iOS Web Development

iOS App Development
Fall 2010 — Lecture 28

Questions?

Announcements

- Reminder — final project dates of interest...
  - Monday, December 13th
    - Project source code must be submitted by 11:59 pm
  - Tuesday, December 14th
    - Special office hours (5:30 – 6:30 pm) if you wish to test your project on the demo hardware
    - Mock App Store page must be submitted by 11:59 pm
  - Wednesday, December 15th
    - Presentation slides must be submitted by 11:59 pm
  - Thursday, December 16th
    - Presentations and demos 6:00 – 8:00 pm

Today's Topics

- App Development Options
- Web Technologies Crash Course
- Detecting Mobile Safari
- Viewports
- Hiding the Location Bar
- Enabling Web Clips
- Web App Capability
- Geolocation
- Offline Web Apps
- Client-Side Storage
- Debugging
- Events
- HTML5 Elements
- Stylesheet device selection
- Performance
- Frameworks
- Transitions
- Dashcode

App Development Options

App Development Options

- There are several options for developing apps for iOS...
  - Native Apps
  - Hybrid Apps
  - Web Apps

Native Apps

- Language: Objective-C (Cocoa Touch)
- Cost: $99 per year
- Approval Process: initial app submission & updates
- App Store Listing: yes
- Paid Apps: App Store — uses integrated charging/billing — Apple takes 30% cut
- APIs: all native APIs available (accelerometer, vibration, camera, sound, GPS, network, etc.)

Web Apps

- Language: HTML, CSS & JavaScript
- Cost: none (aside from hosting)
- Approval Process: none
- App Store Listing: no
- Paid Apps: you're on your own for processing & authorization to content (if not free)
- APIs: much more limited — basic orientation & location (iOS 4.2.x added some finer motion/orientation controls)

Hybrid Apps

- Language: some Objective-C, though mostly HTML, CSS & JavaScript
- Cost: $99 per year
- Approval Process: Initial app submission
- App Store Listing: yes
- Paid Apps: go it yourself or leverage App Store
- APIs: some native APIs available if using a bridging framework such as PhoneGap (http://phonegap.com)

Web Technologies

Crash Course

HTML

- HyperText Markup Language (HTML) is the fundamental mark-up language used to create web content
- Think of HTML as the content or structure behind your documents (e.g. I'm a heading)
- Mobile Safari has excellent support for HTML & XHTML including many new features in the emerging HTML5 spec

Sample HTML

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>Sample Document</title>
    <link rel="stylesheet" href="style.css" />
    <script src="../common/js/jquery-1.4.4.min.js"></script>
    <script src="script.js"></script>
  </head>
  <body>
    <h1>Sample Document</h1>
    <div id="container">
    </div>
  </body>
</html>
```

CSS

- Cascading Style Sheets (CSS) facilitates the separation of the presentation details from the HTML content
- Think of CSS as the presentation instructions for your documents (e.g. make me bold & green)
- Mobile Safari also has very good CSS support including some of the latest features in emerging specs

Sample CSS

```css
body {
  text-align: center;
  font-family: Helvetica, sans-serif;
}

h1 {
  color: #33c;
}

#container {
  border: 5px dotted #900;
  margin: 0 10em;
  padding: 1em;
}

.highlight {
  background-color: #ffa;
}
```

JavaScript

- JavaScript is a small, lightweight, object-oriented, cross-platform scripting language
- Primarily used in the browser to provide interaction
- Mobile Safari has a pretty good JavaScript engine that supports some of the newer JavaScript APIs typically associated with HTML5

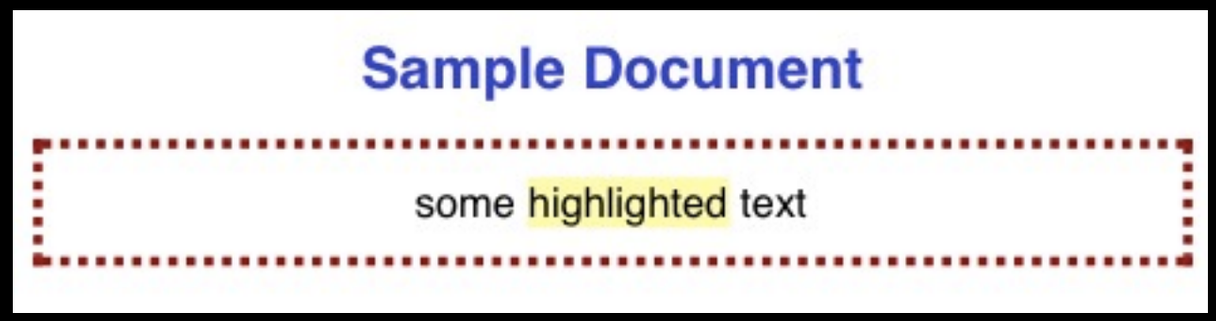
Sample JavaScript

```
// run this when the page is ready
$(function() {

  // add this content to the element id'd container
  $('#container').html('some <span class="highlight">highlighted</span> text');

});
```

Web Server

- The role of the web server is to serve up various resources over HTTP
- These resources may be static
  - e.g. HTML, CSS, JavaScript, images, etc.
- Or, they may be dynamic and created by a server-side language
  - e.g. PHP, Ruby, Python, JavaEE, .NET, etc.

Output When Viewed in iPhone



Detecting Mobile Safari

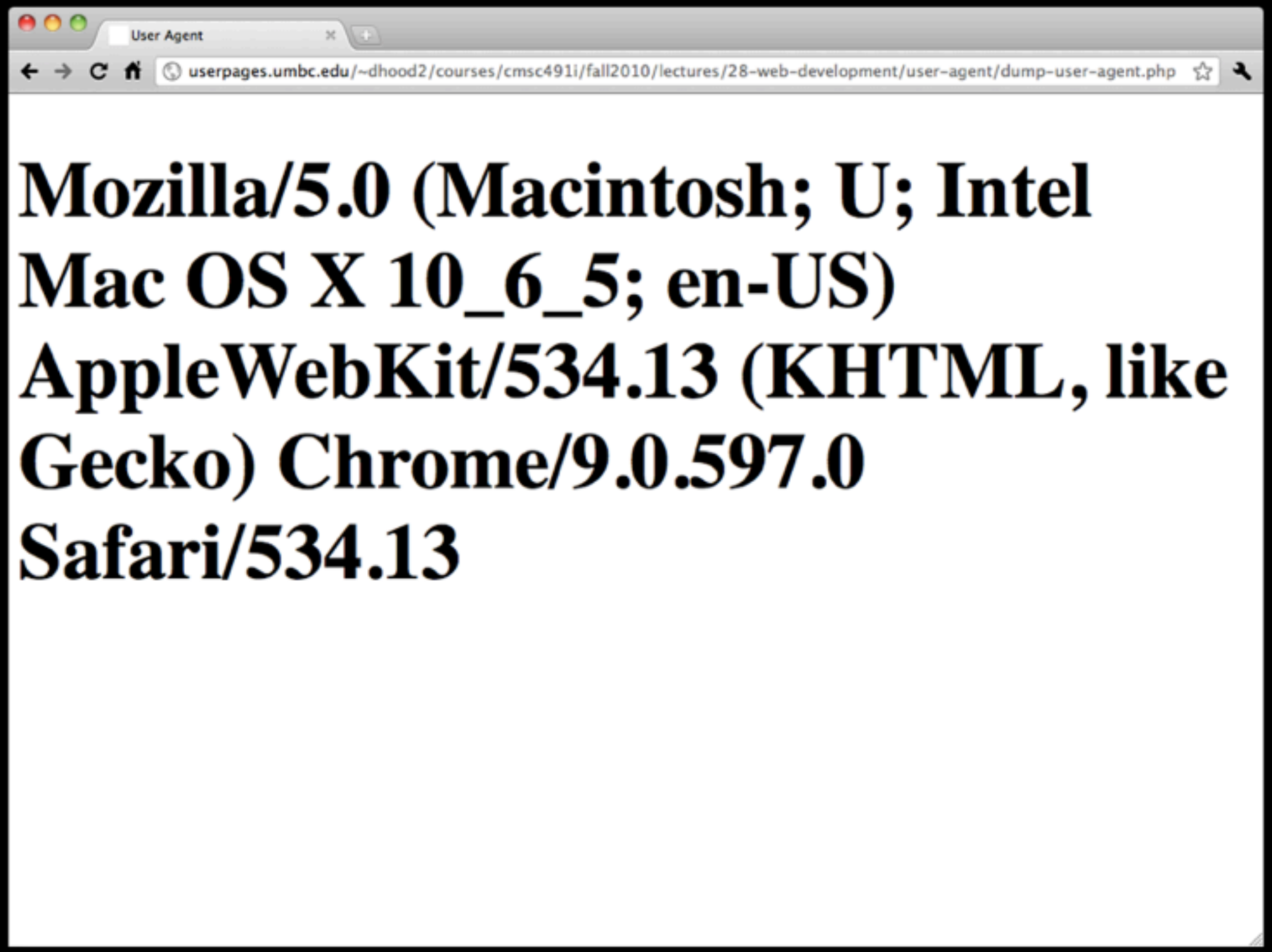
Detecting Mobile Safari

- The easiest way to detect Mobile Safari is to examine the user agent string
- What's a user agent string?
  - The user agent string is something that a web browser passes up to the server as part of the HTTP request to identify itself

Dumping the User Agent (PHP)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>User Agent</title>
    <style>
       h1 { font-size: 400%; }
    </style>
  </head>
  <body>
    <h1><?= $_ENV['HTTP_USER_AGENT'] ?></h1>
  </body>
</html>
```

