

COEN 168/268

# Mobile Web Application Development

## **HTML 5**

Peter Bergström (pbergstrom@scu.edu)

Santa Clara University

# HTML



# A brief history lesson...

- Vannevar Bush published "As We May Think" in 1945
- Described the concept of the web
- However, envisioned with the limited analog technology at the time using microfilm, etc
- Not really feasible

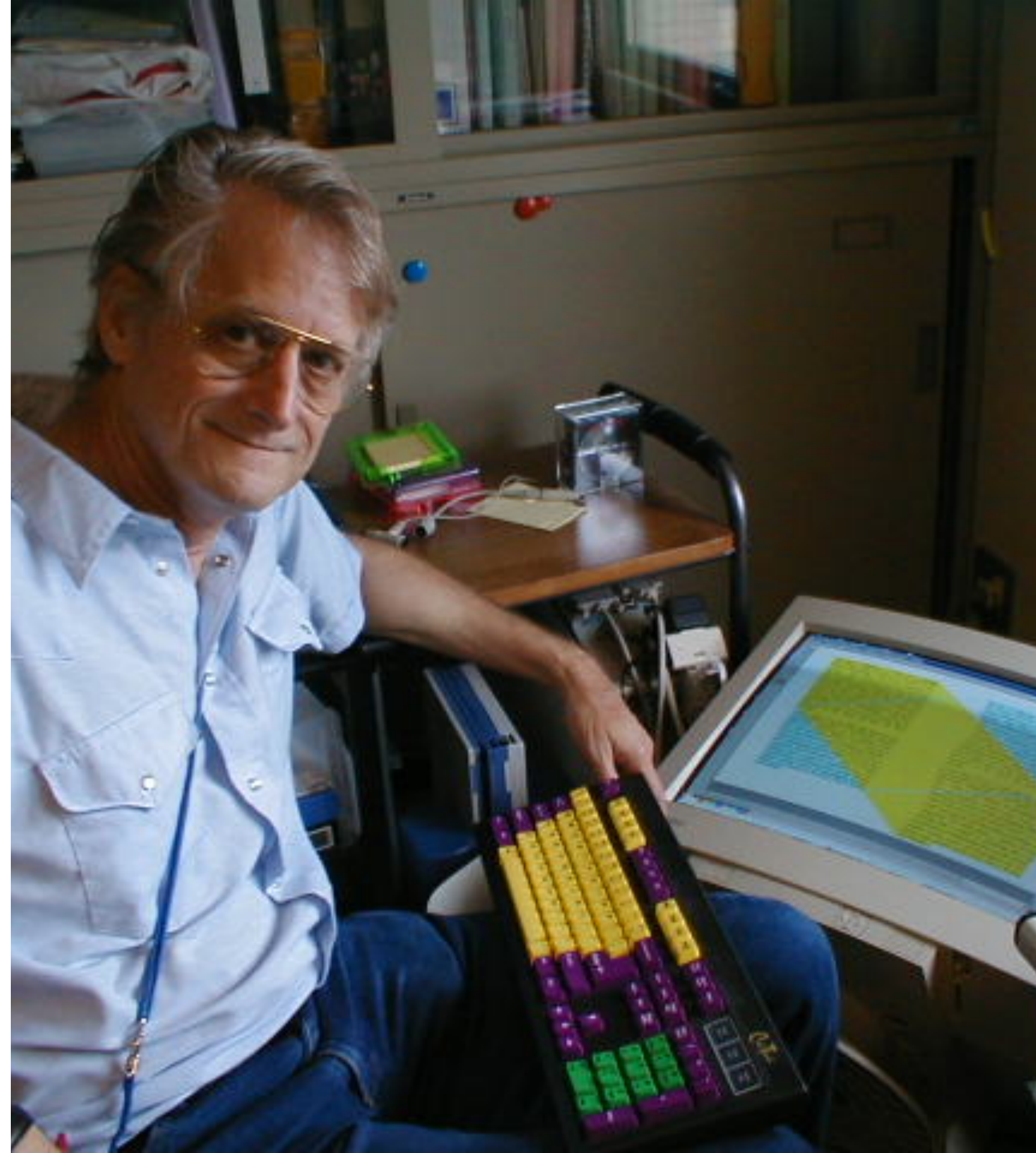


... wholly new forms of  
encyclopedias will appear, ready  
made with a mesh of associative  
trails running through them, ready  
to be dropped into the memex and  
there amplified...

