

**If you have not already done so,
please download Aptana:
<http://aptana.com>**

GDI Cincinnati

Intro to HTML/CSS: Class 4

Erin M. Kidwell / @erinmkidwell/ erin@girldevelopit.com

Heather Glenn Rock / @hglennrock / heather@hgr.me

don't be shy. develop it.

Agenda

1. Review of terms, topics, and styling
2. Print stylesheets
3. Layouts
4. HTML5
5. Detecting browser support
6. CSS3

Rounded corners / unevenly rounded corners

Drop shadows / inset shadows

Text shadows

Color: RGBA / HSL / HSLA

Animations: CSS Transforms / Transitions

7. Bonus exercises

Building a menubar

Building a two column layout

Review: Terms

Brief Review of HTML Terms

- Tag
- Elements
- Attributes

Brief Review of CSS Terms

- Element Selector
- Class Selector
- Id Selector
- Pseudoclasses

Quiz

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

- A) ID Selector
- B) Element Selector
- C) Class Selector

Quiz

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

- A) Property
- B) Pseudoclass
- C) Attribute

Review: Topics

CSS:

- Margin and Padding
- Borders
- Float

Positioning:

- Static
- Fixed
- Relative
- Absolute

Quiz

```
<html>
<head>
  <style>
    #centerMe
    {
      ??? : ??? ;
      margin: 0px auto;
    }
  </style>
</head>
<body>
```

```
<div id="centerMe">I should be centered on the page!</div>
```

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

- A) Position
- B) Width
- C) Float

Review: Styling

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.

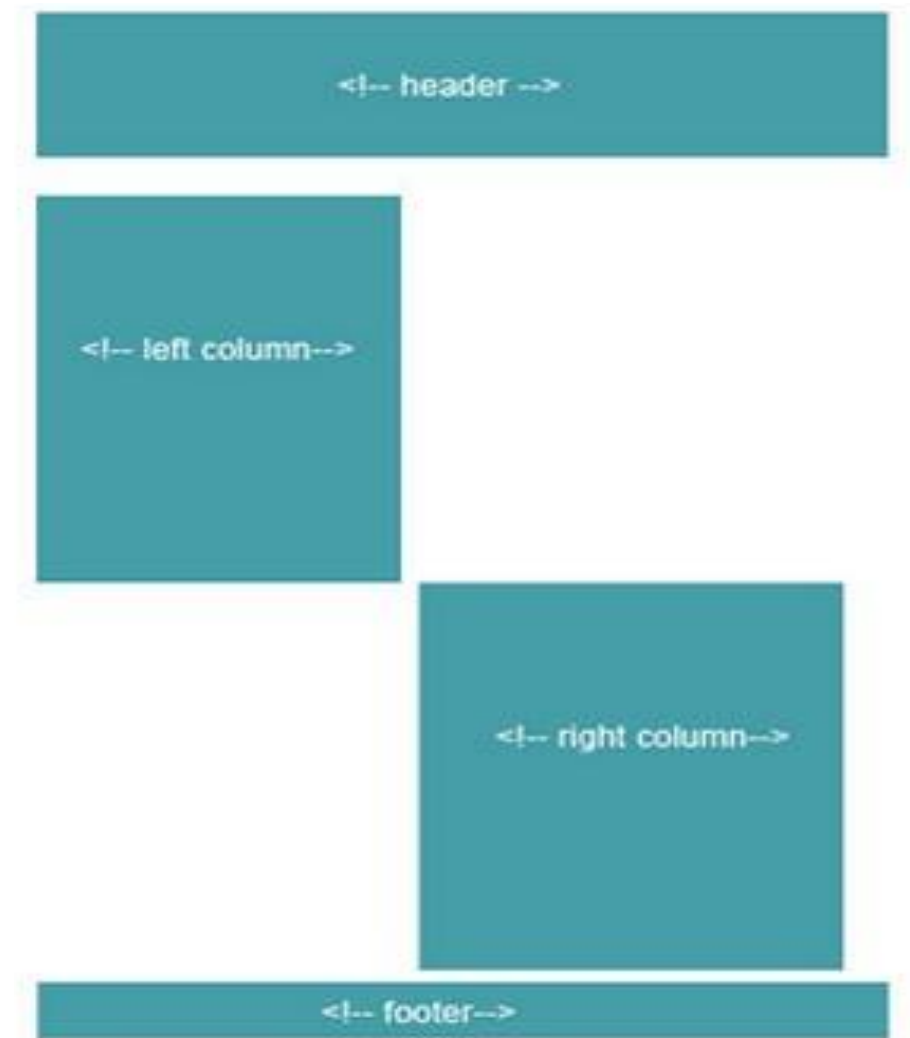
The **float** property specifies whether or not an element should float. It also specifies which direction it should float (left, right). Example:

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


CSS 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.