User Agent — Chrome (Desktop)



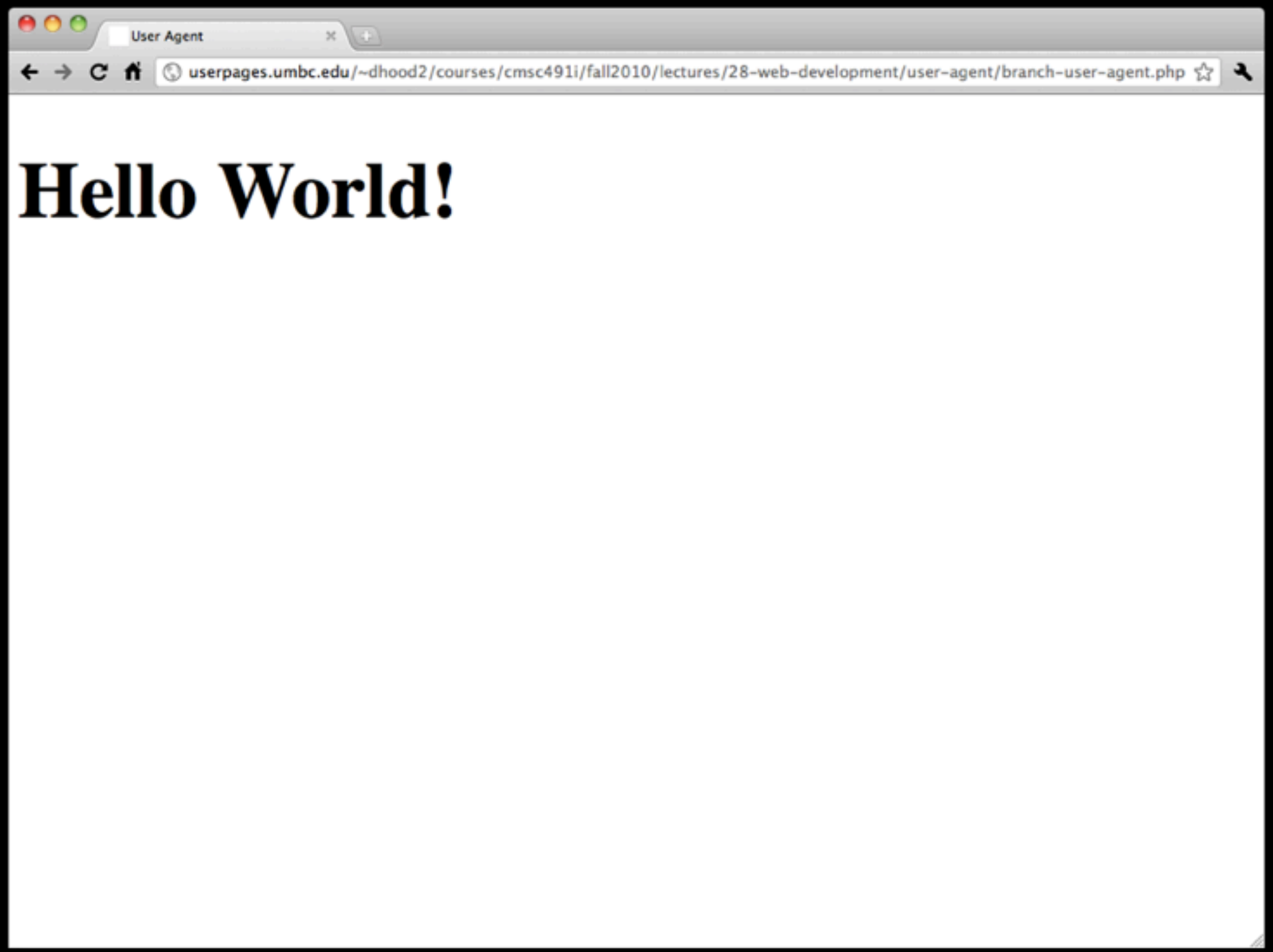
User Agent — iPhone Simulator

User Agent — iPhone 4 (Actual Device)

Branching for Mobile Safari (PHP)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>User Agent</title>
    <style>
       h1 { font-size: 400%; }
    </style>
  </head>
  <body>
<? if ((strpos($_ENV['HTTP_USER_AGENT'], 'iPhone') !== FALSE) ||
       (strpos($_ENV['HTTP_USER_AGENT'], 'iPod') !== FALSE) ) { ?>
    <h1>Hello iPhone or iPod touch!</h1>
<? } else if (strpos($_ENV['HTTP_USER_AGENT'], 'iPad') !== FALSE) { ?>
    <h1>Hello iPad!</h1>
<? } else { ?>
    <h1>Hello World!</h1>
<? } ?>
  </body>
</html>
```

What Non-iOS Device See



What an iPhone/iPod touch Sees

What an iPad Sees

Viewports

Viewport

- You may have notices that the “as is” view is pretty zoomed out and hard to read
- Mobile Safari defaults to a viewport of 980 px
  - Meaning, it will scale down a 980 px wide section to fill the width of the screen which is 320 px
- On the desktop, the viewport is analogous to the size of the browser window
  - Users resize the viewport by resizing the window

Viewport

- On iOS devices, the viewport is a rectangular area that determines how content is laid out and where text wraps on a webpage
  - Unlike on the desktop, the viewport can be larger or smaller than the visible area
  - Users can move the viewport by dragging with a finger
  - Users can change the scale of the viewport by pinching, or double-tapping
- The viewport size and scale can be set by a meta tag to improve the page presentation on iOS devices

Meta Tags

- Meta elements (which appear in the head section) are used to provide metadata about the document
- Meta tags can be used for a variety of purposes
  - Historically they were primarily used for search engine optimization
- Meta tags are one of the primary mechanisms by which can specify iPhone specific metadata about our web pages

Before Adjusting the Viewport

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>Viewport Sample</title>
  </head>
  <body>
    <h1>Viewport Sample</h1>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut ac ligula
      nec felis cursus ullamcorper at in ligula. Ut et tellus nunc, at
      ullamcorper leo. Donec sed mauris vel orci viverra bibendum a eget
      sem. Nullam at ligula eu magna sodales dictum. Curabitur dictum
      pellentesque mi, sit amet tempor purus rutrum sit amet. Nam magna risus,
      tincidunt at commodo non, scelerisque vestibulum ligula. Nullam eu urna
      enim, ac laoreet erat. Aenean dictum auctor diam, et tristique massa
      ultrices sed. Nullam condimentum magna sed sem sagittis quis commodo sem
      interdum. Aenean risus diam, sollicitudin non dapibus ut, tincidunt vel
      lectus. Vivamus eget augue et quam mollis adipiscing.
    </p>
  </body>
</html>
```

