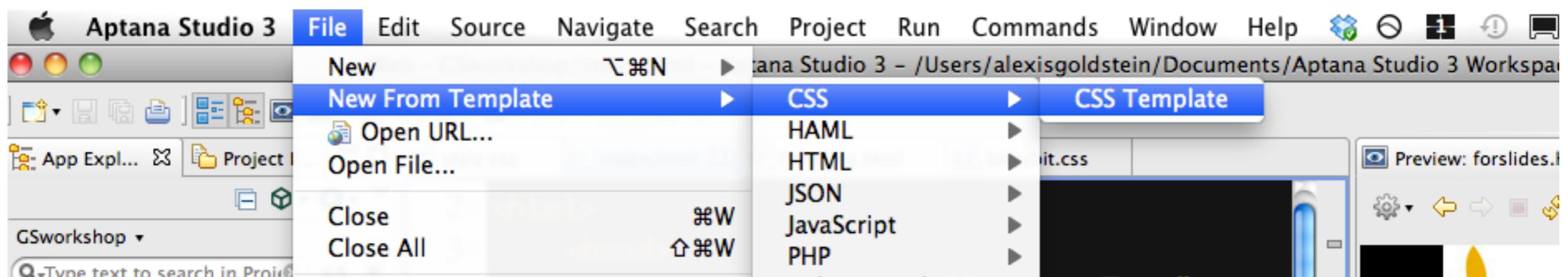
# Adding Three Columns

- Next, inside the **wrapper** div we created, we'll add three additional divs:

```
<body>
  <div id="header">header!</div>
  <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>
  <div id="footer">footer!</div>
</body>
```

# Using an External Stylesheet

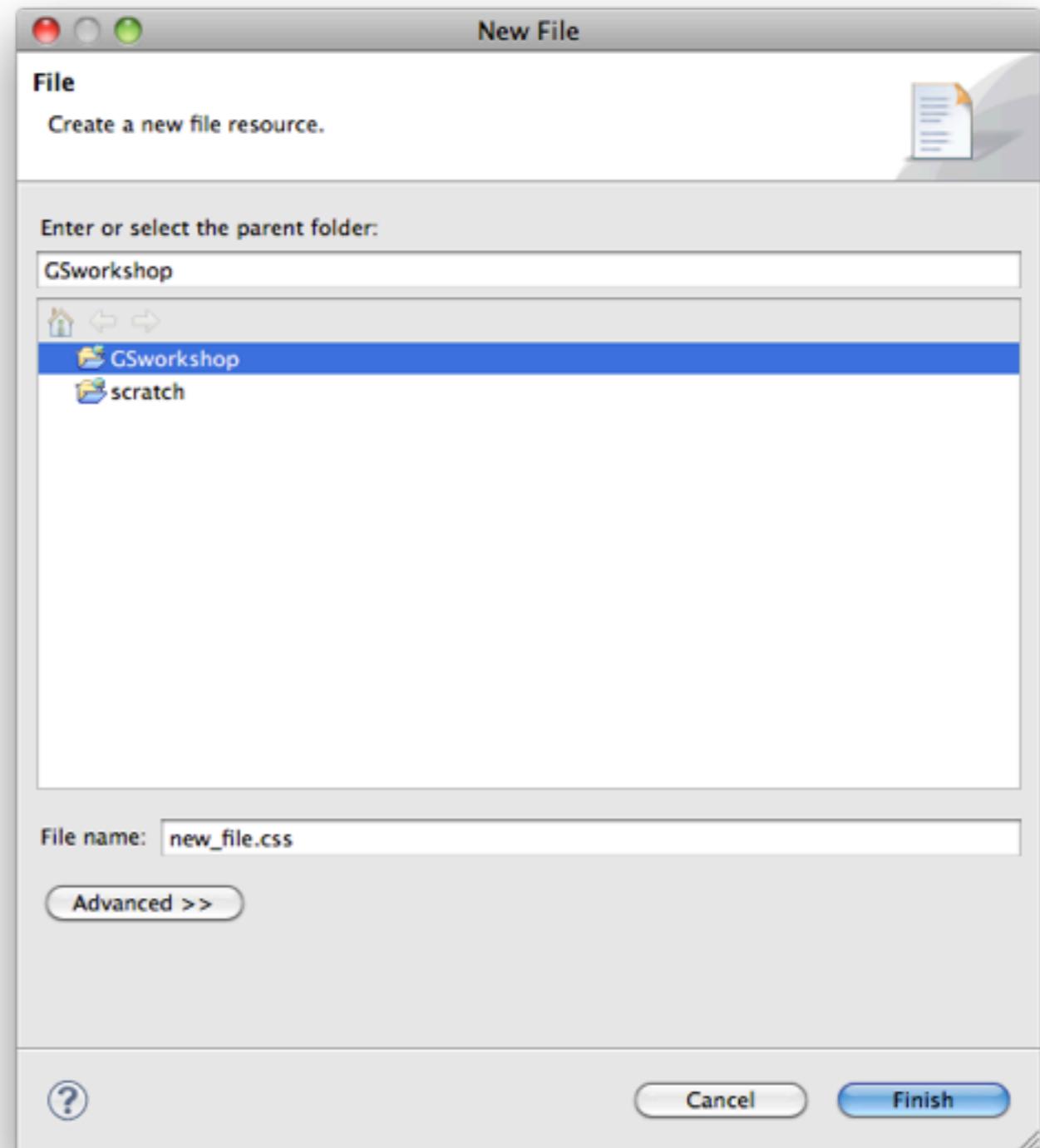
- Now that we've added the basic HTML, we need to add a new CSS file, and link the HTML page to it.
- To add a new CSS file to our project, go to **File > New from Template > CSS > CSS Template**



# Creating a separate CSS file

A window will open asking you:

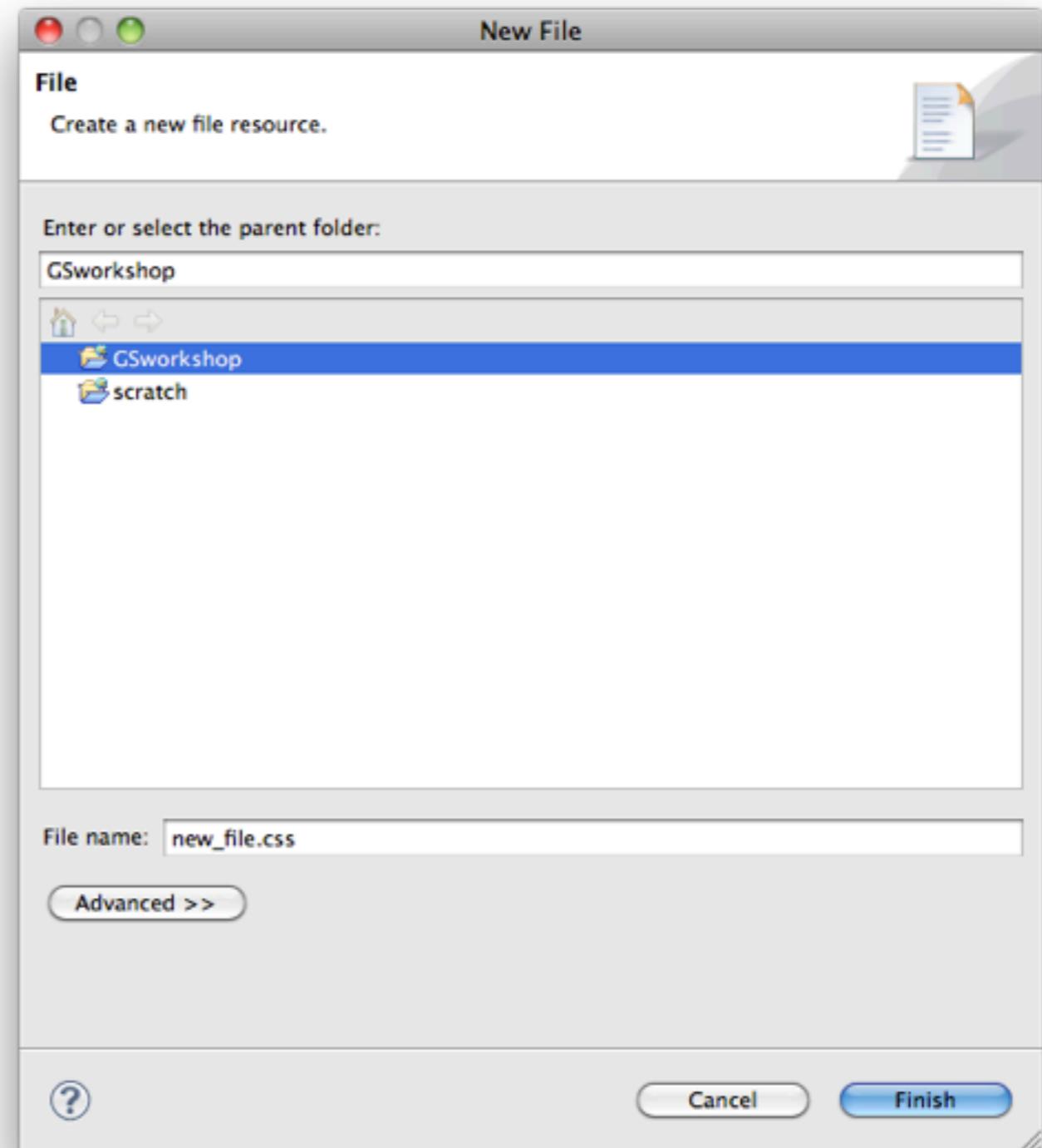
- Which project to add the file to
- What to name the file
  - Make sure you add the new file to the same project you've been working on



# Creating a separate CSS file

A window will open asking you:

- Which project to add the file to
- What to name the file (I suggest “style.css”)
  - Make sure you add the new file to the same project you’ve been working on



# Using an External Stylesheet

- Now that we have the external CSS file, we need to link our HTML page to it.

```
<head>
    <title>New Web Project</title>
    <link rel="stylesheet" href="style.css">
</head>
```

# Styling the Header and Footer

- Now, let's begin to add some styles to the divs, so we can see them.
- First, we'll style both the **header** and the **footer** divs to have a background-color and a specific size:

```
#header, #footer
{
    background-color: beige;
    height: 50px;
    width: 440px;
    padding: 5px;
}
```

header!

I'm the left column  
I'm the middle column  
I'm the right column  
footer!

# Styling the Wrapper

- We are going to style the **wrapper** div to use **position:relative;**
- This will create a new “containing block”
- That means that any elements positioned *inside* this div will use **wrapper** as the starting location
  - Recall that the default starting location is the top left corner of the browser window

# Styling the Wrapper

- We are going to style the **wrapper** div to use **position:relative;**
- Any elements positioned *inside* this div will use **wrapper** as the starting location

```
#wrapper
{
    width: 100%;
    position: relative;
    top: 10px;
}
```

# Styling the Three Columns

- We'll give all of the divs that are nested inside the **wrapper** div the same base CSS.
- We'll do this using the **nesting selectors** concept we discussed last week.
- This CSS will style *all divs that live inside the div id="wrapper" element.*

```
#wrapper div {  
    display:inline;  
    position:absolute;  
    padding:5px;  
}
```