— *Vannevar Bush*

# Then the term "hypertext" came to be

- Hypertext predates HTML itself by almost 20 years
  - The term was coined by Ted Nelson in 1963
  - Basically, it is a text document that contains links to other documents
- Many hypertext concepts predated HTML and the web, but never took off
- However, they showed the future to come



# The Mother of All Demos

Doug Engelbart demoing NLS in 1968

STATEMENT OF ...  
...  
...  
...  
...





# The Mother of All Demos

- NLS in 1968, before we went to the moon, had many of the elements of modern computing
- Showed things that we now take for granted, but was unheard of then
  - Such as the mouse, video conferencing, word processing, hyperlinks, etc







# HTML and the Web

- Invented by Tim Berners-Lee while at CERN in 1990
- HTML was influenced by SGML, a 1980s markup languages used for text documents
- However, HTML added links to connect documents together
- And so, the web was born and here we are, 24 years later

# The first web server



What is HTML?

**HyperText Markup Language**

# What is HTML?

- It is a **MARKUP** language, not a programming language
- Describes the content and structure
- Does not apply any styling to it
- Focuses on **semantics** to add context to what you are describing



# HTML Tag Syntax

`<tagname>`Your content goes here.`</tagname>`

- The **tagname** is a keyword surrounded by angle brackets < >
- Come in sets. Opening tag and closing tag
- Opening tag: <tagname>
- Closing tag: </tagname> <- note the /
- The element content is everything between the tags
- Should all be lowercase

For example, the Paragraph `<p>` tag

`<p>`This is a paragraph.`</p>`

`<p>`This is a second paragraph.`</p>`

~

This is a paragraph

This is a second paragraph

# HTML is made to be nested

```
<p>This is a paragraph.</p>
```

```
<p>This is a <strong>second</strong> paragraph.</p>
```

~

This is a paragraph

This is a **second** paragraph

# Block Elements

- Most HTML elements are **block** level
- This means that they break the flow of the content
- Define flow of the document
- Some examples are: `<h1>`, `<p>`, `<ul>`, `<table>`, `<div>`



# Unordered lists

```
<ul>  
  <li>Item 1</li>  
  <li>Item 2</li>  
</ul>
```

- Item 1
- Item 2

# Ordered lists

```
<ol>  
  <li>Item 1</li>  
  <li>Item 2</li>  
</ol>
```

1. Item 1

2. Item 2

# Unordered and ordered lists can be nested

```
<ul>  
  <li>  
    Item 1  
    <ol>  
      <li>Sub item 1</li>  
      <li>Sub item 2</li>  
    </ol>  
  </li>  
  <li>Item 2</li>  
</ul>
```

# Unordered and ordered lists can be nested

- Item 1
  - 1. Sub item 1
  - 2. Sub item 2
- Item 2



# Inline Elements

- **Inline** elements do not break the flow of content
- No newline is created
- Most often used for styling
- Some examples are `<b>`,

This is a `<b>bold</b>` statement.

# <div> and <span> elements

- <div> elements are **block** level containers that groups other elements
- <span> elements are **inline** level containers that groups text

A span of text is inline.

A span of text **is inline**.

# There are a lot of tags...

- Headings are described with the <h1> to <h6> tags
- *Italics* text is described with the <i> tags
- **Bold** text is described with the <b> or <strong> tags
- And so, so many more...

# HTML Element **Attributes**

- Attributes provide additional information of the elements
- Attributes are specified in the start tag
- Attributes are key="value" pairs:

This is a `<span style="color: red;"></span>`.