Before Adjusting the Viewport

Adjusting the Viewport (HTML)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width" />
    <title>Viewport Sample</title>
  </head>
  <body>
    <h1>Viewport Sample</h1>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut ac ligula
      nec felis cursus ullamcorper at in ligula. Ut et tellus nunc, at
      ullamcorper leo. Donec sed mauris vel orci viverra bibendum a eget
      sem. Nullam at ligula eu magna sodales dictum. Curabitur dictum
      pellentesque mi, sit amet tempor purus rutrum sit amet. Nam magna risus,
      tincidunt at commodo non, scelerisque vestibulum ligula. Nullam eu urna
      enim, ac laoreet erat. Aenean dictum auctor diam, et tristique massa
      ultrices sed. Nullam condimentum magna sed sem sagittis quis commodo sem
      interdum. Aenean risus diam, sollicitudin non dapibus ut, tincidunt vel
      lectus. Vivamus eget augue et quam mollis adipiscing.
    </p>
  </body>
</html>
```

After Adjusting the Viewport

Hiding the Location Bar

Hiding the Location Bar

- At you probably notices, when you load most pages in Mobile Safari, the location bar hangs around at the top of the page until it is scrolled out of view
- Since the toolbar takes up considerable space, it's worthwhile to make it go away
  - We can perform a simple trick to make it "disappear" by scrolling it off the top of the page

Before Hiding the Location Bar

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width" />
    <title>Location Bar Sample</title>
  </head>
  <body>
    <h1>Location Bar Sample</h1>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut ac ligula
      nec felis cursus ullamcorper at in ligula. Ut et tellus nunc, at
      ullamcorper leo. Donec sed mauris vel orci viverra bibendum a eget
      sem. Nullam at ligula eu magna sodales dictum. Curabitur dictum
      pellentesque mi, sit amet tempor purus rutrum sit amet. Nam magna risus,
      tincidunt at commodo non, scelerisque vestibulum ligula. Nullam eu urna
      enim, ac laoreet erat. Aenean dictum auctor diam, et tristique massa
      ultrices sed. Nullam condimentum magna sed sem sagittis quis commodo sem
      interdum. Aenean risus diam, sollicitudin non dapibus ut, tincidunt vel
      lectus. Vivamus eget augue et quam mollis adipiscing.
    </p>
  </body>
</html>
```

Before Hiding the Location Bar

Hiding the Location Bar (HTML)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width" />
    <script src="../common/js/jquery-1.4.4.min.js"></script>
    <script src="script.js"></script>
    <title>Location Bar Sample</title>
  </head>
  <body>
    <h1>Location Bar Sample</h1>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut ac ligula
      nec felis cursus ullamcorper at in ligula. Ut et tellus nunc, at
      ullamcorper leo. Donec sed mauris vel orci viverra bibendum a eget
      sem. Nullam at ligula eu magna sodales dictum. Curabitur dictum
      pellentesque mi, sit amet tempor purus rutrum sit amet. Nam magna risus,
      tincidunt at commodo non, scelerisque vestibulum ligula. Nullam eu urna
      enim, ac laoreet erat. Aenean dictum auctor diam, et tristique massa
      ultrices sed. Nullam condimentum magna sed sem sagittis quis commodo sem
      interdum. Aenean risus diam, sollicitudin non dapibus ut, tincidunt vel
      lectus. Vivamus eget augue et quam mollis adipiscing.
    </p>
  </body>
</html>
```

Hiding the Location Bar (JavaScript)

```
$(function() {

  // if content is less than available height, make room
  if (window.innerHeight < (window.outerHeight + 20)) {
    $('html').css({'min-height':(window.outerHeight + 20) + 'px'});
  }

  // scroll location bar off the screen
  setTimeout(function() {
    window.scrollTo(0, 1);
  }, 1);

});
```

After Hiding the Location Bar

Enabling Web Clips

Enabling Web Clips

- As you're probably aware you can add a web page link to your home screen
- By default, a scaled down version of the page will be used
- You can however, over-ride this and provide your own image...
  - You can set in icon on a per-page basis setting an apple-touch-icon meta tag
  - Or, you can set site wide by simply dropping apple-touch-icon.png or apple-touch-icon-precomposed.png into the root of your web server (similar to fav icons)

Before Enabling Web Clips

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width" />
    <title>Web Clip Example</title>
  </head>
  <body>
    <h1>Web Clip Example</h1>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut ac ligula
      nec felis cursus ullamcorper at in ligula. Ut et tellus nunc, at
      ullamcorper leo. Donec sed mauris vel orci viverra bibendum a eget
      sem. Nullam at ligula eu magna sodales dictum. Curabitur dictum
      pellentesque mi, sit amet tempor purus rutrum sit amet. Nam magna risus,
      tincidunt at commodo non, scelerisque vestibulum ligula. Nullam eu urna
      enim, ac laoreet erat. Aenean dictum auctor diam, et tristique massa
      ultrices sed. Nullam condimentum magna sed sem sagittis quis commodo sem
      interdum. Aenean risus diam, sollicitudin non dapibus ut, tincidunt vel
      lectus. Vivamus eget augue et quam mollis adipiscing.
    </p>
  </body>
</html>
```

Before Enabling Web Clips

Enabling Web Clips (HTML)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width" />
    <link rel="apple-touch-icon"
          href="../common/images/touch-icon-iphone.png" />
    <link rel="apple-touch-icon" sizes="72x72"
          href="../common/images/touch-icon-ipad.png" />
    <link rel="apple-touch-icon" sizes="114x114"
          href="../common/images/touch-icon-iphone4.png" />
    <title>Web Clip Example</title>
  </head>
  <body>
    <h1>Web Clip Example</h1>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut ac ligula
      nec felis cursus ullamcorper at in ligula. Ut et tellus nunc, at
      ...
      interdum. Aenean risus diam, sollicitudin non dapibus ut, tincidunt vel
      lectus. Vivamus eget augue et quam mollis adipiscing.
    </p>
  </body>
</html>
```

touch-icon-iphone.png

touch-icon-ipad.png

touch-icon-iphone4.png

After Enabling Web Clips

Web App Capable

Web App Capable

- You can tell your app to suppress all Safari toolbars if added to and launched from the home screen
  - To do so, you need to specify an apple-mobile-web-app-capable meta tag
  - This allows it to look more like a native app
- If you're displaying in this full screen mode, you also have some control over the status bar's appearance
  - You can set a meta tag by setting an apple-mobile-web-app-status-bar-style meta tag
  - Possible combinations are default (gray), black or black-translucent

Before Making Web App Capable

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width" />
    <link rel="apple-touch-icon"
          href="../common/images/touch-icon-iphone.png" />
    <link rel="apple-touch-icon" sizes="72x72"
          href="../common/images/touch-icon-ipad.png" />
    <link rel="apple-touch-icon" sizes="114x114"
          href="../common/images/touch-icon-iphone4.png" />
    <title>Web App Example</title>
  </head>
  <body>
    <h1>Web App Example</h1>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut ac ligula
      nec felis cursus ullamcorper at in ligula. Ut et tellus nunc, at
      ...
      interdum. Aenean risus diam, sollicitudin non dapibus ut, tincidunt vel
      lectus. Vivamus eget augue et quam mollis adipiscing.
    </p>
  </body>
</html>
```

Before Making Web App Capable

Making Web App Capable

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width" />
    <link rel="apple-touch-icon"
          href="../common/images/touch-icon-iphone.png" />
    <link rel="apple-touch-icon" sizes="72x72"
          href="../common/images/touch-icon-ipad.png" />
    <link rel="apple-touch-icon" sizes="114x114"
          href="../common/images/touch-icon-iphone4.png" />
    <meta name="apple-mobile-web-app-capable" content="yes" />
    <meta name="apple-mobile-web-app-status-bar-style" content="black" />
    <title>Web App Example</title>
  </head>
  <body>
    <h1>Web App Example</h1>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut ac ligula
      nec felis cursus ullamcorper at in ligula. Ut et tellus nunc, at
      ...
      interdum. Aenean risus diam, sollicitudin non dapibus ut, tincidunt vel
      lectus. Vivamus eget augue et quam mollis adipiscing.
    </p>
  </body>
</html>
```

After Making Web App Capable

Geolocation

Geolocation

- Mobile Safari implements the draft Geolocation API Specification
- This allows Mobile Safari to lookup or even track in real-time the location of the device
- Like core location, the underlying lookup technology (GPS, WiFi, IP, etc.) is abstracted away

Geolocation (HTML + JavaScript)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no,
width=device-width, maximum-scale=1.0" />
    <script src="../common/js/jquery-1.4.4.min.js"></script>
    <script>
      $(function() {
        navigator.geolocation.getCurrentPosition(function(position) {
          var lat = position.coords.latitude;
          var long = position.coords.longitude;
          $('#loc').html('lat: ' + lat + '<br />long: ' + long);
        }, function() {
          $('#loc').html('error getting location');
        });
      });
    </script>
    <title>Geolocation</title>
  </head>
  <body>
    <h1>Geolocation</h1>
    <div id="loc">unknown</div>
  </body>
</html>
```