# Styling the Three Columns

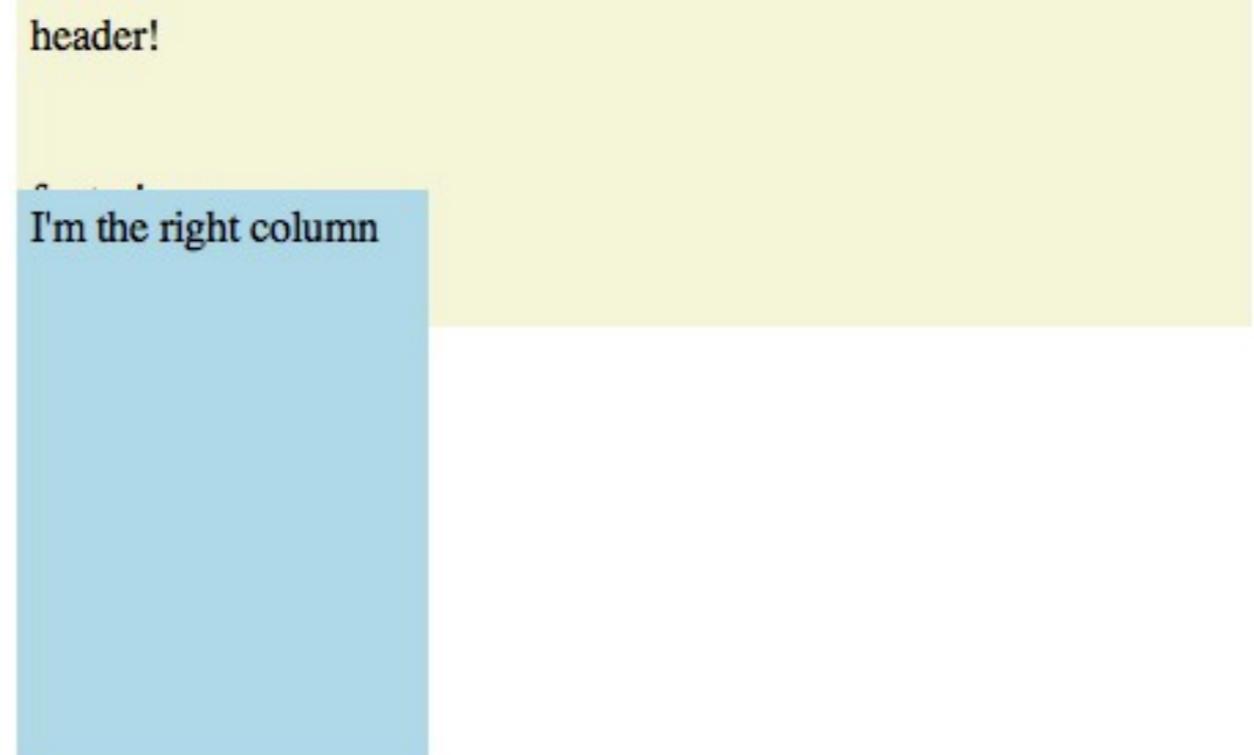
- We make all the divs `display:inline` so they will show up one right after the other horizontally.
- We give them a default `position:absolute` because we'll be using the **left** property to put each column in a different location.

```
#wrapper div {  
    display:inline;  
    position:absolute;  
    padding:5px;  
}
```

# Styling the Outer Columns

- Now let's give some color to the left and right divs.
- We also give the left and right divs a specific size.
- They will be positioned on top of one another--we'll fix that next.

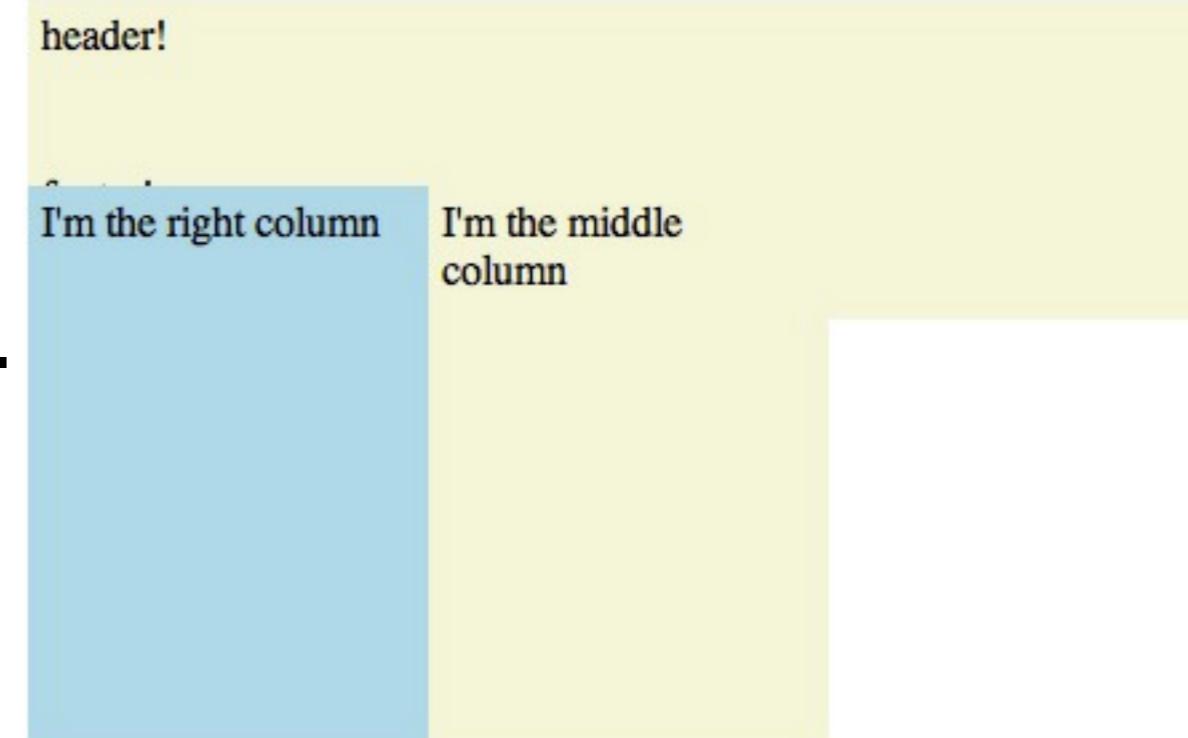
```
#left, #right
{
    background-color: lightblue;
    width: 140px;
    height: 200px;
}
```



# Styling the Middle Column

- Now let's give a different color to the middle div.
- We also give it a specific size.
- Next, we use the **left** property to move the middle div **150px** to the left of its parent element--the **wrapper** div.

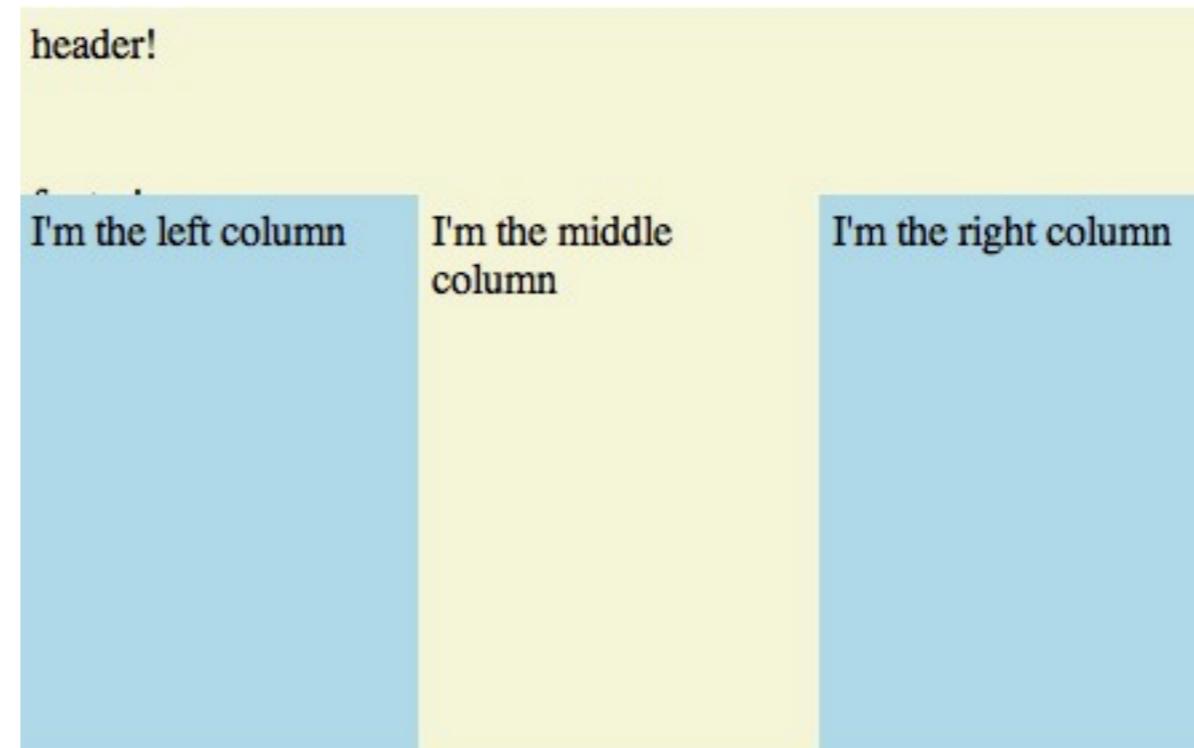
```
#middle
{
    position: absolute;
    left: 150px;
    width: 140px;
    height: 200px;
    background-color: beige;
}
```



# Styling the Right Column

- Now let's position the right div.
- We position it **300px** from the left of the containing **wrapper** div

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



# Styling the Footer

- We already set the common styles that the header and footer divs share. Now let's give the footer an unique position
- We position it **290px** from the left of the containing **wrapper** div

```
#footer  
{  
    position: absolute;  
    top: 298px;  
}
```

header!

I'm the left column

I'm the middle column

I'm the right column

footer!

# Liquid Layout

# 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/Carbh/>
- Finished file: <http://jsfiddle.net/fxN3Z/>

# 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

- We need to *remove* the default absolute positioning from the divs inside the **wrapper** element
- We also need to *remove* the `display:inline`

Before:

```
#wrapper div {  
    display:inline;  
    position:absolute;  
    padding:5px;  
}
```

After:

```
#wrapper div {  
    padding:5px;  
}
```

# Changing our Fixed Layout to Liquid

- We need a new selector for the left div:

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

# Changing our Fixed Layout to Liquid

- Changing the CSS:
  - #right should not be **position:absolute** anymore. Let's change #right instead to have **float:right;**

Before:

```
#right
{
    position: absolute;
    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:

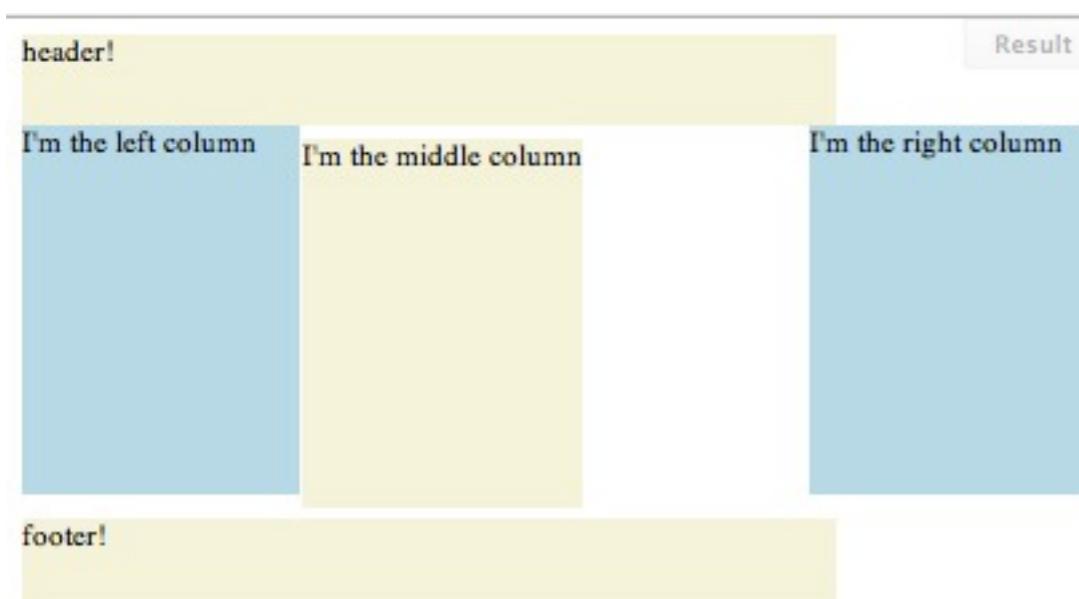
```
#middle
{
  position: absolute;
  left: 160px;
  height: 200px;
  background-color: beige;
}
```

After:

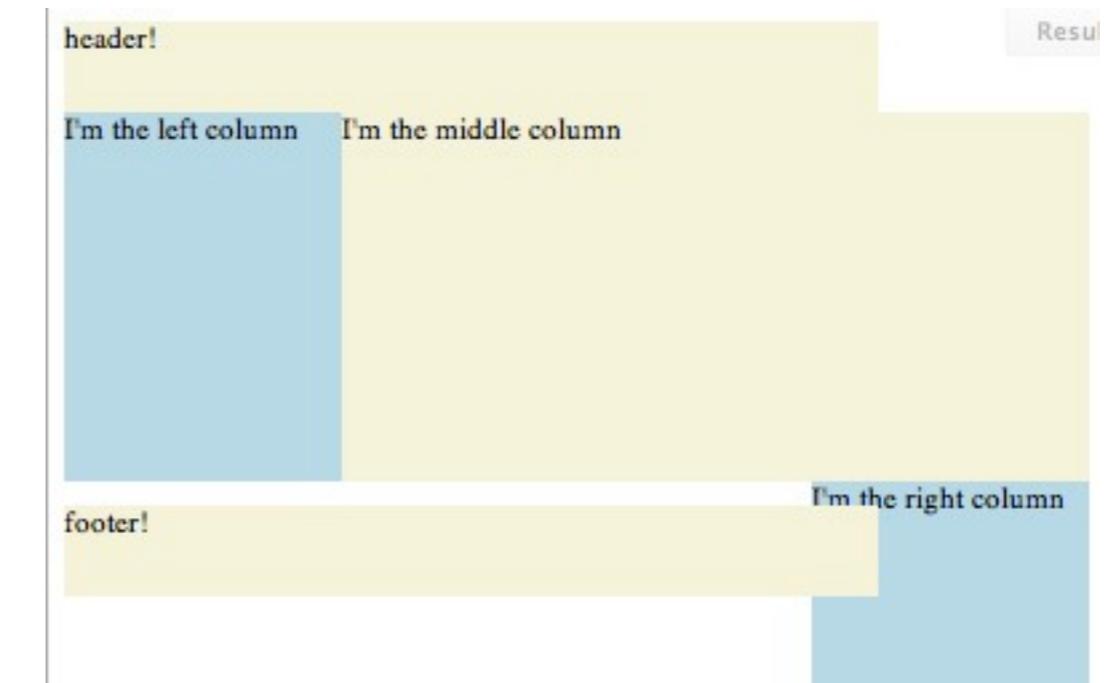
```
#middle
{
  height: 200px;
  background-color: beige;
}
```

# 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>

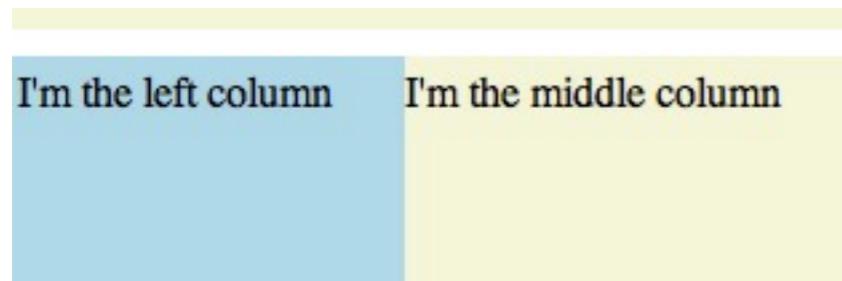
  <div id="middle">
    I'm the middle column
  </div>
```