# Some common attributes

- `id`: For adding a unique id for an element
- `class`: For adding one or more classnames for an element for CSS targeting
- `style`: For inline CSS style for an element
- `title`: Extra title as a "tool tip"

# Hyperlinks using the <a> tag

```
<a href="{url}" title="{text}">Link Text</a>
```

- **href** attribute defines the URL that the link takes you to:

```
<a href="http://coen268.peterbergstrom.com">COEN 168/268!</a>
```

- Also, <a> serves as anchors inside a document:

```
<a href="#hyperlink-section">Learn more about hyperlinks.</a>
```

...

```
<a id="hyperlink-section">Hyperlinks FTW!</a>
```

# <img> tag



```

```

- <img> tag doesn't need a closing tag. It is self closing using />
- src attribute points to the URL of the image
- alt attribute specifies a alternative text description of the image

# A HTML document contains nested elements

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>
      My Document's Title
    </title>
  </head>

  <body>
    My content goes here.
  </body>
</html>
```

# The Document Type

`<!DOCTYPE html>`

- The document type describes the type of document.
- It could be HTML4, XHTML, HTML5
- This is the document type, for HTML5, it is just 'HTML'
  - We will use this one only in this class

# <html>, <head>, and <body>

- The <html> tag wraps the whole document.
- The <head> tag is where you define top level info
- The <body> tag where your content belongs

# <head> element components

- Document title goes inside the <title> element, obviously (Semantics FTW)
- Let's talk about the others a bit more...

# Defining CSS in the <head>

- Inline CSS Stylesheet:

```
<style>
  .fire {
    color: red;
  }
</style>
```

- External CSS Stylesheet:

```
<link rel="stylesheet" href="styles.css"/>
```



# Defining JavaScript in the <head>

- Inline JavaScript:

```
<script>
window.onload = function() {
    // Things that Batman does.
    alert('POW! BAM! BOOM! WHAM!')
};
</script>
```

- External JavaScript:

```
<script src="app.js"></script>
```

# Defining <meta> tags in the <head>

- Provides additional metadata about the document.

Character encoding:

```
<meta charset="utf-8">
```

Viewport sizing:

```
<meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1">
```

Search keywords:

```
<meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript">
```

Periodical page refresh:

```
<meta http-equiv="refresh" content="30">
```

# Use **UTF-8** Character Encoding!

- **UTF-8** should be defined in your documents!