Geolocation

Tracking Location + Google Maps

- Or, a more interesting example where we use the geolocation API in conjunction with Google Maps to track the user's location in real time

Tracking Location (HTML)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no,
width=device-width, maximum-scale=1.0" />
    <script src="http://maps.google.com/maps/api/js?sensor=true"></script>
    <script src="../common/js/jquery-1.4.4.min.js"></script>
    <script src="script.js"></script>
    <link rel="stylesheet"  href="style.css" />
    <title>Geolocation + Google Maps</title>
  </head>
  <body>
    <div id="map"></div>
  </body>
</html>
```

Tracking Location (JavaScript)

```
$(function() {
  document.body.onorientationchange = orient;
  var latlng = new google.maps.LatLng(-34.397, 150.644);
  var myOptions = {
     zoom: 16, center: latlng, mapTypeId: google.maps.MapTypeId.ROADMAP
  };
  map = new google.maps.Map(document.getElementById("map"), myOptions);
  marker = false;
  navigator.geolocation.watchPosition(function(position) {
    var lat = position.coords.latitude;
    var long = position.coords.longitude;
    var myLatLng = new google.maps.LatLng(lat, long);
    map.setCenter(myLatLng);
    if(!marker) {
      marker = new google.maps.Marker({
        position: myLatLng, map: map
      });
    } else {
      marker.setPosition(myLatLng);
    }
  });
  setTimeout(orient, 1);
});
/* ... */
```

Tracking Location (JavaScript)

```
/* ... */

function orient() {
  // portrait
  if(window.orientation % 180 == 0) {
    $('body').height(window.navigator.standalone ? 460 : 416);
  // landscape
  } else {
    $('body').height(window.navigator.standalone ? 300 : 268);
  }
  hide_address_bar();
}

function hide_address_bar () {
  setTimeout(function () {
    window.scrollTo(0, 1);
  }, 1);
}
```

style.css

```
body {
  border: 0;
  margin: 0;
  width:100%;
}

#map {
  width:100%;
  height:100%;
}
```

Tracking Location + Google Maps (ITE Building to Sondheim Hall)

Offline Web Apps

Offline Web Apps

- Mobile Safari has support for HTML5 offline web applications
- One of the long-time limitations of web apps were that they required you to be connected to the network
- HTML5 offline application caching APIs can minimize or altogether eliminate the need for an active network connection

Offline Manifest

- We can tell Mobile Safari to cache certain resources when adding a web app to the home screen
- This manifest contains 2 primary sections...
  - CACHE MANIFEST — the resources that should be cached (able to be run offline)
  - NETWORK — those resources which shouldn't be cached
- This manifest is then specified on the root html tag as a manifest attribute
- To be honored by the client, the manifest must be served up with a content type of text/cache-manifest

Offline Web App (HTML)

```
<!DOCTYPE html>
<html lang="en" manifest="offline.manifest">
  <head>
    <meta charset="utf-8" />
    <meta name="apple-mobile-web-app-capable" content="yes" />
    <meta name="apple-mobile-web-app-status-bar-style" content="black" />
    <meta name="viewport" content="width=device-width">
    <link rel="stylesheet" href="style.css" />
    <script src="../common/js/jquery-1.4.4.min.js"></script>
    <script src="script.js"></script>
    <title>Clock</title>
  </head>
  <body>
    <div id="clock"></div>
  </body>
</html>
```

Offline Web App (.manifest)

```
CACHE MANIFEST

index.html
../common/js/jquery-1.4.4.min.js
script.js
style.css

NETWORK:
```

Offline Web App (CSS)

```css
body {
  font-family: sans-serif;
  color: white;
  background-color: black;
  font-size: 75px;
  text-align: center;
  padding-top: 50px;
}
```

Offline Web App (JavaScript)

```
$(function() {
  setInterval(function() {
    var now = new Date();
    var h = now.getHours();
    var m = now.getMinutes();
    var s = now.getSeconds();
    h = h < 10 ? '0' + h : h;
    m = m < 10 ? '0' + m : m;
    s = s < 10 ? '0' + s : s;
    $('#clock').html(h + ':' + m + ':' + s);
  }, 100);
});
```

Offline Web App

Client-Side Storage

Client-Side Storage

- Historically the only way to store data client-side was to write a cookie — which came with a number of limitations...
  - Limited size, can only save a string, must transmit data to server, etc.
- Mobile Safari supports the HTML5 client-side storage specification
  - This allows you to save data (session & persistent) without the limitations of cookies

Key-Value Storage

- Mobile Safari supports HTML5 key-value storage objects...
  - localStorage
  - sessionStorage
- These basically function like ObjC dictionaries and contain methods for getting and setting keys
  - Contains getter, setter and removal methods

Database Storage

- Mobile Safari supports the HTML5 JavaScript database class
- Provides a relational database that allows you to perform common operations such as...
  - Create and open local databases
  - Create and drop tables
  - Create, read, update and delete records

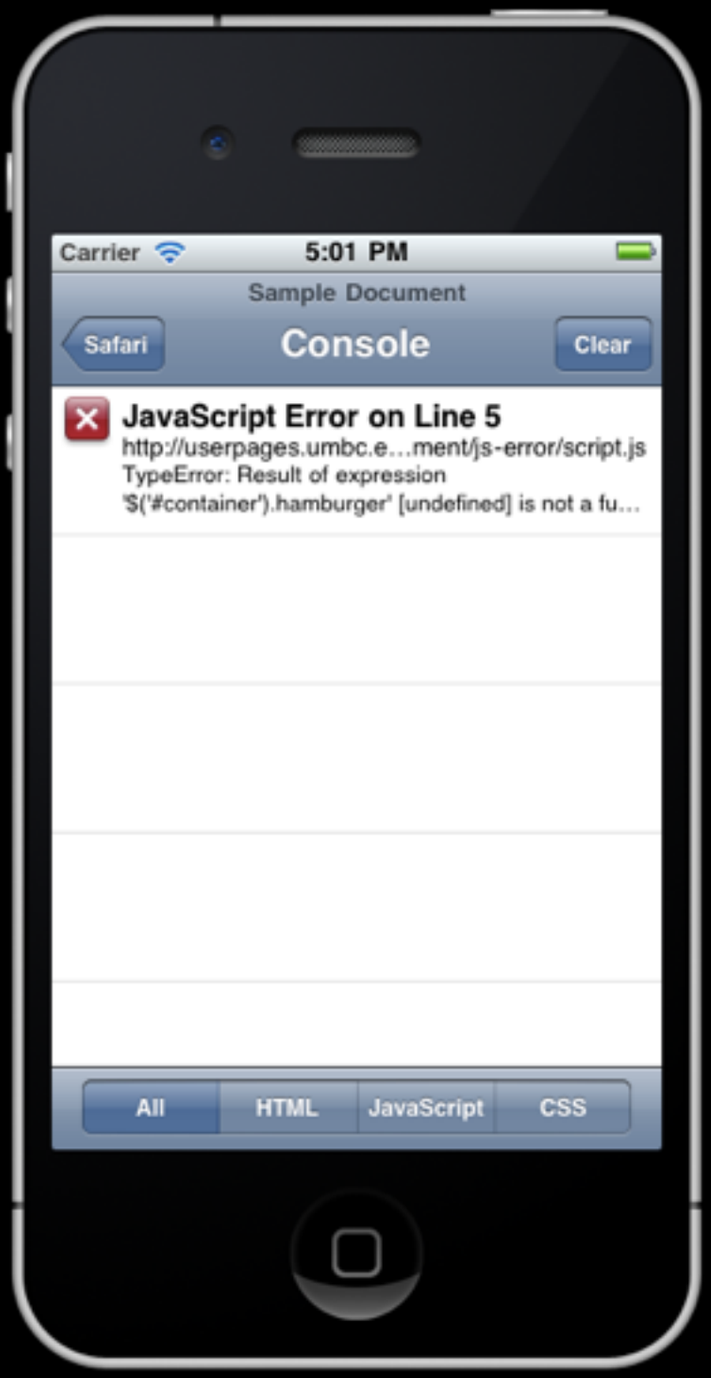
Debugging in Mobile Safari

Debugging in Mobile Safari

- Most web developers have come to appreciate the rise of "in browser" developer tools in the last couple of years
  - i.e. Firebug, Web Developer Add On, MSIE Developer Toolbar, Safari's Develop menu, etc.
- Mobile Safari has a simple built in errors console for recording HTML, CSS & JavaScript errors, though it's disabled by default
  - It can be enabled in the Settings app, under Safari

Debugging in Mobile Safari

Debugging in Mobile Safari



Events

Hovering Events/Styles

- There are several notable differences with respect to events when using Mobile Safari...
  - For example, mouseover & mouseout events do not behave as they would on the desktop — as there's no mouse
- Same thing applies for the :hover pseudo-style

Motion & Movement

- Mobile Safari adds the following JavaScript events to the DOM...
  - orientationchange (on body) — sent when the user changes the orientation of the iOS device
  - devicemotion (new in iOS 4.2.x) — when significant change in motion occurs (encapsulates interval, rotation rate and acceleration)
  - deviceorientation (new in iOS 4.2) — when the device has a gyroscope and while the user is rotating the device around any axis (encapsulates row, pitch and yaw)

Touch Events

- The following single touch events are also added to the DOM...
  - touchstart — when a finger touches the surface
  - touchmove — finger moves on the surface
  - touchend — when a given finger lifts from the surface
  - touchcancel — when the system cancels tracking the touch

Multi-Touch Events

- Multi-touch events are also added...
  - gesturestart — when two or more fingers touch
  - gesturechange — when fingers are moved during a gesture
  - gestureend — when the gesture ends (when there are 1 or 0 fingers touching the surface)

HTML5 Elements

HTML5 Elements

- One of the more significant changes in HTML5 is the addition of several form elements including...
  - Email & URL inputs
  - Number & range inputs
  - Several different date/time inputs
- Mobile Safari honors the newly added email, url and number input types by displaying the appropriate tailored keyboard for the given input
- It also supports the new placeholder attribute (similar to what's available on UITextFields in UIKit)

HTML5 Elements

Stylesheet Device Detection

Stylesheet Device Detection

- You can selectively serve up a stylesheet that makes the most sense for the given device by specifying the media attribute like so...
- iPhone & iPod touch