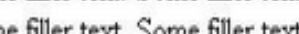
CSS Clear

The clear
property controls
the flow of text
when you're using
float.

Float an image left with no clearance



Floating an image to the left within a div. Notice how the text wraps to the right of the image. Some filler text. Some filler text.
Some filler text. Some filler text. Some filler text. Some filler text.
Some filler text. Some filler text. Some filler text. Some filler text.
Some filler text. Some filler text. Some filler text. Some filler text.

The screenshot displays a web page where a small square image has been floated to the left side of its container. The surrounding text, which appears as generic placeholder text like "Some filler text.", flows naturally from top to bottom, automatically wrapping around the right edge of the image without requiring manual adjustments or additional CSS rules. This visualizes the concept of float-based layout control in older versions of CSS.

Float left and Right with clearance on both



Floating 2 images to left and right within a div. Notice how the text clears the images.
Some filler text. Some filler text. Some filler text. Some filler text. Some filler text. Some
filler text. Some filler text. Some filler text. Some filler text. Some filler text. Some filler text.

Print Stylesheets

If your webpage contains a variety of background colors it can be difficult for visitors to print.

We can create a separate stylesheet just to allow visitors to print by using a new attribute, MEDIA.

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?

We can remove ALL background colors with one simple line in the body selector:

background: white;

```
Body {  
    background: white;  
}
```

Print Stylesheets

We may also want to remove the menu bar, which is currently in the footer.

We can do that by leveraging the **display** property:

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

Good resource for tips and tricks on what to add to your print.css stylesheet:

<http://www.alistapart.com/articles/goingtoprint/>

Liquid vs Fixed Layout

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.

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

Fixed-width Layouts Versus Liquid Layouts:

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

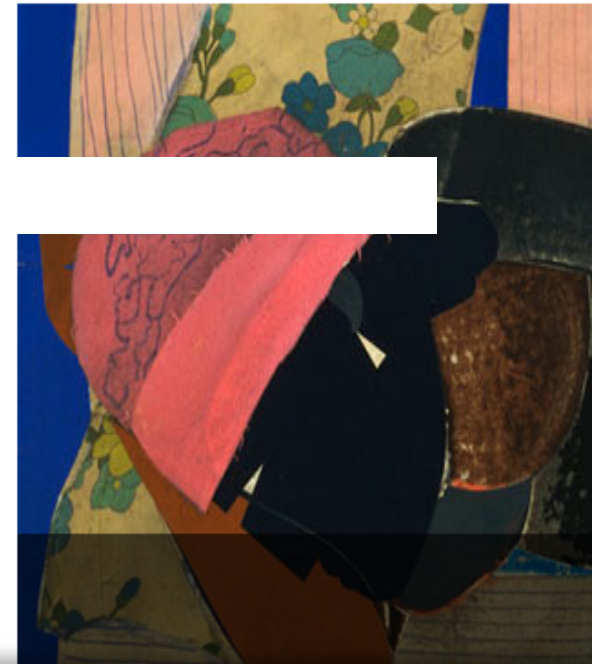
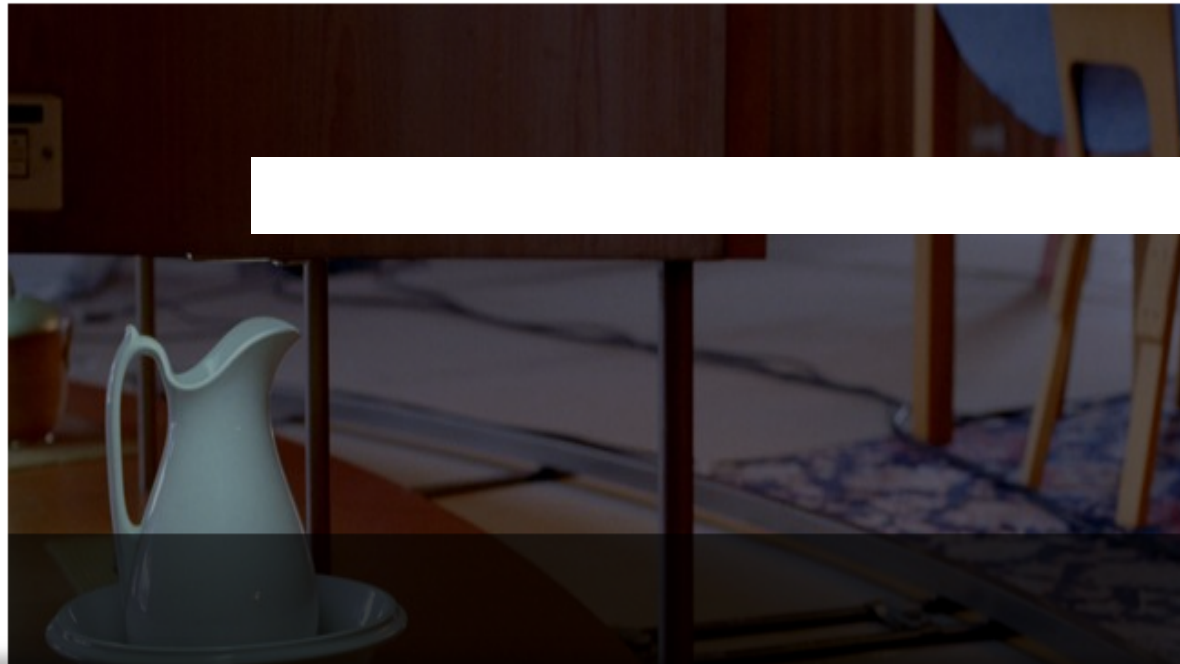
Exercise: fixed layout to liquid