```
<meta charset="utf-8">
```

- It allows you use unicode characters natively without HTML entity encoding
- Instead of hundreds of characters, you can use more than 10000

# HTML entities

- **UTF-8** is good for most characters and will let you avoid this
- There are reserved chars as they are used in HTML itself:
  - less than: `<` -- `&lt;`;
  - greater than: `>` -- `&gt;`;
  - ampersand: `&` -- `&amp;`;
  - non-breaking space: `&nbsp;`;

# So, that's the basics of HTML

*Sorry if that was a bit boring*

I'd recommend **w3schools.com** for more info

# HTML



HTML 5 is where the fun starts

Makes client-side web apps possible...

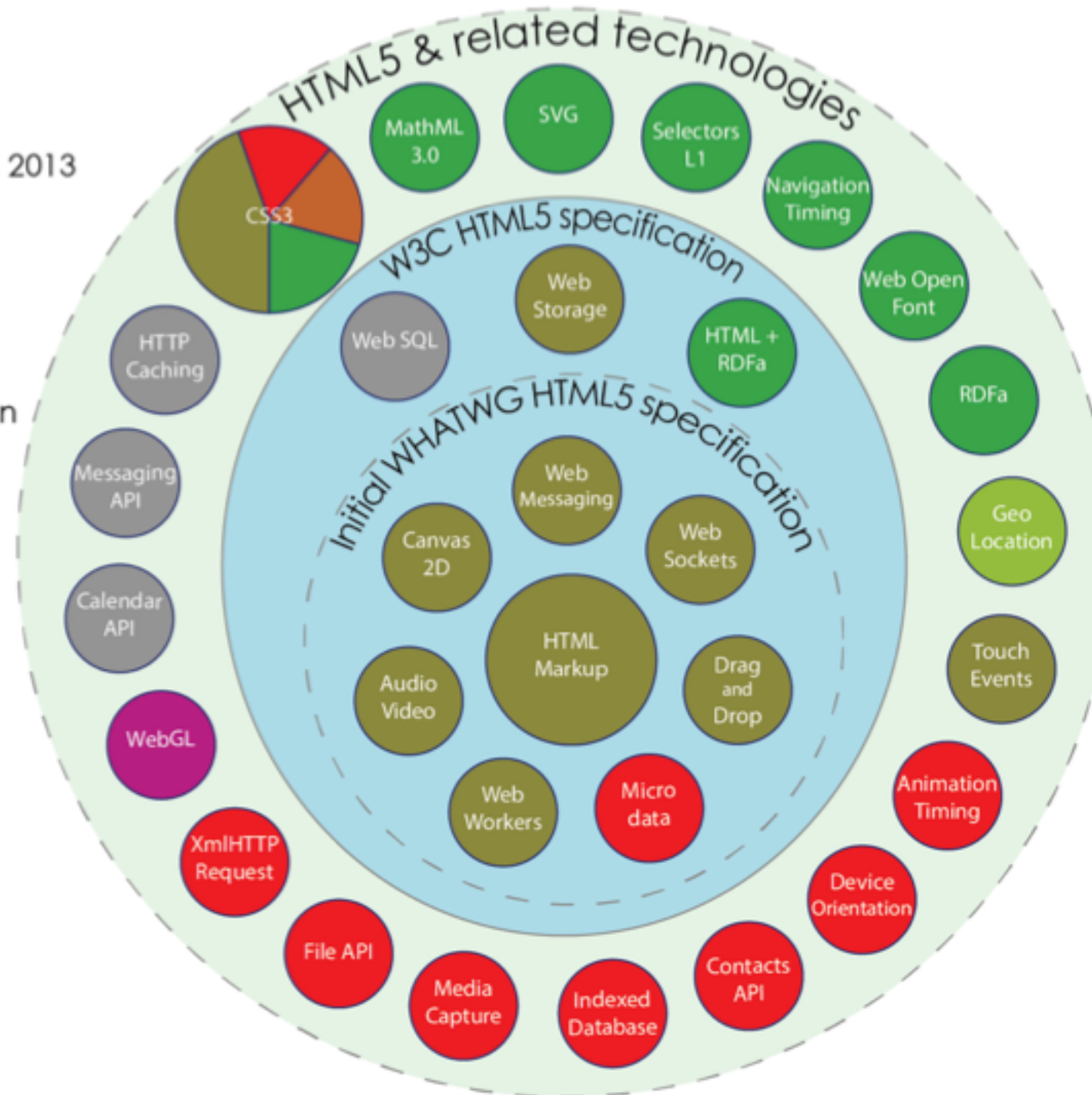
# HTML 5 introduces a lot of new things

- Semantic elements like `<header>`, `<nav>`, `<footer>`, `<section>`, ...
- CSS3
- SVG and `<canvas>` tag for 2D and 3D graphics
- `<video>` and `<audio>` elements with playback
- APIs for local storage and SQL, geolocation, drag & drop, ...
- Form elements for tel, date, time, calendar, email, search, ...



# HTML5

Taxonomy & Status on January 20, 2013



by Sergey Mavrody (CC) BY · SA

HTML5 related APIs. [12]

# HTML 5 has a focus on Semantics

This means that elements actually describe the content that it contains

# Before HTML 5

```
<div class="post">  
  <p class="date">June 25, 2014</p>  
  <h2>  
    HTML 5  
  </h2>  
  ...  
</div>
```

# With HTML 5