```
media="only screen and (max-device-width: 480px)"
```

- Retina display iPhone & iPod touch

```
media="only screen and (max-device-width: 480px) and (-webkit-min-device-pixel-ratio: 2)"
```

- Everything else...

```
media="screen and (min-device-width: 481px)"
```

Stylesheet Device Detection (HTML)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>Sample Page</title>
    <link media="only screen and (max-device-width: 480px)" href="mobile.css"
rel="stylesheet" />
    <link media="only screen and (max-device-width: 480px) and (-webkit-min-device-
pixel-ratio: 2)" href="retina.css" rel="stylesheet" />
    <link media="screen and (min-device-width: 481px)" href="normal.css"
rel="stylesheet" />
    <meta name="viewport" content="width=device-width" />
  </head>
  <body>
    <div id="header">
      <h1>Sample Page</h1>
    </div>
<!-- ... -->
```

Stylesheet Device Detection (HTML)

```
<!-- ... -->
    <div id="sidebar">
      <ul>
        <li>
          <a href="#">Link 1</a>
          <ul>
            <li><a href="#">Link A</a></li>
            <li><a href="#">Link B</a></li>
            <li><a href="#">Link C</a></li>
          </ul>
        </li>
        <li>
          <a href="#">Link 2</a>
          <ul>
            <li><a href="#">Link D</a></li>
            <li><a href="#">Link E</a></li>
            <li><a href="#">Link F</a></li>
          </ul>
        </li>
<!-- ... -->
```

Stylesheet Device Detection (HTML)

```
<!-- ... -->
        <li>
          <a href="#">Link 3</a>
          <ul>
            <li><a href="#">Link G</a></li>
            <li><a href="#">Link H</a></li>
            <li><a href="#">Link I</a></li>
          </ul>
        </li>
      </ul>
    </div>
    <div id="content">
       <!-- ... -->
    </div>
    <div id="footer">
      <p>
        Copyright &copy; 2010 Foo Bar Baz
      </p>
    </div>
  </body>
</html>
```

Stylesheet Device Detection (normal.css)

```
body {
    margin: 0;
    padding: 0;
    font-family: Helvetica, Arial, sans-serif;
}
p {
    margin: 2em 0;
    line-height: 150%;
}
#header {
    border-bottom: 5px solid #ddbb2b;
    margin: 0;
    padding: 10px 10px 10px 77px;
    background: #f5ca5c url('../common/images/touch-icon-iphone.png') no-repeat 10px
50%;
}
#header h1 {
    margin: 0;
    padding: 10px;
}

/* ... */
```

Stylesheet Device Detection (normal.css)

```
/* ... */

#sidebar {
    float: left;
    width: 200px;
    margin: 0;
    padding: 10px;
}
#sidebar ul {
    list-style: none;
    padding: 0;
}
#sidebar ul ul {
    margin-left: 20px;
}

/* ... */
```

Stylesheet Device Detection (normal.css)

```
/* ... */

#sidebar li a {
    text-decoration: none;
    display: block;
    color: #222;
    background-color: #ddd;
    margin: 15px;
    padding: 10px;
    border: 1px solid #bbb;
    border-left: 3px solid #bbb;
}
#sidebar li a:hover {
    background-color: #eee;
}
#content {
    margin-left: 220px;
}
#footer {
    clear: both;
    background-color: #f5ca5c;
    border-top: 5px solid #ddbb2b;
    padding: 1em;
}
```

Stylesheet Device Detection (mobile.css)

```
body {
    margin: 0;
    padding: 0;
    font-family: Helvetica, Arial, sans-serif;
}
p {
    margin: 1em 0;
    line-height: 150%;
}
#header {
    border-bottom: 2px solid #ddbb2b;
    padding: 2px 2px 2px 61px;
    background: #f5ca5c url('../common/images/touch-icon-iphone.png') no-repeat 10px
50%;
}
#header h1 {
    margin: 0;
    padding: .25em;
    font-size: 2em;
}

/* ... */
```

Stylesheet Device Detection (mobile.css)

```
/* ... */

#sidebar {
    margin: 0;
    border-bottom: 1px solid #bbb;
}
#sidebar ul {
    list-style: none;
    margin: 0;
    background-color: #ddd;
    text-align: center;
}
#sidebar ul ul {
    display: none;
}
#sidebar li {
    display: inline;
    margin: 0 .5em;
}

/* ... */
```

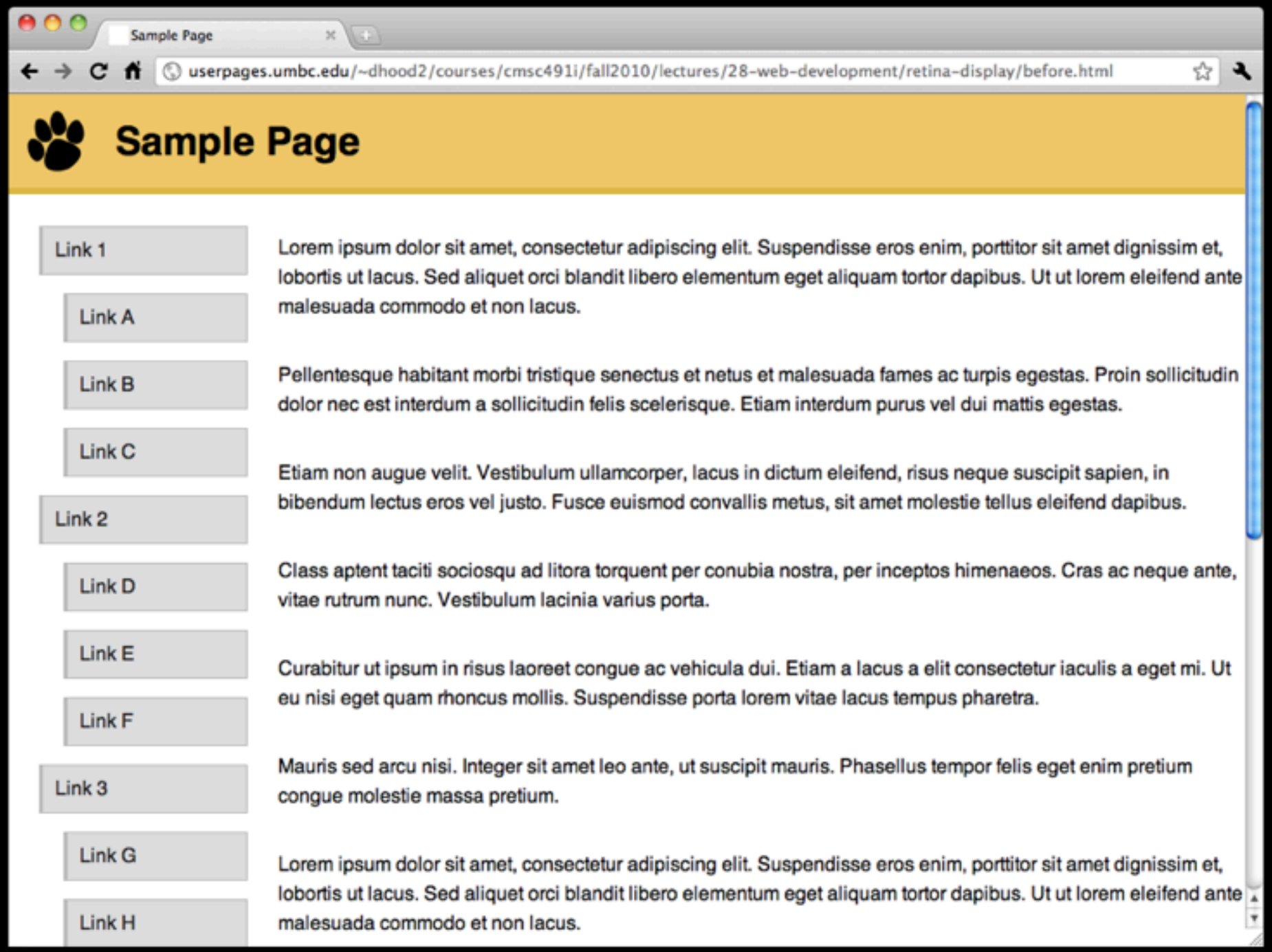
Stylesheet Device Detection (mobile.css)

```
/* ... */

#sidebar li a {
    text-decoration: none;
    color: #222;
}
#content {
    padding: .5em;
}
#footer {
    background-color: #f5ca5c;
    border-top: 2px solid #ddbb2b;
}
#footer p {
    margin: 0;
    padding: .25em;
}
```