# A Few Tweaks

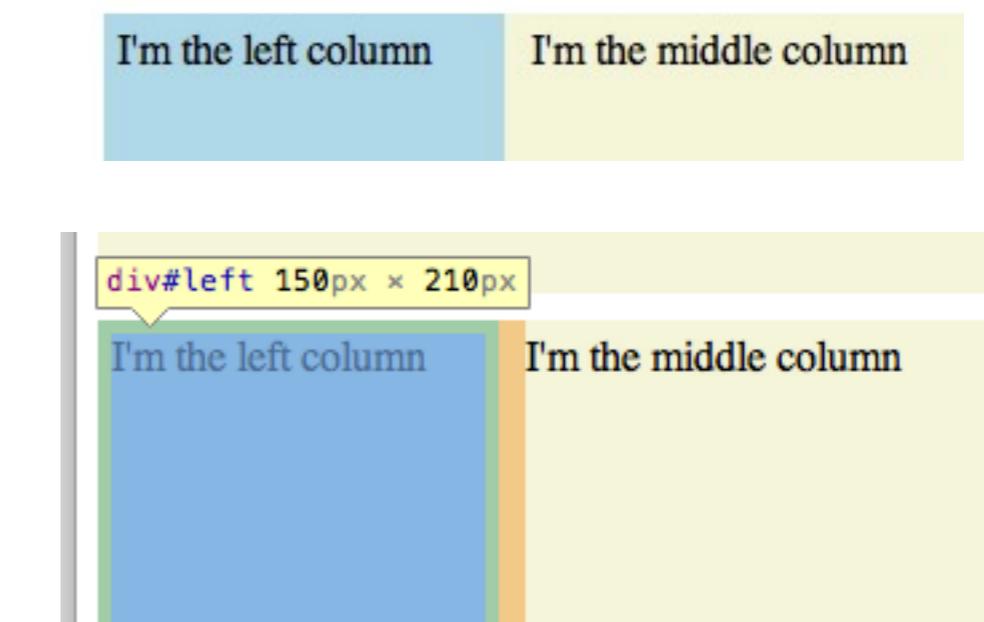
- Adding a margin to the right of the **left** column ensures the middle div's text isn't right next to the **left** column.

```
#left {  
    float:left;  
    margin-right:10px;  
}
```

Before:



After:



# A Few Tweaks

- Removing the position from the **footer** and simply using **margin-top** to space it away from the wrapper

Before:

```
#footer
{
    position: absolute;
    top: 298px;
}
```

After:

```
#footer
{
    margin-top: 20px;
}
```

# A Few Tweaks

- Taking away the default margin on the **body** that's added by the browser will ensure the header and footer are truly 100% of the page width.

```
body {  
    margin:0px;  
}
```

# More Code Examples

# Imitation is the highest form of flattery

The screenshot shows the homepage of the MoMA PS1 website. At the top right, there is a navigation bar with links for CALENDAR, BUY TICKETS, JOIN, ABOUT, and A-Z ART. Below the navigation is a large image of a painting by Yayoi Kusama, featuring a red figure with white dots against a yellow background. To the left of the painting is a video thumbnail showing a white ceramic pitcher on a table, with text overlay: "FILM | VIEW ALL » Modern Mondays ONGOING". Below the main image are four colored boxes: a green box for "Calendar" showing a photo of a jeep and people; a purple box for "Donate Online" showing a photo of an art gallery; a yellow box for "Shop" with a photo of a coffee cup and the text "KITCHEN DESIGN TO LOVE"; and a blue box for "VISIT" with a photo of a person walking. At the bottom, there is a red footer bar with links for "Sign in here", "Welcome. Are you...?", "Register to text content to a mobile number", and a search bar.

CALENDAR BUY TICKETS JOIN ABOUT A-Z ART

FILM | VIEW ALL »  
**Modern Mondays**  
ONGOING

THE COLLECTION | VIEW ALL »  
**Prints and Illustrated Books**

Calendar

Donate Online

Shop

KITCHEN DESIGN TO LOVE

**MoMA PS1**

VISIT EXPLORE LEARN SUPPORT SHOP

+ Sign in here

Welcome. Are you...?

Register to text content to a mobile number

Search

# 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>
      <img id="momaLogo" s
    </li>
    <li>VISIT</li>
    <li>EXPLORE</li>
    <li>LEARN</li>
    <li>SUPPORT</li>
    <!-- take away border from the last item-->
    <li style="border:none;">SHOP</li>
  </ul>
</div>
```

. MoMA PS1

- VISIT
- EXPLORE
- LEARN
- SUPPORT
- SHOP



# Styling the navigation: part I

```
#nav
```

```
{
```

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

```
}
```



# Styling the navigation: part 2

```
#nav ul, #nav 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;
}
```



# Styling the navigation: part 3

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

}
```



# 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;  
}
```



# 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: black;
```

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

```
  height: 70px;
```

```
}
```

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

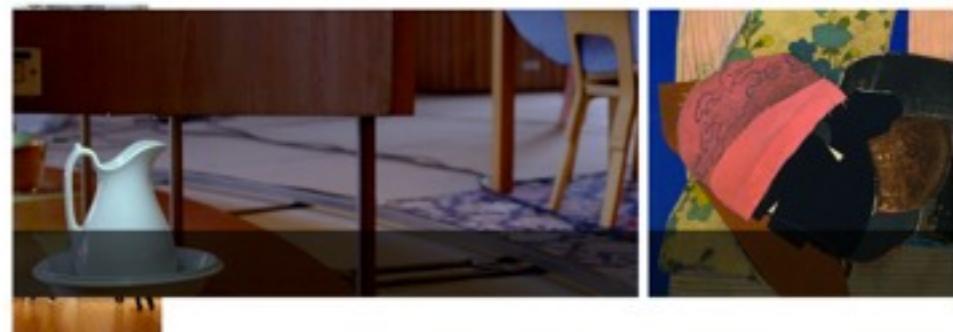
I'm sneaking in some CSS3;  
rgba() is a CSS3 property

# 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>
.
```



# Styling the bottom 3 boxes: part I

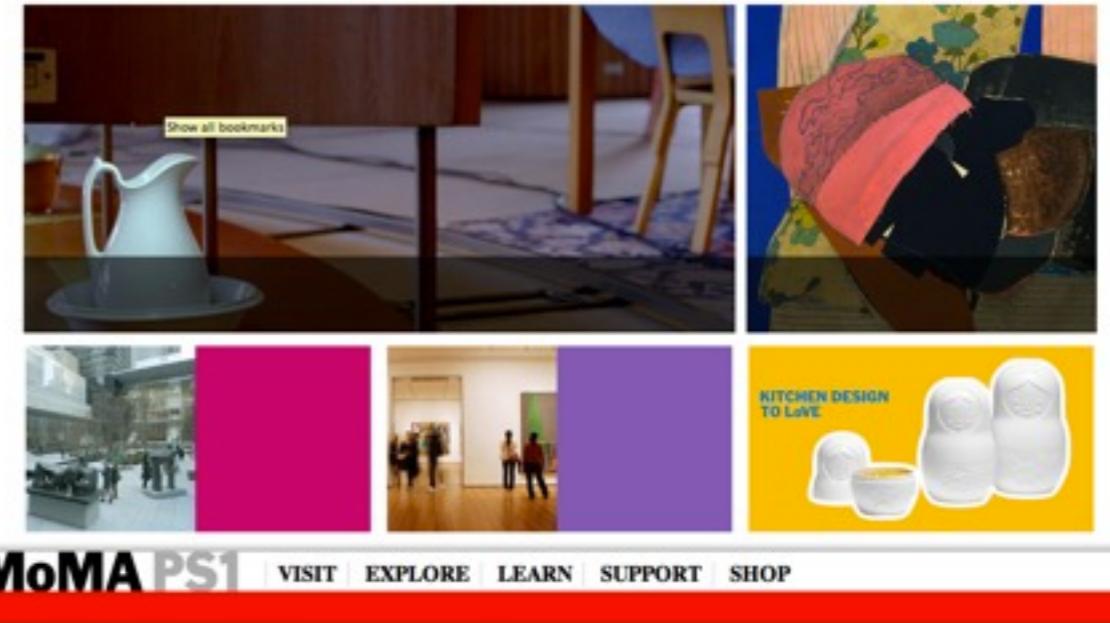
```
.bottomBox
{
    width: 315px;
    height: 170px;
    border: 2px solid white;
}
```

# 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;
}
```



MoMA PS1 VISIT EXPLORE LEARN SUPPORT SHOP

# Changing as we hover!

```
#leftBox:hover
```

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

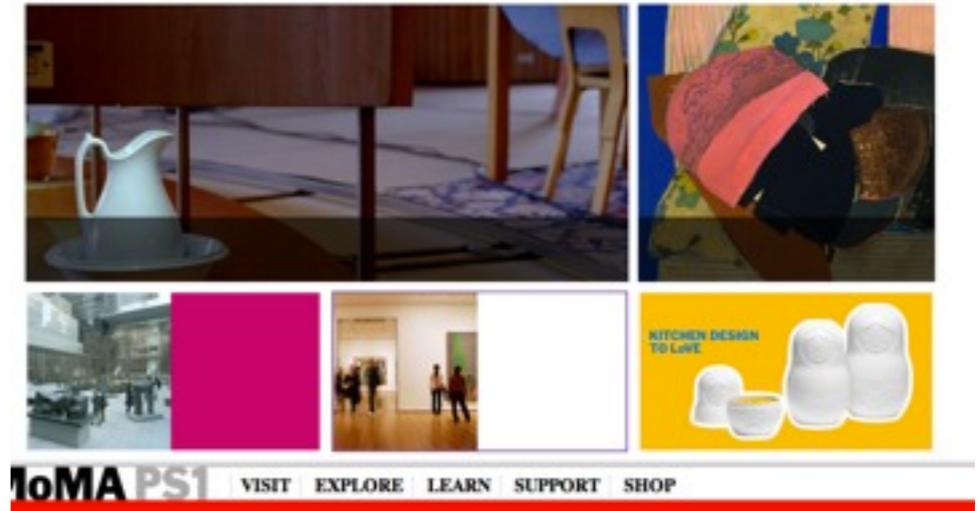
```
}
```

```
#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](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-canvastext 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", sans-serif;
}
```

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

# Which font file to use?

It is suggested that in the **url** property, you point to a **woff** version of the font.