Let's assume we have a three-column layout that uses Absolute Positioning in CSS, which we reviewed last week.

File: <http://livetotry.com/GDI/codeSamples/imitationIsFlattery.html>

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

Finished file: <http://jsfiddle.net/GzwVb/5/>



MoMA PS1

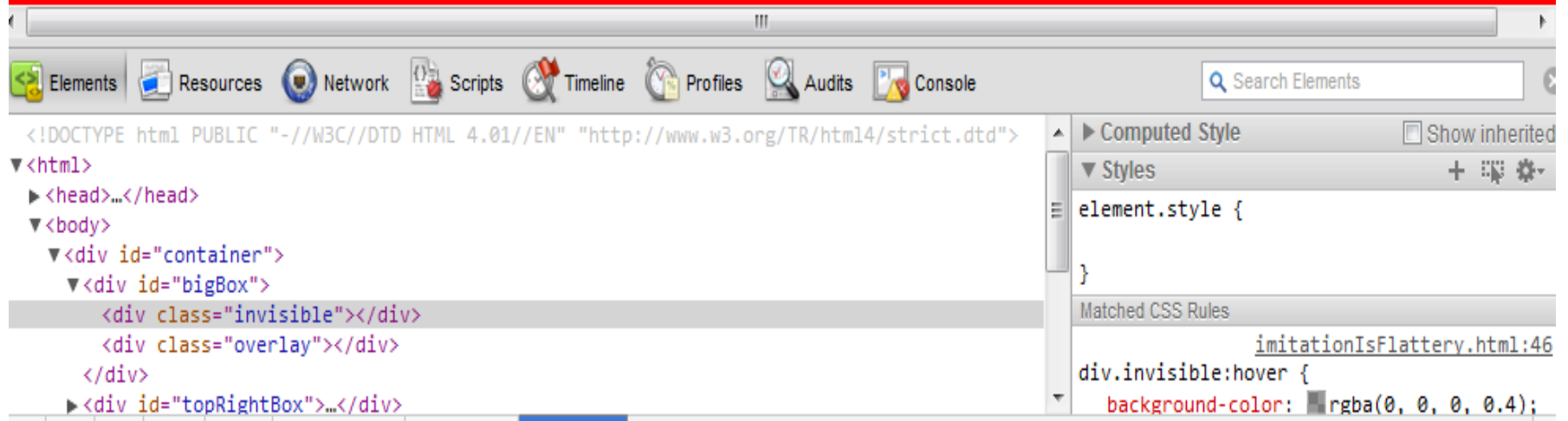
[VISIT](#)

[EXPLORE](#)

[LEARN](#)

[SUPPORT](#)

[SHOP](#)



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



HTML5: Progress and Implementation

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 browser support

Modernizr: 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.