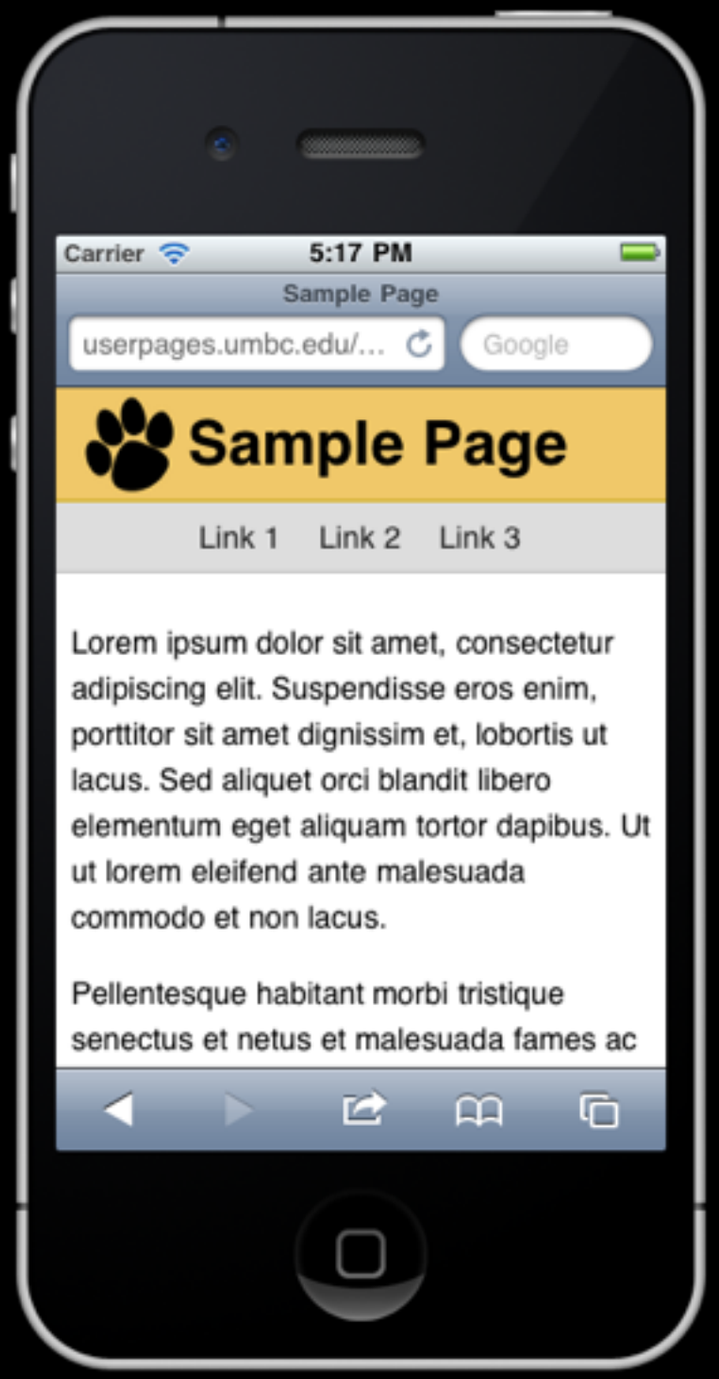
Stylesheet Device Detection (retina.css)

```
#header {
    background: #f5ca5c url('../common/images/touch-icon-iphone4.png') no-repeat
10px 50%;
    background-size: 57px 57px;
}
```

Chrome (Desktop)



iPhone


Carrier
5:17 PM
Sample Page
userpages.umbc.edu/...
Google
Sample Page
Link 1
Link 2
Link 3
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse eros enim, porttitor sit amet dignissim et, lobortis ut lacus. Sed aliquet orci blandit libero elementum eget aliquam tortor dapibus. Ut ut lorem eleifend ante malesuada commodo et non lacus.
Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac

Performance

Performance

- The throughput on a mobile network is obviously slower than a WiFi or wired network connection
- As such, when building mobile web apps there are some things that we need to be extra diligent about

Optimize at the Protocol Level (HTTP)

- Utilize a Content Delivery Network (CDN) so that the content is "closer" to the origin of the request on the net
- Set an Expires or a Cache-Control header for resources that do not frequently change — allows the client to cache
- Gzip delivery — most web servers are capable of detecting if the client supports gzip-ing the response, if so doing so can greatly compress most webpages
- Minimize redirects in your server-side applications
- Reduce the size of cookies, as they're passed on all requests to the server

JavaScript Optimization

- Minimize number of requests
- Minify JavaScript using a JavaScript minification tool such as YUI Compressor or the Google Closure Compiler
- Do not repeat common JavaScript on multiple pages, externalize it so the client can cache a single copy
- Move scripts at the bottom of HTML document

CSS

- Minimize number of requests
- Minify CSS using a CSS compression tool
- Do not repeat common CSS on multiple pages, externalize it so the client can cache a single copy
- Avoid including stylesheets via the @import in favor of using the link tag
- Utilize CSS sprites to minimize the quantity of image requests
- Optimize images using an image compression tool such as Pngcrush, OptiPNG, PNGOUT, jpegtran, or jpegoptim

Mobile HTML/CSS/JavaScript Frameworks

Frameworks

- Early frameworks
  - iUI
  - WebApp.Net
- Newer frameworks
  - jQTouch
  - Sencha Touch
  - jQuery Mobile

jQuery Mobile

- jQuery Mobile is a “touch-optimized web framework for smartphones & tablets”