- **woff** is the most supported font type.
- You can use fontsquirrel to convert the font file you have to woff if needed.

```
@font-face
{
    font-family: "Bitstream Vera Serif Bold";
    src:
        url("http://yoursite.com/VeraSeBd.woff") format('woff');
}

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

# Which font file to use?

Before iOS 4.2, the only supported font type was **svg**.

- You will likely want to also provide an svg file
- The SVG specification contains an additional hash tag as a unique identification number. This is necessary as SVG files may contain several fonts.

```
@font-face
{
    font-family: "Bitstream Vera Serif Bold";
    src:
        url("http://yoursite.com/VeraSeBd.woff") format('woff'),
        url('Graublauweb.svg#svgGraublauweb') format('svg'); /* Legacy
ios */;
}

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

# Supporting all browsers with @font-face

- To also support IE 8 and earlier, here is a complete example:

```
@font-face {  
    font-family: Graublauweb;  
    src: url('Graublauweb.eot'); /* IE9 Compatibility Modes */  
    src: url('Graublauweb.eot?') format('eot'), /* IE6-IE8 */  
        url('Graublauweb.woff') format('woff'), /* Modern Browsers */  
        url('Graublauweb.ttf') format('truetype'), /* Safari, Android,  
iOS */  
        url('Graublauweb.svg#svgGraublauweb') format('svg'); /* Legacy  
iOS */  
}
```

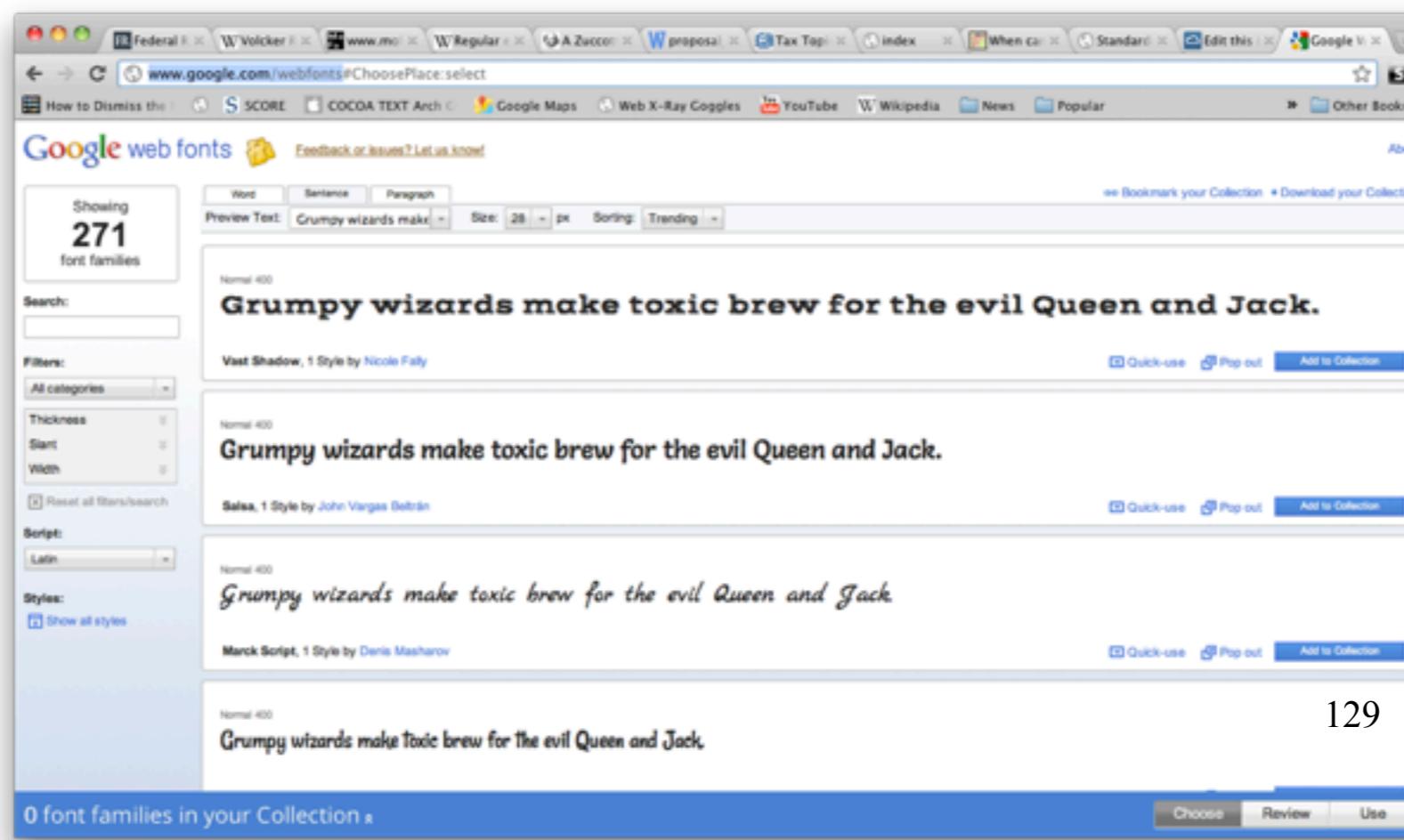
- The **eot** font name is followed by a question mark.
  - This fools IE into thinking the rest of the string is a query string and loads just the **eot** file.
  - Read more at: <http://www.smashingmagazine.com/2011/03/02/the-font-face-rule-revisited-and-useful-tricks/>

# 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>

For now, I do suggest taking a look at Google's WebFonts:  
<http://www.google.com/webfonts>



# Using Google Web Fonts

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
    <title>fontTest</title>
    <meta name="generator" content="Studio 3 http://
aptana.com/">
    <meta name="author" content="edgefreak">
    <link href='http://fonts.googleapis.com/css?family=Exo'
rel='stylesheet' type='text/css'>
    <!-- Date: 2012-02-09 -->
    <style>
      body {
        font-family: 'Exo', sans-serif;
      }
    </style>
  </head>
  <body>
    helooo!!!!
  </body>
</html>
```

# Font loading

Most browsers won't show any text before all Web fonts are imported.

- So your non-text elements will usually show up slightly before the text itself, if the whole page uses `@font-face`.
- Firefox, however, will display the default (typically Times New Roman) font first...
- ...and then re-render once the Web Font is loaded.
- To reproduce Firefox's default behavior in all other modern browsers, you can use Google's **Web Font Loader** JavaScript library:
  - [http://code.google.com/apis/webfonts/docs/webfont\\_loader.html](http://code.google.com/apis/webfonts/docs/webfont_loader.html)

# 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

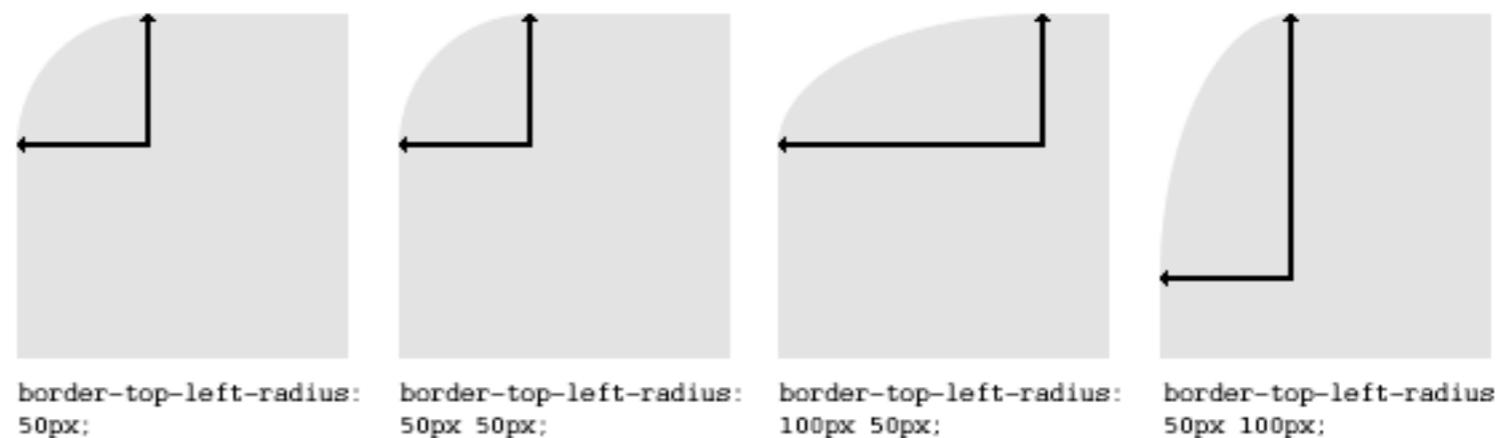
Why do we use

- 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;
    /* safari and chrome */
    -webkit-box-shadow: black 0px 5px 5px;
    -ms-box-shadow: black 0px 5px 5px;
    /* fallback */
    box-shadow: black 0px 5px 5px;
}
```

A diagram illustrating the components of a drop shadow. It shows three colored rounded rectangles: a light blue one labeled '0px' for horizontal offset, a light purple one labeled '5px' for vertical offset, and a light green one labeled '5px' for blur radius. Blue arrows point from the labels 'horizontal offset' and 'vertical offset' to their respective colored boxes. A green arrow points from the label 'blur radius' to the green box.