Download Modernizr and then include it in your <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 browser support

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 browser support

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 >
```

Modernizer CSS Example

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

If the browser *doesn't* support CSS columns, as determined by the “`nocsscolumns`” 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.

```
.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;  
}
```

Modernizer

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/>

CSS3 Effects

CSS3 is the latest standard for CSS.

It is backwards compatible, so you do not have to change existing designs.

Browsers will always support CSS2; many of the CSS3 properties have been adopted by modern browsers as well.

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 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.

Browser Prefixes

The CSS3 (and HTML5) specs are still in draft format.

Using the browser prefixes ensures that the functionality will work, even if the w3c changes the standard.

Chrome/Safari: **-webkit** prefix

Firefox: **-moz** prefix

While the names and parameters of the new CSS properties are not likely to change, there is no guarantee that they won't.

In cases where the spec has been mostly finalized you can simply use the property name. Ex: **border-radius**

Exercises: CSS3

Please use: [JSFiddle](#):

Refer to Handout 1 for
Instruction:

Rounded Corners

Unevenly rounded corners

Drop shadows

Inset shadows

Text shadows

Color

RGBA

HSL

HSLA

Transforms

Border-Radius

**Varying
Border
Radius**

Drop Shadow

Inset Shadow

Text Shadow

rgb: old way, no alpha

rgba: new way, with alpha

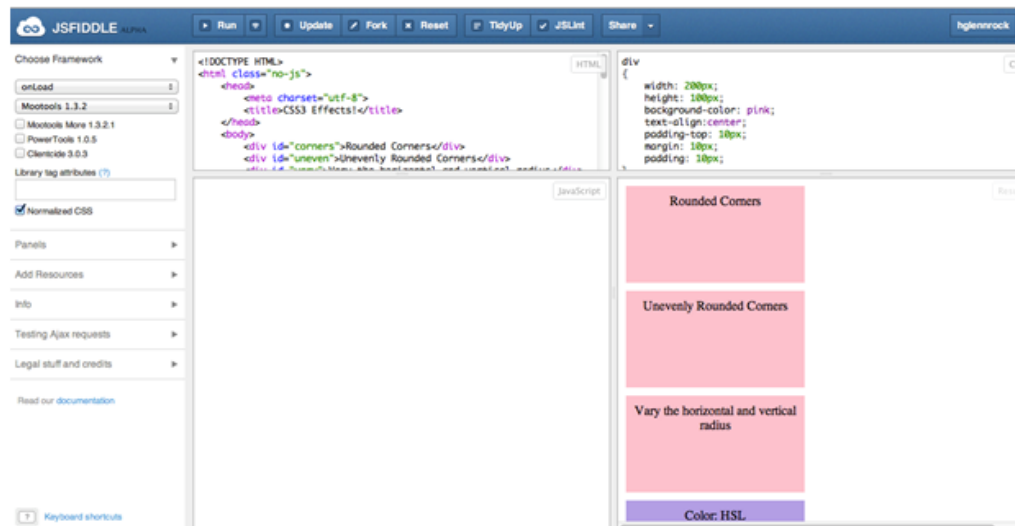
Gradient

Radial Gradient



Intro to HTML/CSS Class 4 Handout: CSS3 with jsfiddle.net

1. Go to <http://jsfiddle.net/7JCWN/1/>



2. Rounded Corners

Find the id selector for #corners in the CSS. Add the following declarations and click "Run":