jQuery Mobile Example

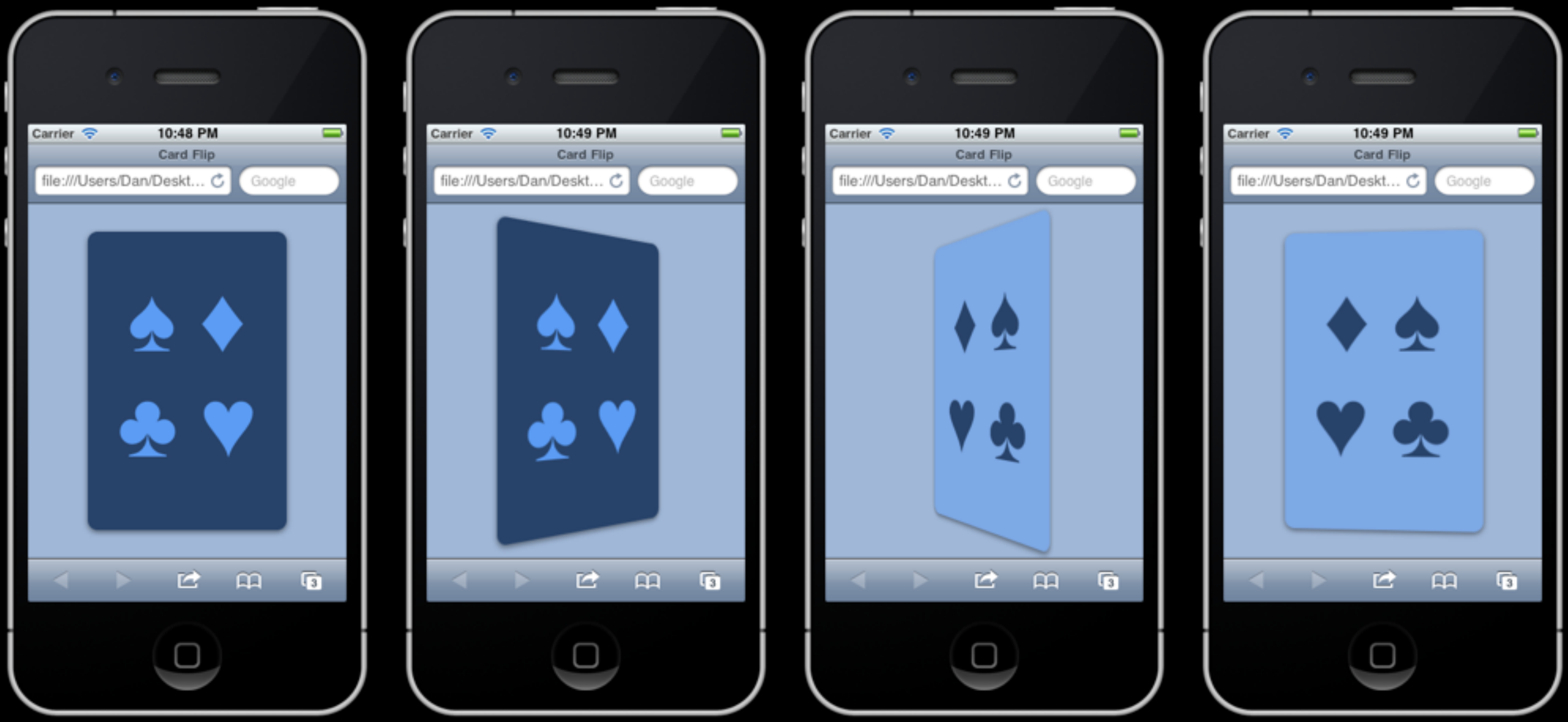
jQuery Mobile Example

CSS Transitions

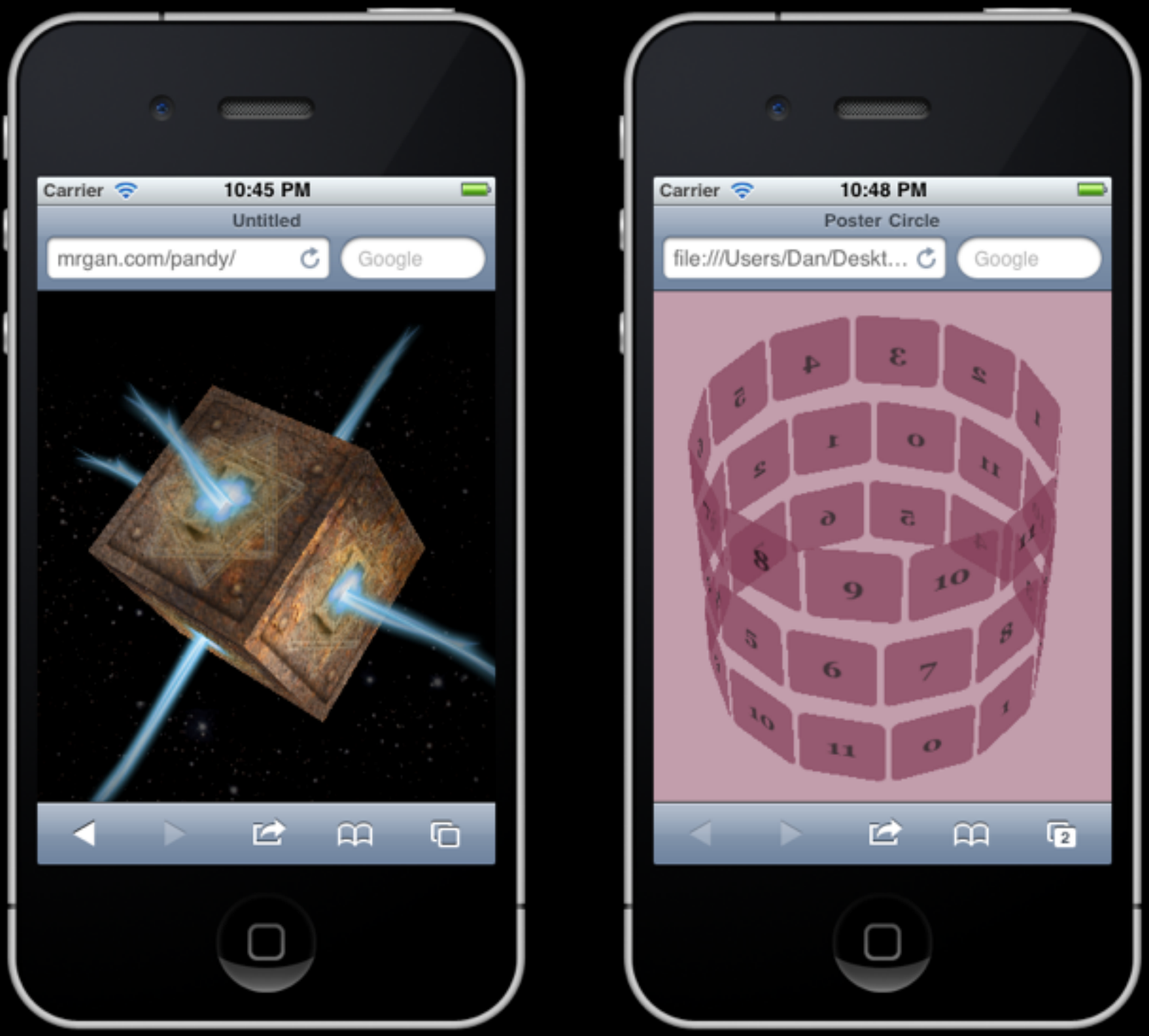
CSS Transitions

- Mobile Safari has added CSS support for transitions
- Normally when you switch a CSS property on an object it happens right away
- However, transitions allow you to animate from the old state to a new state (much like Core Animation)

