

# The Future

*“In the year 2000...”*  
– **Conan O’Brien**



# Course Updates

Final project due... *today!*

Design write-up due... *today!*

If you didn't know... *surprise!*

Today, we look at demos of HTML5/CSS3.

And relax.

And eat pizza.

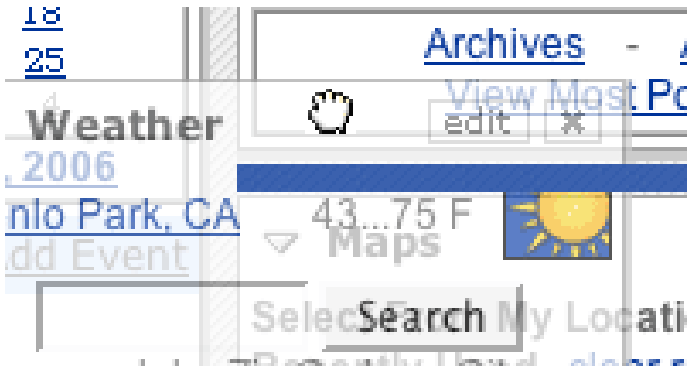
# HTML5

Some cool stuff we'll talk about:

- Advanced drag and drop
- `<CANVAS>`
- `<AUDIO>` and `<VIDEO>`



# Drag and drop!



We can do it using Javascript already, but a big PITA, and we're locked into one window.

How do we drag from one window to another?  
Can we have something that's easy to engineer?

# Drag and drop!

HTML5 provides built-in support for drag and drop now, even allowing for a user to drag information from one window to another.

<http://decafbad.com/2009/07/drag-and-drop/outline.html>

<http://decafbad.com/2009/07/drag-and-drop/api-demos.html>



# <CANVAS>

<CANVAS> provides an easy way to dynamically draw and manipulate graphics on the screen, opening up a world of possibilities that previously required Flash.

```
<canvas id="myCanvas"></canvas>
```

```
<script>
```

```
    var context = document.getElementById('myCanvas').getContext("2d");
```

```
    context.moveTo(30, 30);
```

```
    context.lineTo(100, 130);
```

```
</script>
```



# <CANVAS>

Useful for way more than simple lines and shapes.

<https://bespin.mozilla.com/>

<http://9elements.com/io/projects/html5/canvas/>

<http://www.nihilogic.dk/labs/wolf/>



```
Bespin > SampleProject - index.html
1 <html>
2 <head>
3   <title>Aw man, testing testing!</title>
4   <style type="text/css">
5     #module {
6       background-color: #c7c7c7;
7       margin: 0 auto;
8     }
9   </style>
10 </head>
11 <body>
12   <h1>This is a test.</h1>
13   <div id="module">
14     <h2>Syntax highlighting</h2>
15     <p>Lorem ipsum...</p>
16   </div>
17 </body>
18 </html>
```



# <AUDIO> and <VIDEO>

How do we embed multimedia into a page?

```
<object>  
  <param name="movie" value="ayoutubevideo.swf" />  
  <param name="allowFullScreen" value="true" />  
  <param name="allowsriptaccess" value="always" />  
  <embed src="ayoutubevideo.swf" />  
</object>
```

Reasons for this horrifying code are, um, historical:



vs.



98|30

HTML5



# <AUDIO> and <VIDEO>

HTML5 introduces <AUDIO> and <VIDEO> tags that will allow us to:

- Easily define alternate content.
- Define multiple source files for different codecs and media types.
- Use new, really intuitive API calls like “play()” to easily create our own custom control interfaces.
- Not want to cut ourselves when we look at our code.



# <AUDIO> and <VIDEO>

## Alternative content:

```
<video src="avideo.swf">
```

```
<p>
```

You can even have alternative content here! Like a link to `<a ...>install the Flash player</a>`.

```
</p>
```

```
</video>
```



# <AUDIO> and <VIDEO>

## Multiple source files:

```
<video>  
  <source src="instructions.3gpp" type="video/3gpp" />  
  <source src="video.mp4" type="video/mp4" />  
</video>
```

Browser chooses whichever one it can support.  
No 3gpp? No problem.