```
/* firefox */
```

[Link to handout](#)

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

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

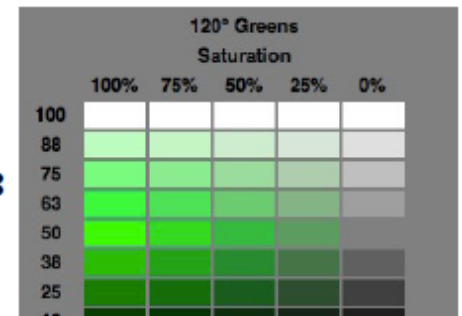
HSLA = Hue, Saturation, Lightness **and Alpha**

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****

Lightness:



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 this!

- Our options: **rotate**, **scale**, **skew** and **translate**.

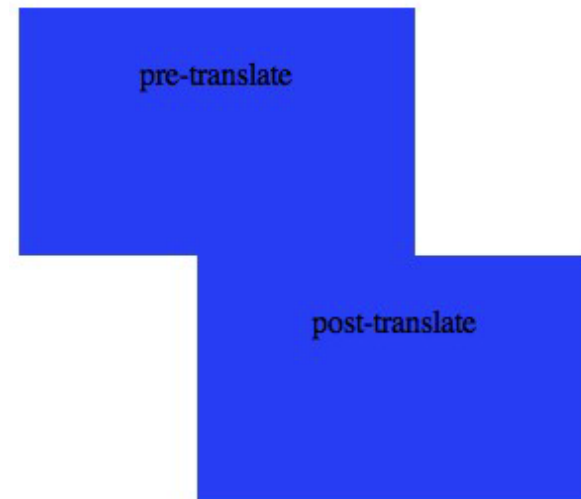
starting file: <http://jsfiddle.net/8etSs/1/>

finished file: <http://jsfiddle.net/fiddlefiddle/8etSs/18/>

CSS Transforms: Translate

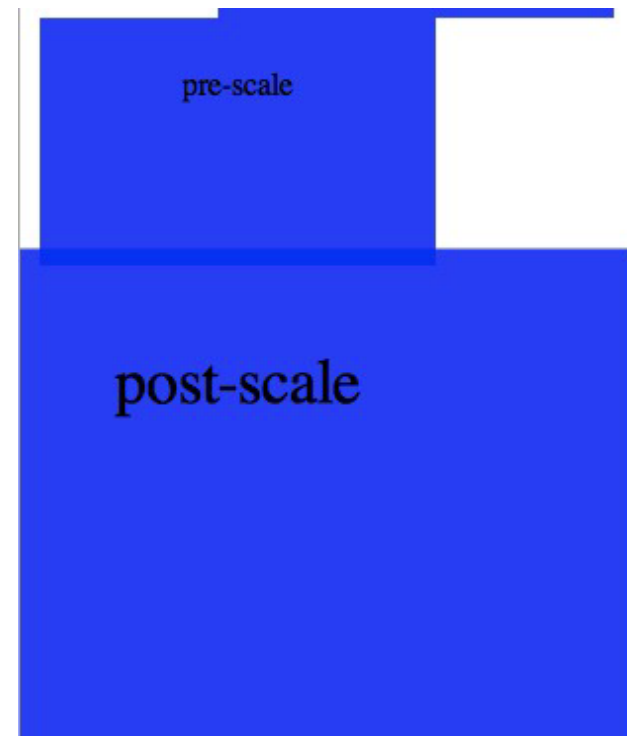
```
-webkit-transform: translateX(90px);  
-moz-transform: translateX(90px);
```

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

CSS Transforms: another example

Starting file: <http://jsfiddle.net/fiddlefiddle/patYu/2/>

Ending file: <http://jsfiddle.net/fiddlefiddle/patYu/4/>



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 **pseudoclasses** 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

```
.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;
}

.paintings img:hover
{
  /*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);
}
```

Normal opacity of the painting is 50%

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

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

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

Current support for CSS3 Transitions:

Chrome

Safari 3.1+ (mobile safari on iPhone if you have iOS 2.0+)

Firefox 4.0

IE 10.0

Opera 10.5x

CSS Transitions

More on CSS3 Transitions:

<http://css3.bradshawenterprises.com/>

<http://samuli.hakoniemi.net/css3-transitions-are-wethere-yet/>

Final version of transform and transitions JSFiddle:

<http://jsfiddle.net/8etSs/>

Further Exercises

Sample 2 and 3 Column Layout

Refer to Handout 2 for instructions

Two: http://bit.ly/two_col

Three: <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)



Intro to HTML/CSS

Class 4 Handout: Two Column Layout w/ CSS + Mobile Web Design

1. Two Column Layout

We are starting out with a two column layout. You can download the code and images here:

<http://ge.tt/6YVcnPL/v/0>

Unzip and open this in [Aptana](#).

Here is the HTML code:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Two Column Layout</title>
    <link rel="stylesheet" type="text/css" href="style.css">
    <meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0,
minimum-scale=1.0, maximum-scale=1.0">
  </head>
  <body>
    <div id="wrapper">

      <div id="header">
        <h1>My Vacation Blog</h1>
        <h2>Places I like to go</h2>
      </div>
```


[Link to Handout](#)

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

<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>

THANK YOU!

Congratulations on completing our Intro to HTML/CSS course!

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 via email at erin@girldevelopit.com