# 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;
}
```

**Inset Shadow**

# Text Shadows

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

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);

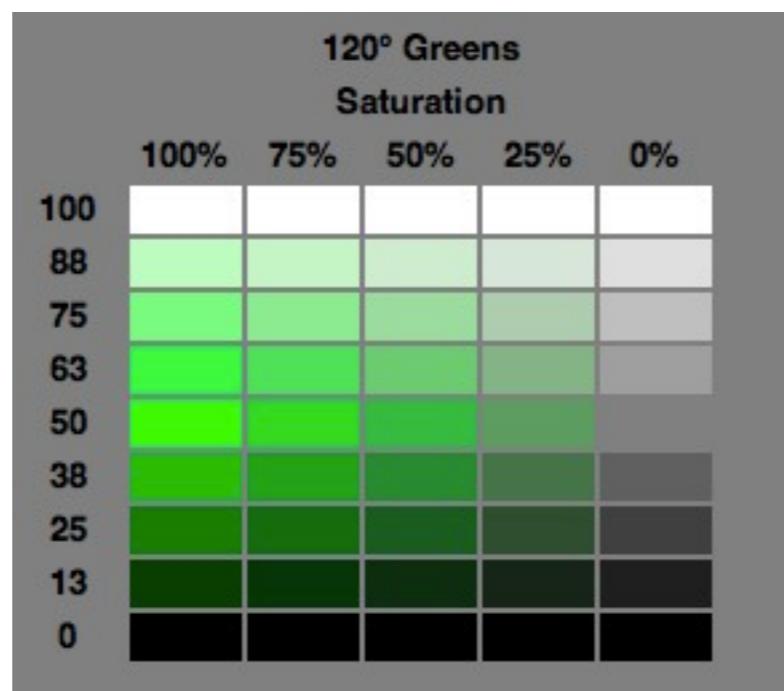
**rgb: old way, no alpha**

**rgba: new way, with alpha**

# Color: hsl and hsla

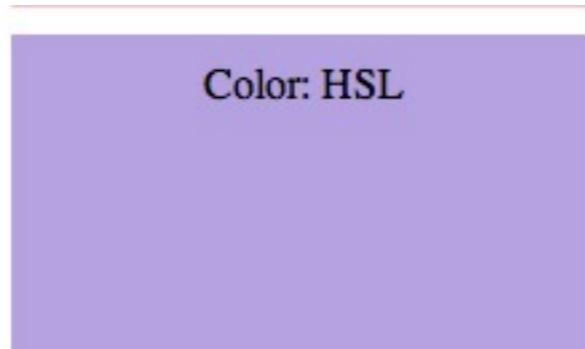
- HSL = Hue, Saturation and Lightness
- HSLA = Hue, Saturation, Lightness **and Alpha**
- **Syntax:**
  - **hsl( hue--in degrees from 0-359, saturation--in % from 0-100%, lightness--in % from 0-100%)**
  - **hsla( hue--in degrees from 0-359, saturation--in % from 0-100%, lightness--in % from 0-100%,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

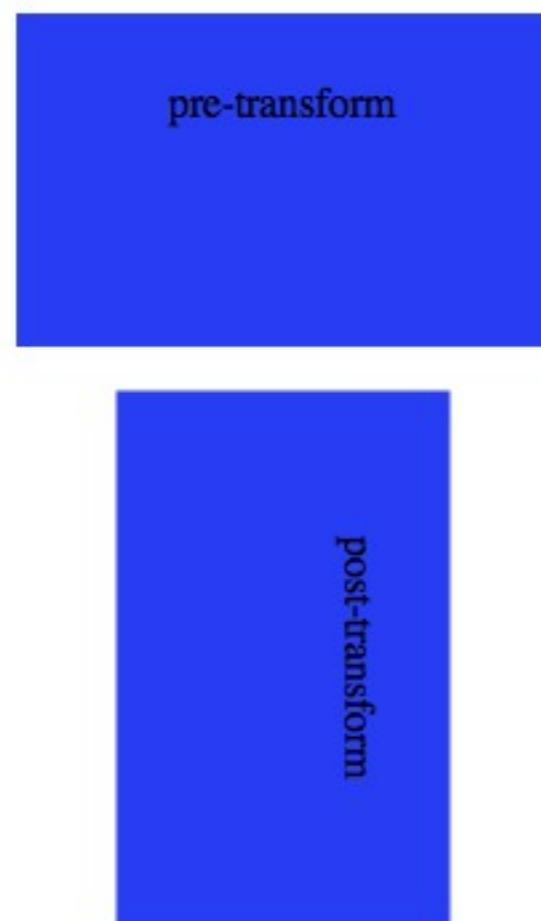
- You can use CSS transforms to rotate or scale elements on your page.
- We used to need JavaScript in order to do stuff like this!
- Our options: **rotate**, **scale**, **skew** and **translate**.

# CSS Transforms

- Our JSFiddle, starting file: <http://jsfiddle.net/8etSs/l/>
- finished file: <http://jsfiddle.net/fiddlefiddle/8etSs/l8/>

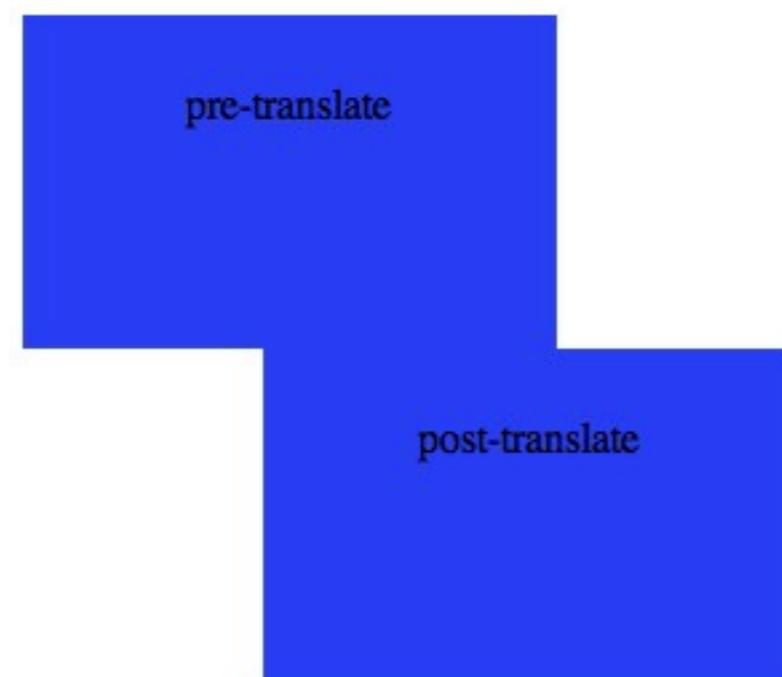
# CSS Transforms: Rotate

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



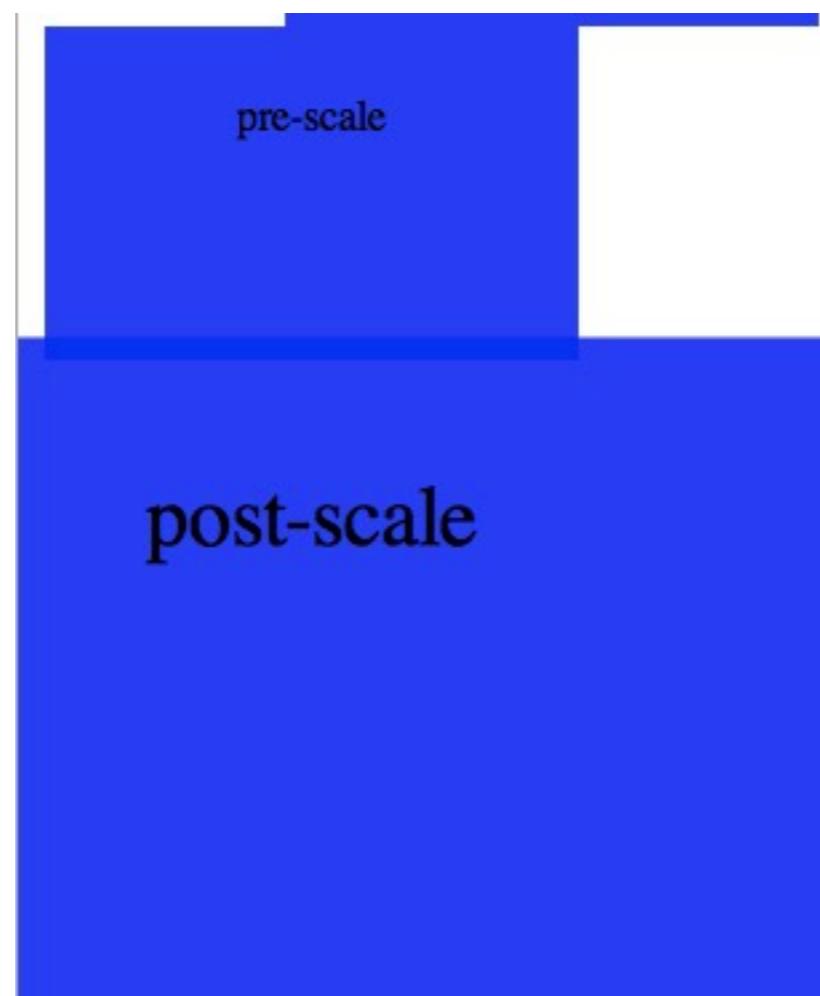
# CSS Transforms: Translate

- `-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

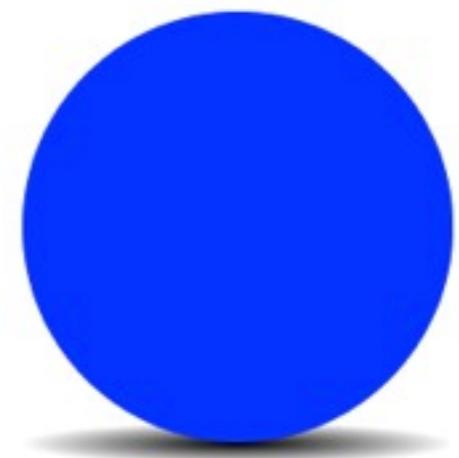
- /\* make a picture 1.25 times its normal size\*/  
    -webkit-transform: scale(1.25);  
    -moz-transform: scale(1.25);  
    -o-transform: scale(1.25);

# 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/>

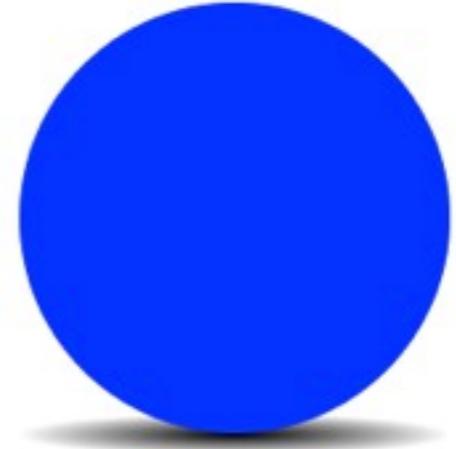


# CSS Transforms: another example

I wrote up a blog entry to describe the shadow trick we did in class more clearly.

You can find that here:

[http://www.fuckyeahtml5.com/  
2011/06/tweak-your-shadows-  
with-css3-gradients-and-  
transforms/](http://www.fuckyeahtml5.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](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;
```

Normal opacity of the painting is 50%

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

```
-webkit-transition : all 1.0s;  
-moz-transition : all 1.0s;  
-o-transition : all 1.0s;
```

```
}
```

```
.paintings img:hover
```

```
{
```

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

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

```
/* 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
- hllo

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
- hllo

text

text

text

text

text

text

Finished product: <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)

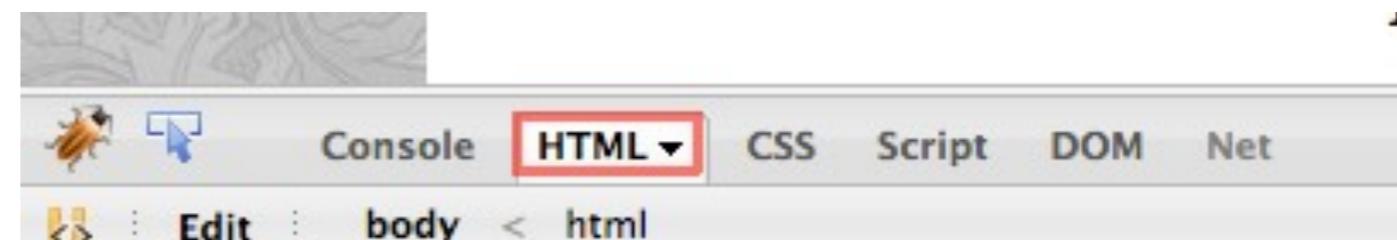
With Firebug open:

1. Click the HTML tab

2. Click on <body>

3. Note the Style details that come up in the right-hand window.

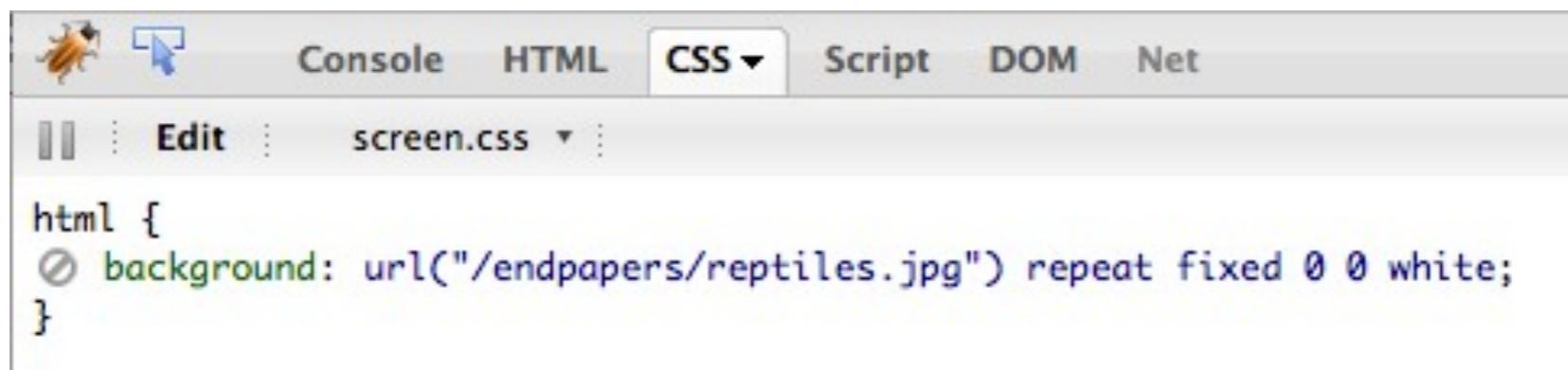
Firebug should display two sets of styles for the **body** element: one set in the file base.css and another set of styles in the file screen.css:



A screenshot of the Firebug interface. The left pane shows the HTML structure with the body element selected. The right pane shows the Style tab with two sets of CSS rules for the body element. The first set, from screen.css (line 2), includes margin: 1em auto;, max-width: 33em;, and padding: 3em;. The second set, from base.css (line 13), includes background-color: #FFFFFF;, color: #000000;, font-family: Georgia, serif;, margin: auto;, and max-width: 33em;. Both sets include a note that they are being overridden by the first rule.

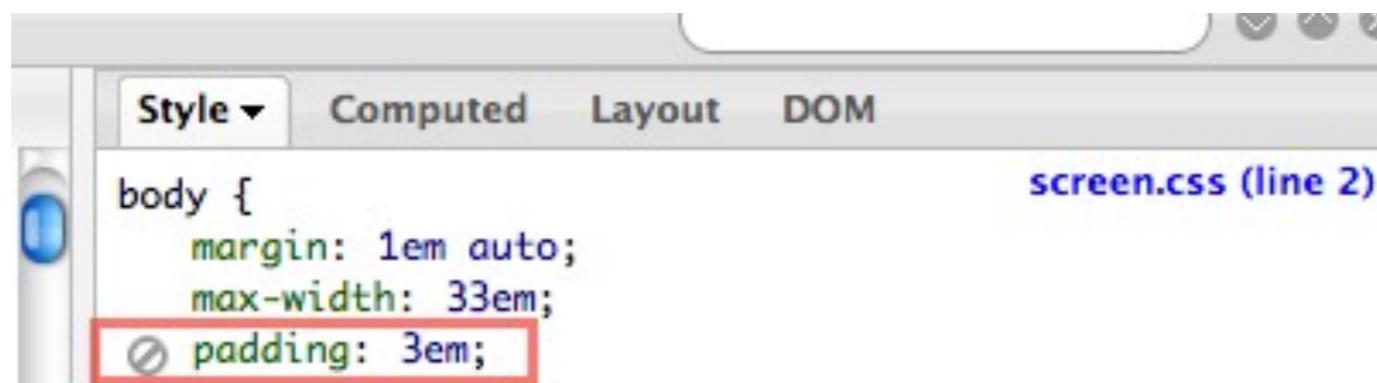
# Examining a Single Column Layout

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