# <AUDIO> and <VIDEO>

## Extensions to the DOM API:

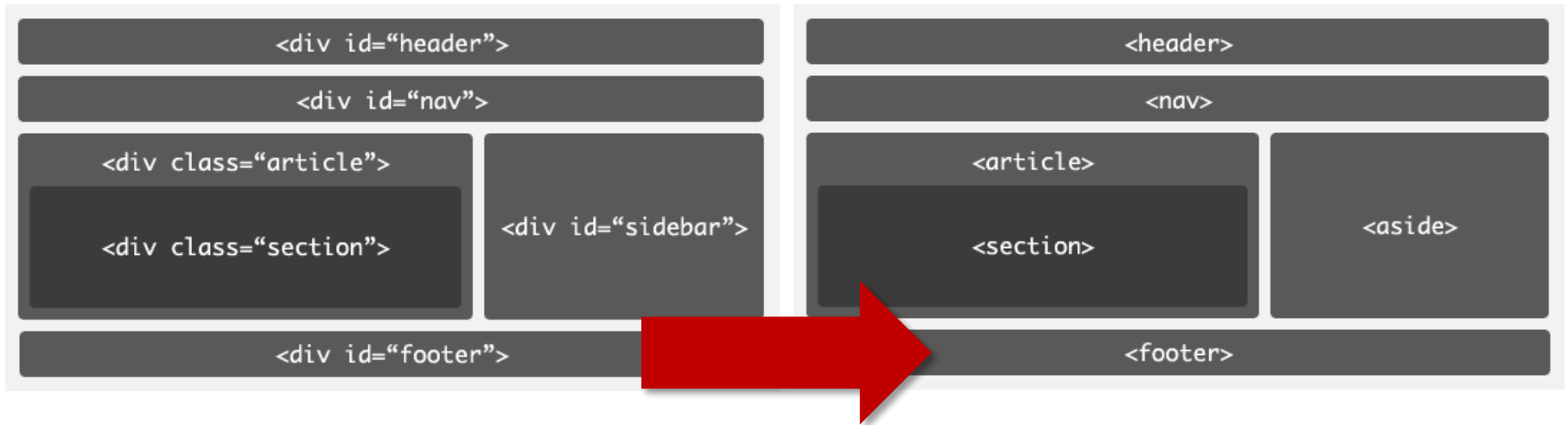
```
<video src="somevideo.wmv" id="video"></video>
```

```
<script> var video = document.getElementById("video"); </script>  
<button type="button" onclick="video.play();">Play</button>  
<button type="button" onclick="video.pause();">Pause</button>  
<button type="button" onclick="video.currentTime += 10;">Skip  
  Forward</button>
```



# Other Stuff

New structural elements...



# Other Stuff

Support for both HTML and XHTML syntax

- HTML5 is being specified based on the DOM tree rather than the syntax of the language (like previous versions were).



# CSS3

Cool stuff includes:

- Rounded corners
- Drop shadows
- Transparency
- Multiple background images
- Multi-column support
- New selectors

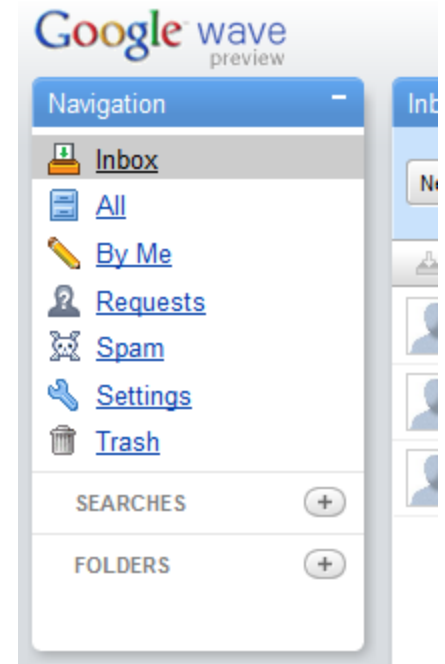


# Rounded Corners

Rounded corners in CSS *sucks*.