```
<article>
  <header>
    <time datetime="2014-06-25" pubdate>June 25, 2014</time>
    <h2>
      HTML 5
    </h2>
  </header>
  ...
</article>
```

# Semantics Help Accessibility Too

- Often you forget making your content accessible to people with disabilities
- With added semantics, screen readers will work MUCH better

Let's try this out

# CSS3

- More powerful selectors
- Backgrounds, borders, shadows
- 2d and 3d transformations
- Animations

We will go into CSS in depth later...

# **SVG** - Scalable **V**ector **G**raphics

- Vector-based graphics
- Can be scaled without degrading
- Defined in XML
- Competes with Flash
- Open standard

Demo: [http://papercube.peterbergstrom.com/#Papers/Per\\_Year/0/16166/2](http://papercube.peterbergstrom.com/#Papers/Per_Year/0/16166/2)



## <canvas> tag

- Raster-based graphics
- Cannot be scaled without degrading
- Written in JavaScript.
- Can create a lot of complex, interactive effects
- Much more common than SVG

Demo: <http://davidwalsh.name/canvas-demos>

Demo: <https://developer.mozilla.org/en-US/demos/tag/tech:canvas>

<canvas> tag code to draw a green rectangle

```
<canvas id="mycanvas" width="200" height="200"></canvas>
```

```
var mycanvas = document.getElementById( 'mycanvas' );  
var ctx      = mycanvas.getContext( '2d' );
```

```
ctx.fillStyle = 'green';  
ctx.fillRect(30, 30, 50, 50);
```

We will go into the <canvas> tag in the context of gaming and advanced graphics later...

# HTML 5 <video> and <audio> elements

- Video on the web became possible with Flash (think Youtube)
- However, Youtube is now moving over to HTML 5 video
  - <http://www.youtube.com/html5>
- The new elements are a lot easier to develop for and use open standards
- Works on mobile

# Before the HTML 5 <video> tag...

- You would have to worry about plug-ins
- Complicated HTML using nested <embed> and <object> tags
- Didn't work on mobile phones at all



```
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000" width="550" height="400" id="movie_name" align="middle">
  <param name="movie" value="batman.swf"/>
  <!--[if !IE]>-->
  <object type="application/x-shockwave-flash" data="batman.swf" width="550" height="400">
    <param name="movie" value="batman.swf"/>
  <!--<![endif]>-->
    <a href="http://www.adobe.com/go/getflash">
      
    </a>
  <!--[if !IE]>-->
</object>
<!--<![endif]>-->
</object>
```

# With the HTML 5 <video> tag

```
<video width="800" height="600" controls>  
  <source src="batman.mp4" type="video/mp4">  
  <source src="batman.ogv" type="video/ogg">
```

This is just sad, upgrade your browser so you can use the video tag.

```
</video>
```

— — — ) ( ° - ° )



# HTML 5 APIs

- Local storage and SQL for persistent storage in the browser
- Web workers for JavaScript multi-threading
- Drag and drop
- Geolocation for, well, geolocation

We will go into these in depth when we get further into the course

# HTML 5 Input Types

- Lots of new form input types that add semantic context
- Specialized native browser controls
- **type** attribute used to define these new types:
  - color, date, datetime, datetime-local, email, month, number, range, search, tel, time, url, week

```
<input name="phone" type="tel" placeholder="(408) 555-1234">
```

# HTML 5 Form Demos

- <http://html5doctor.com/demos/forms/forms-example.html>
- <http://demothenes.info/blog/588/The-HTML5-Color-Input>

# Further information about HTML 5

- **Dive Into HTML:** <http://diveintohtml5.info>
- **W3 Schools:** <http://www.w3schools.com>
- **World Wide Web Consortium:** <http://www.w3.org>

# HTML



Now, let's start our running example

## The Calculator

COEN 168/268

# Mobile Web Application Development

## **HTML 5**

Peter Bergström (pbergstrom@scu.edu)

Santa Clara University