```
Console  HTML  CSS ▾  Script  DOM  Net  
Edit  screen.css ▾  
  
html {  
    background: url("/endpapers/reptiles.jpg") repeat fixed 0 0 white;  
}
```

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



# Examining a Single Column Layout

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

The screenshot shows the 'Style' tab of a browser's developer tools. It displays the CSS rule for the 'body' selector from the 'screen.css' file at line 2. The rule is: `body { margin: 1em auto; max-width: 5em; padding: 3em; }`. The 'max-width' value '5em' is highlighted with a blue selection box.

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

The screenshot shows the 'Style' tab of a browser's developer tools. It displays two CSS rules for the 'body' selector. On the left, under 'base.css:13', the 'body' rule is: `body { background-color: white; color: black; font-family: Georgia, serif; margin: auto; max-width: 33em; }`. On the right, under 'screen.css (line 2)', the 'body' rule is: `body { margin: 1em auto; max-width: 33em; padding: 3em; }`. The 'margin: auto' rule in the 'base.css' panel has a red circle with a slash over it, indicating it is being overridden by the 'margin: 1em auto;' rule in 'screen.css'.

# 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>

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

# Sample Two-Column Layout

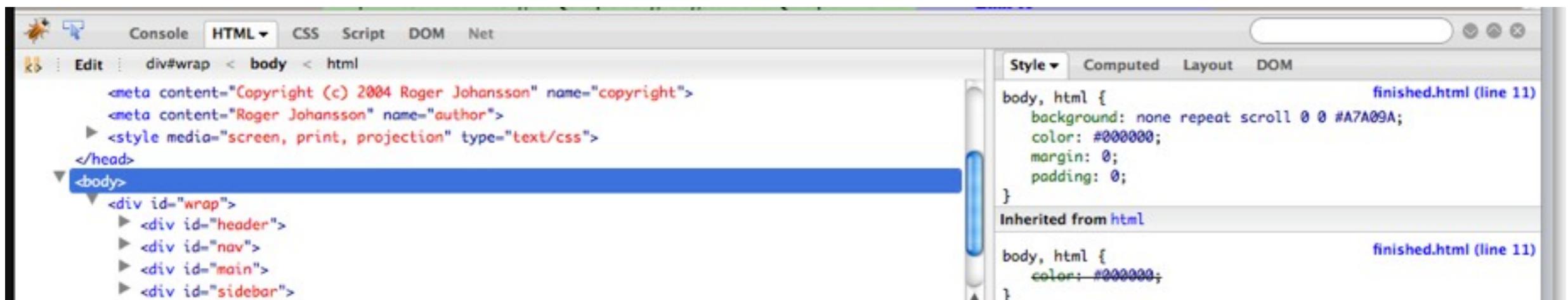
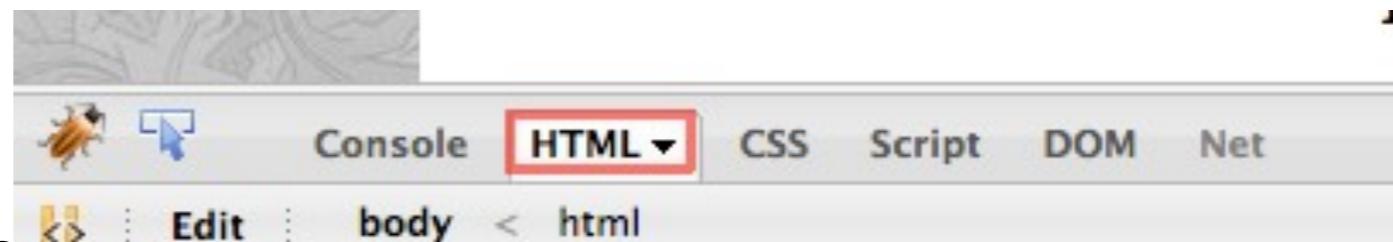
- [http://www.456bereastreet.com/lab/developing\\_with\\_web\\_standards/csslayout/2-col/finished.html](http://www.456bereastreet.com/lab/developing_with_web_standards/csslayout/2-col/finished.html)
  - shorter version of the same link: [http://bit.ly/two\\_col](http://bit.ly/two_col)

# 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.

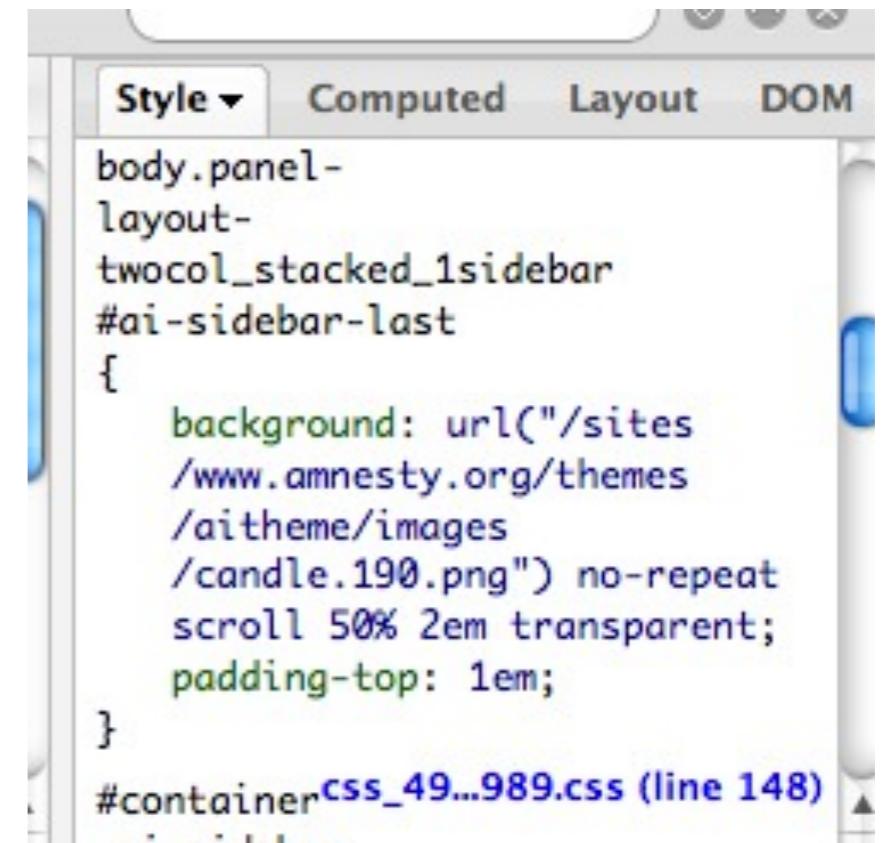
- Scroll down a bit in the style window in Firebug until you see a **background** property.

- 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.



# Examining a Three-Column Layout

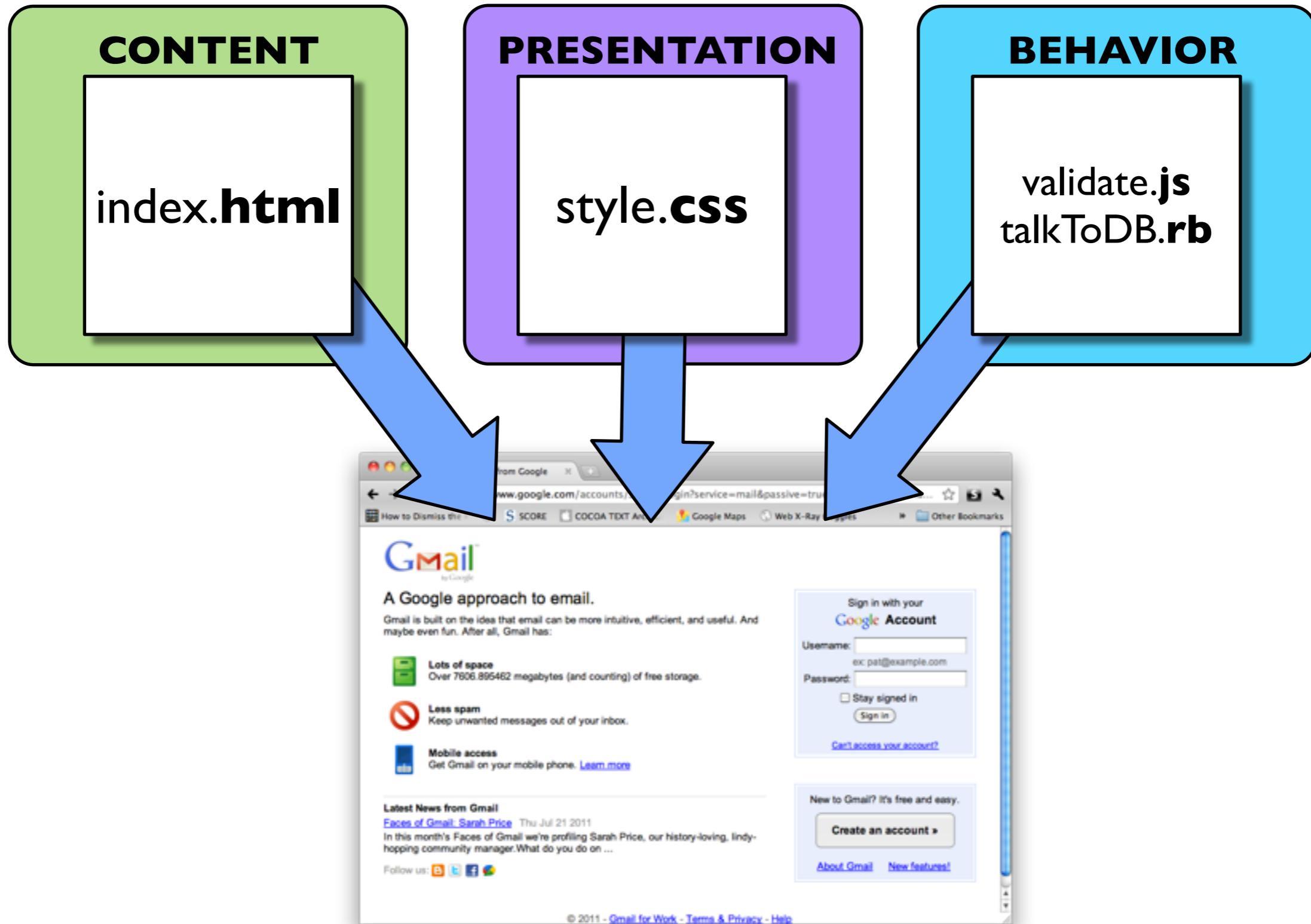
To reiterate, peeking into the guts of more complex pages requires:

- 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

# HTML + CSS = A Website



# HTML + CSS + Code = A Dynamic Website



# What really makes a website?

- Every single webpage has a URL
- URL = Uniform Resource Locator
  - aka Find this Page on the Internet
- URL includes Domain + Content
- Domain (Location)
  - <http://www.google.com>
- Content (Resources)
  - <http://www.google.com/analytics>
  - <http://www.myblog.com/i-love-websites>

# Domain Registration

# Domain Registration

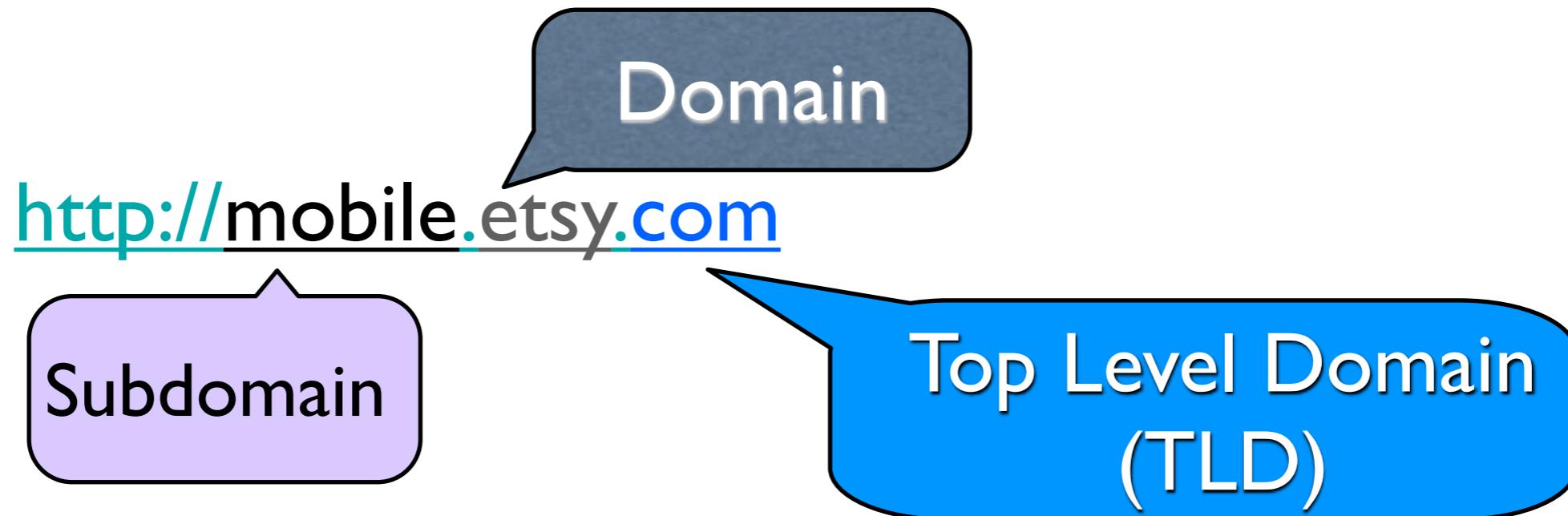
The process of claiming / reserving a domain name.

Lasts for one year, then you must renew the domain if you want to keep it.

Should cost ~\$10/year.

# URL = subdomain.domain.tld

There are several parts to every URL:



# Top Level Domains (TLD)

The most common TLDs:

- .com
- .net
- .org
- .edu
- .gov
- If possible, you should get a .com domain

# Domain Search

Domainr

<http://domai.nr>



search

Give your domain search a happy ending.

When you want a [short URL](#) or [something big](#),  
Domainr will find it, fast. Some of our favorites:  
[cr.yp.to](#), [itha.ca](#) and [jish.nu](#).

# Domain Search - Results

Domainr

Available on the App Store

codeforhumanity

■ [codeforhumanity.com](#)

■ [codeforhumanity.net](#)

**codeforhumanity.org**

■ [codeforhumanity.it/y](#)

[codeforhumani.ty](#)

■ [codeforhum.an/ty](#)

■ [codeforhu.ma/nty](#)

[codeforh.um/anity](#)

■ [codefor.hu/manity](#)

■ [code.fo/rhumanity](#)

■ [codef.rhuman.it/y](#)

■ [cdfrh.mn/ty](#)

## codeforhumanity.org

This domain is taken.

[Make Offer](#)

Register at:

[Go Daddy](#)  
[Name.com](#)  
[iWantMyName](#)  
[101domain.com](#)  
[Namecheap](#)  
[More...](#)

TLD: .org IDN

[Wikipedia](#)  
[IANA](#)  
[More Info](#)

Tools

[Visit Site \(www\)](#)  
[WHOIS](#)  
[Open Site Explorer](#)



### Drag & Drop for WordPress

A revolutionary new way to build pro sites; faster & easier, than ever before.

# Country Code TLDs

- Each country has a top level domain.
- Each country has its own requirements and fees for registering a site using their TLD.
  - You may need to reside in the country