Basically need an image for every corner.

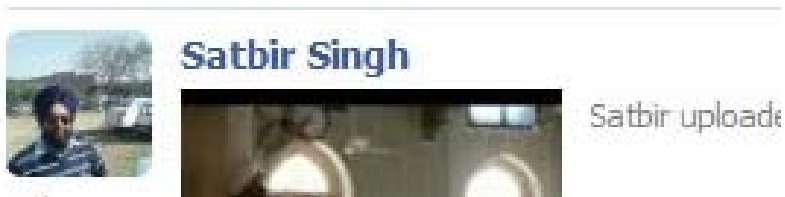
Then, because we can only have one background image per element, need an element for every corner.



# Rounded Corners

Difficulty is one of the many reasons why Facebook dropped all of their rounded corners.

*“...they add an extra layer of complexity to the code.”*  
– **Alexandre Roche**, Facebook Product Designer



# Rounded Corners

But in CSS3:

**border-radius: 3px;**

...currently implemented as:

**-moz-border-radius** (Firefox)

**-webkit-border-radius** (Safari/Chrome)

<http://tonypoor.com/98|30/examples/corners.html>



# Drop Shadows

Shadows on text are done via **text-shadow**.

Shadows on an entire element via **box-shadow**.

**text-shadow:** 2px 2px 1px #cccccc;

[offset-down] [offset-right] [blur] [color]

box-shadow is currently implemented as -moz-box-shadow and -webkit-box-shadow.

<http://tonypoor.com/98|30/examples/shadow.html>



# Transparency

New color unit: **rgba()**.

Red/green/blue/alpha, where alpha ranges from 0 (fully transparent) to 1 (fully opaque).

**background-color: rgba(25, 25, 25, .5);**  
= half-translucent grey.

Also the **opacity** property.



# Multiple Background Images

background:

```
url(bg1.gif) top left no-repeat,  
url(bg2.gif) top right no-repeat,  
url(bg3.gif) bottom left no-repeat;
```

**This is a big deal!**



# Other Stuff

Other cool CSS3 stuff includes **new selectors** ("select the third LI in every UL") and support for **multiple columns** (and perhaps even grids).

But CSS is still a long way off, and we have to wait for browsers to catch up and implement the specifications.

For a while, we'll have just only limited implementation in select browsers... or can we fix that?



# The Role of Javascript

Browsers are making their Javascript faster. A lot faster. Screenreaders are becoming better at reading JS, etc.

*Meanwhile, in W3C land...*

Contributors are arguing with each other, having to create reference implementations for their specs, and overall making slow progress.

<http://www.ishtml5readyyet.com/>\*

\* this is a little misleading. don't panic. :)



# The Role of Javascript

We see that browser vendors are taking matters into their own hands.

*-moz-border-radius, -webkit-box-shadow...*

...can we do this too?

```
#module {  
    -98|30-transform: rotate(-15);  
}
```



# eCSStender

With Javascript, we can.

[eCSStender](#) is a *really young* Javascript library that:

1. Scans through your CSS looking for “custom” properties that you tell it to look for.
2. Takes those properties and uses Javascript to render them accordingly in a cross-browser way



# eCSStender

Eric Meyer: “**Javascript will save us all.**”

Well... maybe.  
Maybe not.  
Who knows?



# Epilogue

Perspectives I hope you gained:

## **Front-end engineering:**

- The importance of semantics and accessibility.
- Separation into structure/appearance/behavior.
- A decent grasp on HTML/CSS concepts.

## **Design:**

- Basic visual design principles.
- The ‘overall picture’ of user experience design.
- An introduction to web usability.



# Epilogue

# THE END

...and if you ever have an incurable CSS problem,  
shoot me an email or IM. 😊

**Thanks!**