CSS Transitions



CSS Transitions

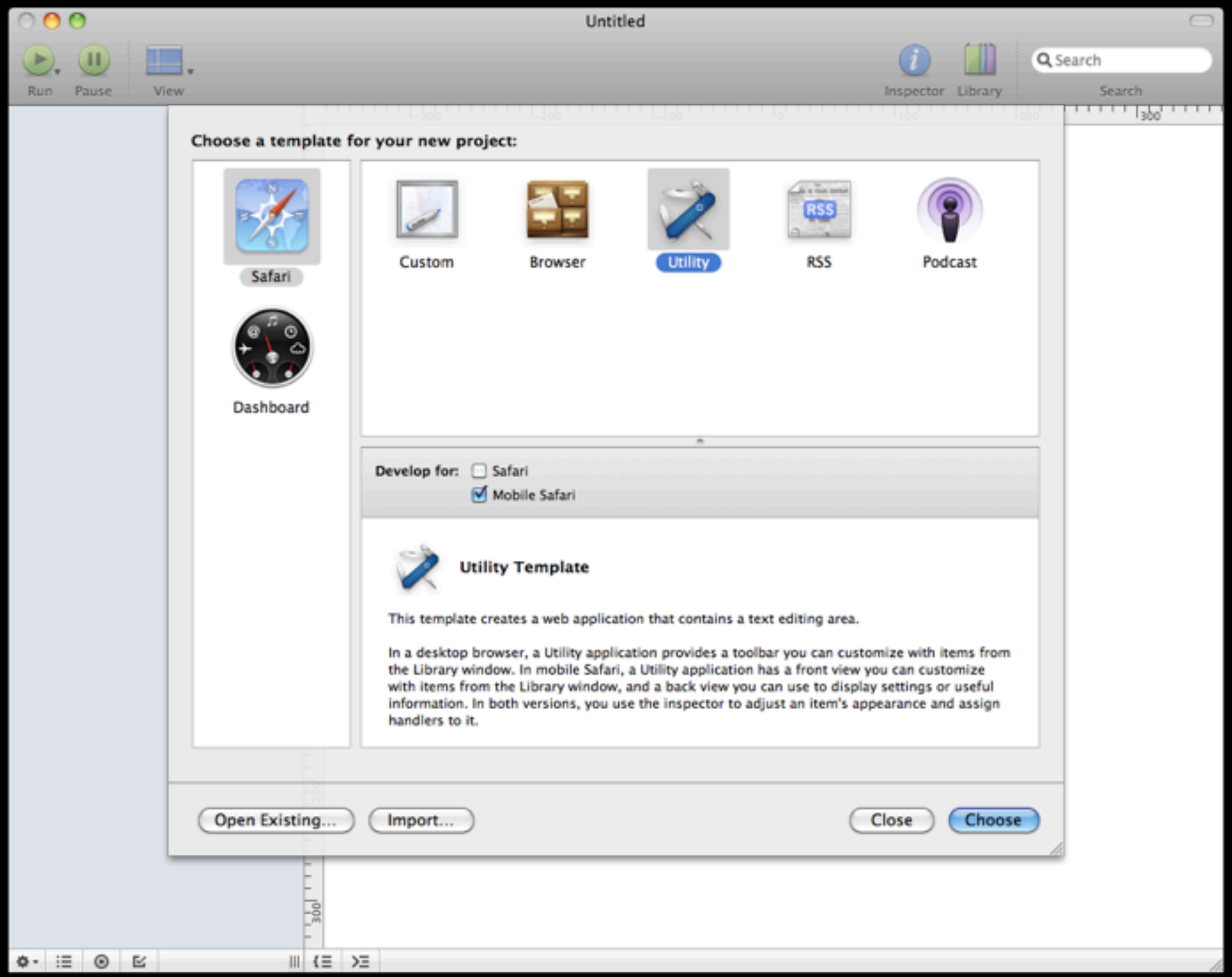


Dashcode

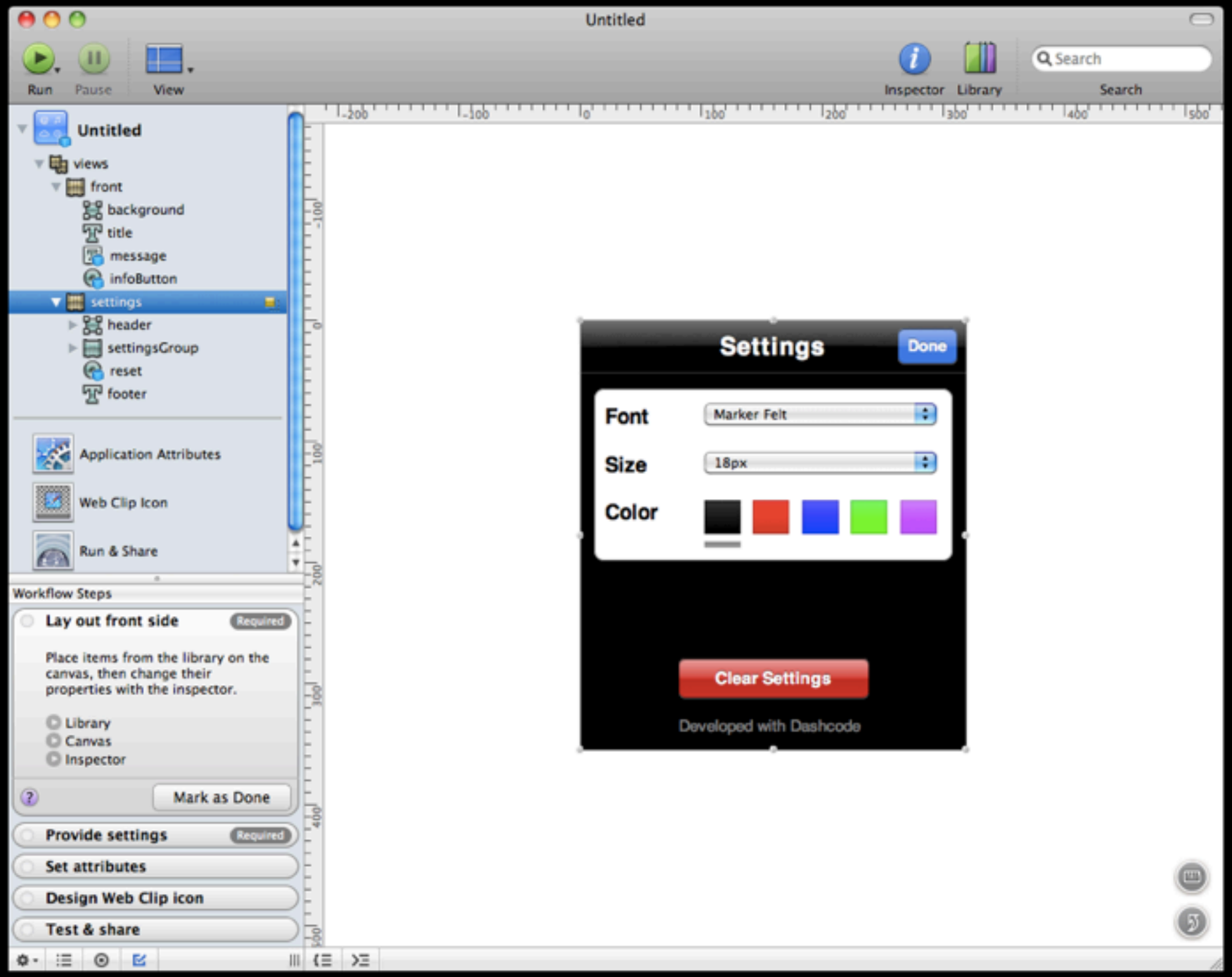
Dashcode

- Apple ships a program called Dashcode as part of the iOS SDK
- Think of it as a combination Xcode and Interface Builder for web apps

Dashcode


Untitled
Run
Pause
View
Inspector
Library
Search
Search
Choose a template for your new project:
Safari
Dashboard
Custom
Browser
Utility
RSS
Podcast
Develop for:
Safari
Mobile Safari
Utility Template
This template creates a web application that contains a text editing area.
In a desktop browser, a Utility application provides a toolbar you can customize with items from the Library window. In mobile Safari, a Utility application has a front view you can customize with items from the Library window, and a back view you can use to display settings or useful information. In both versions, you use the inspector to adjust an item's appearance and assign handlers to it.
Open Existing...
Import...
Close
Choose

Dashcode


Untitled
Run
Pause
View
Inspector
Library
Search
Search
Untitled
views
front
background
title
message
infoButton
settings
header
settingsGroup
reset
footer
Application Attributes
Web Clip Icon
Run & Share
Workflow Steps
Lay out front side
Required
Place items from the library on the canvas, then change their properties with the inspector.
Library
Canvas
Inspector
Mark as Done
Provide settings
Required
Set attributes
Design Web Clip icon
Test & share
Settings
Done
Font
Marker Felt
Size
18px
Color
Clear Settings
Developed with Dashcode

Additional Resources

- HTML5 Specification
  - http://dev.w3.org/html5/spec/
  - http://www.whatwg.org/specs/web-apps/current-work/
- Geolocation API Specification
  - http://dev.w3.org/geo/api/
- Web Storage Draft
  - http://dev.w3.org/html5/webstorage/
- Offline Web Applications
  - http://www.w3.org/TR/offline-webapps/

Additional Resources

- Safari Web Content Guide
  - http://developer.apple.com/library/safari/#documentation/AppleApplications/Reference/SafariWebContent/
- Client-Side Storage and Offline Applications Programming Guide
  - http://developer.apple.com/library/safari/#documentation/iPhone/Conceptual/SafariJSDatabaseGuide/
- Safari Visual Effects Guide
  - http://developer.apple.com/library/safari/#documentation/InternetWeb/Conceptual/SafariVisualEffectsProgGuide/

Additional Resources

- Dashcode User Guide
  - http://developer.apple.com/safari/library/documentation/AppleApplications/Conceptual/Dashcode_UserGuide/Contents/Resources/en.lproj/
- Safari DOM Additions for iOS
  - http://developer.apple.com/library/safari/navigation/#section=Libraries&topic=Safari%20DOM%20Additions%20for%20iOS
- Safari Reference Library
  - http://developer.apple.com/library/safari/navigation/

Additional Resources

- Dive into HTML5 (free online book)
  - http://diveintohtml5.org/
- Building iPhone Apps with HTML, CSS, and JavaScript
  - http://ofps.oreilly.com/titles/9780596805784/