- Some popular TLDs that are country codes:
  - .ly (Libya)
  - .me (Montenegro)
  - .us (United States)

# Libyan Domain Registration

The screenshot shows the LibyanSpider Client Area interface. At the top, there's a navigation bar with links for Domains, Hosting, Web Design, Software, About Us, and Client Login. Below the navigation is a blue header bar with the text "Client Area". The main content area has a breadcrumb navigation "Portal Home > My Domains". It features a search bar with "Enter Domain to Find" and a blue arrow button. Below the search bar, it says "1 Records Found" and "Show: 10 25 50 100 All". A table displays the found record:

Registration Date	Domain	Next Due Date	Price	
12/06/2010	<a href="#">paperless.ly</a>	11/06/2012	\$75.00 USD	<a href="#">View Details</a>

At the bottom of the page, it says "Page 1 of 1". On the left side, there's a sidebar with links: Register Domain, Bulk Domain Transfer, My Domains (which is selected), My Web Hosting & Services, My Invoices, My Support Tickets, My Profile Details, and My Emails.

# Domain Registration Companies

- Lots of choices:
  - nearlyfreespeech.net
  - domain.com
  - register.com
  - dreamhost.com
  - networksolutions.com
- It's easiest to register the domain with the same company you use to host

# Web Hosting

# A Place for your Website to Live

- Once you've secured the domain, you need a way to make it available for others to see!
- You need someone to **host your domain** in order to use a domain you've registered.
- A hosting service provides you with a **web server**.
  - The **web server** is what actually makes your **website viewable from any browser, on any network, anywhere**.

# Jargon Alert! Server



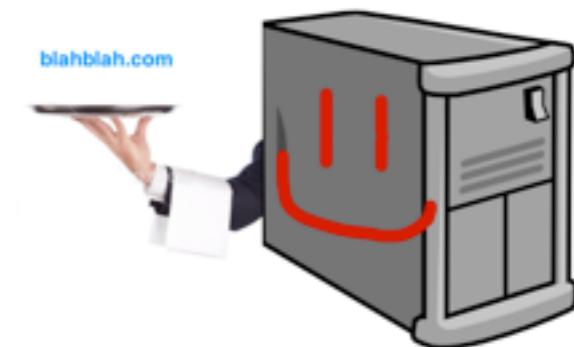
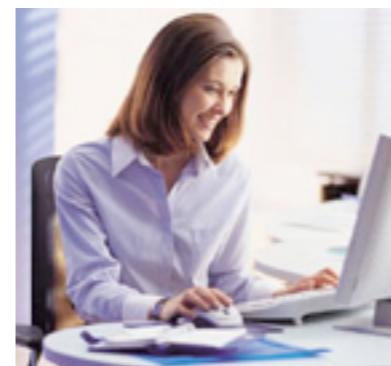
A **server** is just a computer!  
More specifically, it's a computer that does nothing but sit around and wait for you to call.  
It runs special **web server software** in order to handle your requests.



# Jargon Alert! Web Server

- Web Servers are a combination of:
  - A computer (the server)
  - Special **software** that knows how to handle website requests
- The job of a web server is to pass a website back to the **client** who requested it

When someone types in `yoursite.com`, the web server receives that request, and **serves** you back the website.



# Client vs. Server



Client

Application

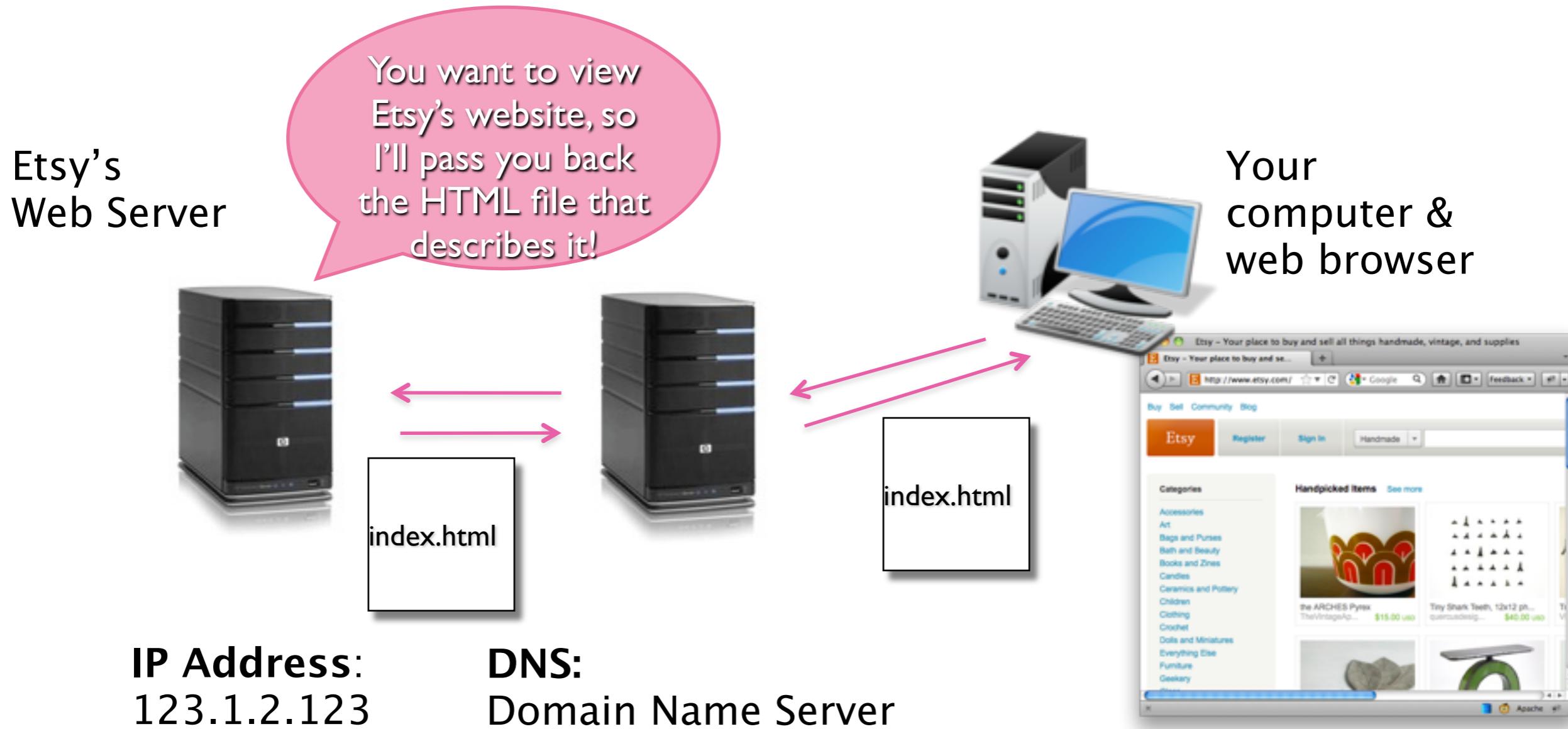
Database

Web Server



Images by Icons Land, Social Peel, and HP

# What Happens when you open a site



# The Domain Name System

- DNS: Domain Name System
  - DNS is like a phone book.
  - It takes a domain (etsy.com) and looks up the **IP address** for that domain.
  - This is the IP address of the **server** that hosts your domain.

# Domain Registrar vs Domain Name System

- Domain Registrar
  - Tracks who controls the domain (you)
- Domain Name System
  - DNS Servers track where a domain should point at any time.
  - Global phone book for domains to specific servers websites live on.

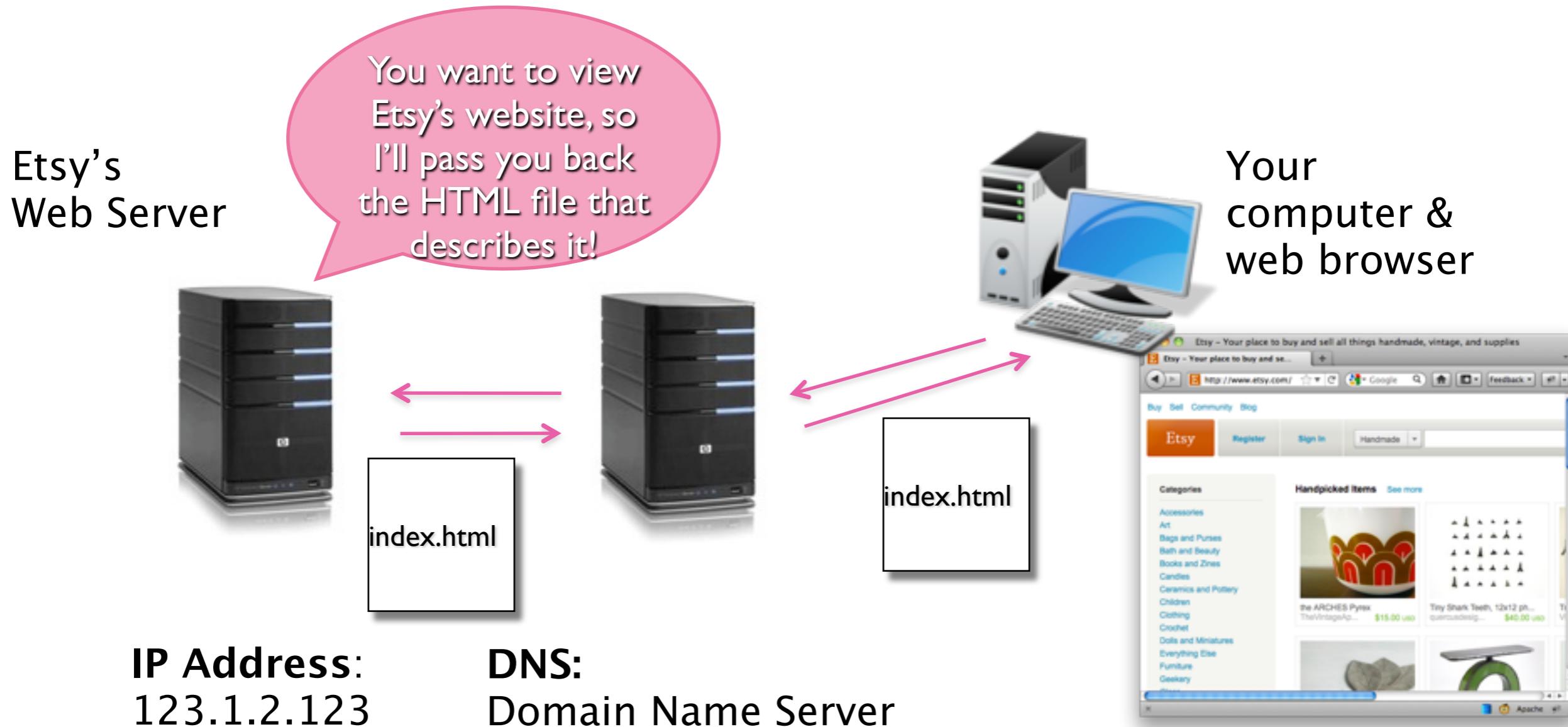
# DNS gives us an Address

What does  
123.1.2.123 mean?  
What does it point  
to?



**DNS:**  
**Domain Name**  
**Server**

# What Happens when you open a site



# Kinds of Web Hosting

- There are several kinds of hosting you can buy:
  - Shared Hosting
  - Dedicated Server
  - Virtual Private Server

# Shared Web Hosting

- Shared Hosting is when multiple sites live on the same web server.
  - Each site has its own separate space on the server.
  - Sites share the server's CPU and memory
  - Cheapest option for hosting (~\$9/month)



Image by MyDocs

# Dedicated Server Hosting

- A dedicated server is a server that is **ONLY** for you
  - You do not share the space with any other customers
  - You get to utilize the full CPU and memory from the computer just for your site
  - Most expensive option (~\$99/month)



Image by MyDocs

# Virtual Private Server

- A Virtual Private Server is a mix of shared hosting and dedicated hosting
  - You still share space on a server
  - But your space is separated into a “virtual server”
  - You get your own dedicated piece of memory and CPU, no more sharing resources
  - Price depends on how much memory you want to use

# Things to Look For in site Hosting

- Don't get charged per email address
- Hosts should have Backup options
- Even better, automatic routine backups for your database and your files
- Hosts may only support specific languages and databases
- Ease of scale if you're building an application
  - Cloud providers like Amazon EC2, Heroku have sliding scales

# Web Hosting Companies

- Companies:
  - Dreamhost
  - asmallorange.com
  - liquidweb.com
  - http://bluehost.com

# FTP: Getting files onto a web server

- How did the index.html file get to the web server?
- Files are copied onto web servers over TCP/IP by using yet another protocol, **FTP: File Transfer Protocol.**
- It is used in applications like Filezilla to upload lots of files in bulk.
- It requires a username, password, and server address.

# Web based FTP

100% Script halted

**net2ftp**  
a web based FTP client

Language: English Skin: Blue

/public\_html/prestashop/modules

Directory Tree: root /public\_html /prestashop /modules

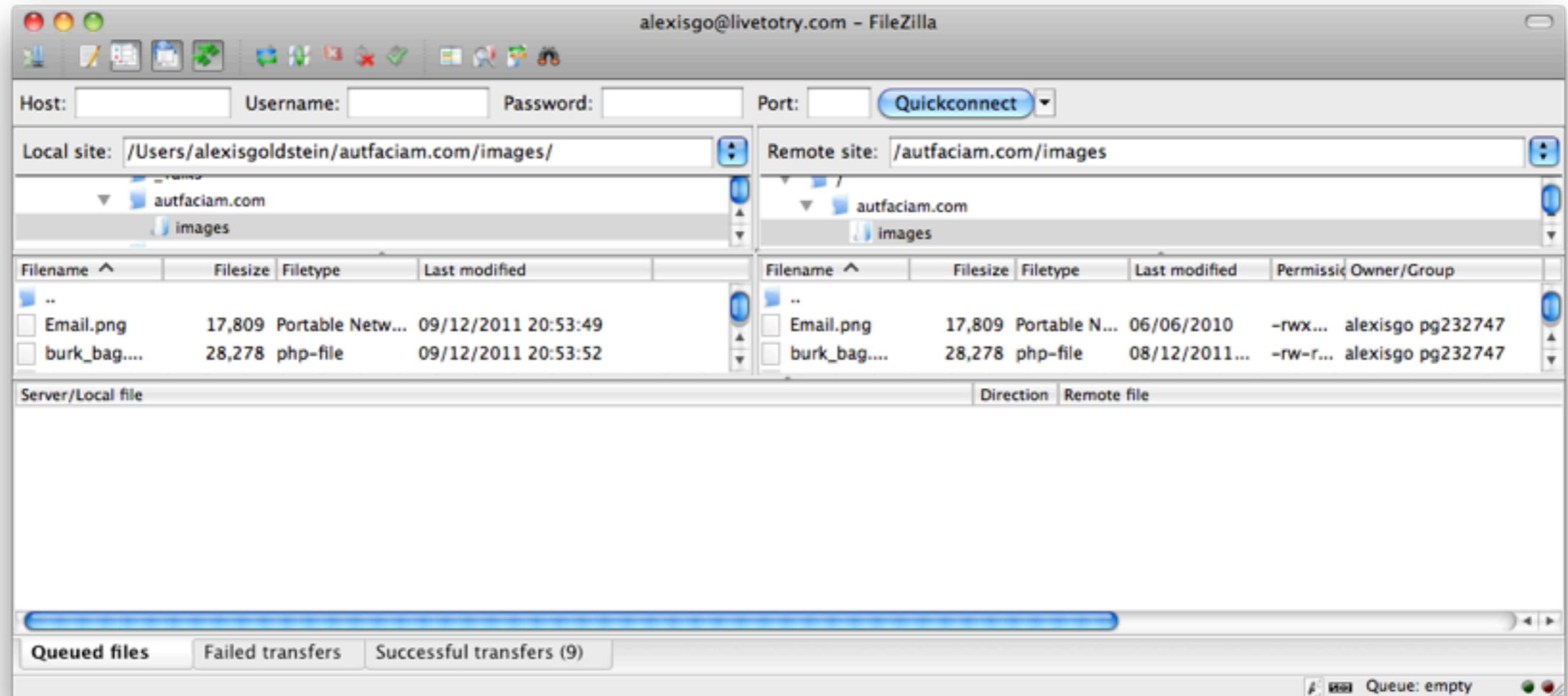
Entries which contain banned keywords can't be managed using net2ftp. This is to avoid Paypal or Ebay scams from being uploaded through net2ftp.

New dir New file Upload Java Upload Flash Upload Install Advanced

Transform selected entries: Copy Move Delete Rename Chmod  
Download Zip Unzip Size Search

All	Name	Type	Size	Owner	Group	Perms	Mod Time	Actions
	Up..							
	themeinstaller	Directory	4096	u172823254	u172823254	nrw-r-x	Jun 11 08:25	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Open</a>
	doc/commitment_screen.gif	GIF file	33052	u172823254	u172823254	n--r--r--	Jun 10 09:36	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Open</a>
	doc/copyright_screen.gif	GIF file	8031	u172823254	u172823254	n--r--r--	Jun 10 09:36	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Open</a>
	doc/hook.gif	GIF file	43764	u172823254	u172823254	n--r--r--	Jun 10 09:36	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Open</a>
	doc/install.txt	Text file	3074	u172823254	u172823254	n--r--r--	Jun 10 09:36	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Open</a>
	doc/slider_screen.gif	GIF file	145183	u172823254	u172823254	n--r--r--	Jun 10 09:36	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Open</a>
	modules/blocklangcurr/blocklangcurr.js	JavaScript file	228	u172823254	u172823254	n--r--r--	Jun 11 04:00	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Open</a>

# Getting files onto a web server



# Site Administration

# Site Admin

- Once you have registered a domain and purchased web hosting, you'll need to learn how to administer your site:
  - Adding and configuring email accounts
  - Managing databases
  - View details of % space you've used
  - Install CMS software like WordPress
- This is typically done from a web-based **Control Panel** that is provided by your hosting company

# Control Panels

- Some companies will have their own custom control panel
- Others will use standard panels:
  - cPanel
  - Webmin
  - Open Panel
- [http://en.wikipedia.org/wiki/  
Web\\_hosting\\_control\\_panel](http://en.wikipedia.org/wiki/Web_hosting_control_panel)

# Admin on Dreamhost

The screenshot shows the DreamHost Admin interface. On the left is a sidebar with a navigation menu:

- DreamHost
- Welcome
- cPanel Importer
- Rewards
- Web Panel API
- Backup Your Account
- Site Of The Month
- Green Hosting
- The DreamHost Shop
- Suggestions
- Donations (To You!)
- Partner Offers
- Give to Charity
- Buy Gift Card
- Domains
- Mail
- Goodies
- E-Commerce
- Users
- Status
- VPS
- Dedicated Servers
- Storage
- Billing
- Support

The main content area has a message: "Now, you can go [install in just one click](#) some weblog software, a forum, a poll, an online store, or even a wiki!"

A tip box titled "Tip #7" says: "Multitude of open-source (read free!) applications to manage your personal and business life are available at [Goodies/One-click Installs](#). Choose between full control (advanced) or hassle-free (easy)."

## Let's Get Started!

Grid of services:

- Website Hosting
- Databases
- Email
- Users
- Domain Registrations
- Free Website Software
- Support
- Billing

# Admin on cPanel



Demo: <http://www.cpanel.net/products/cpanelwhm/try-demo.html>

# 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!