

Going Mobile with jQuery

Nathaniel T. Schutta
@ntschutta

The Plan

- Why mobile?
- Why jQuery Mobile?
- Basics
- Events
- Methods, layout, themes
- Native vs. HTML

Assume you have:
browser, text editor.

Copy zip.

Or clone it from github.

```
git clone git://github.com/ntschutta/  
jQuery_mobile_workshop
```

Extract to...somewhere ;)

Shout if you have ??s

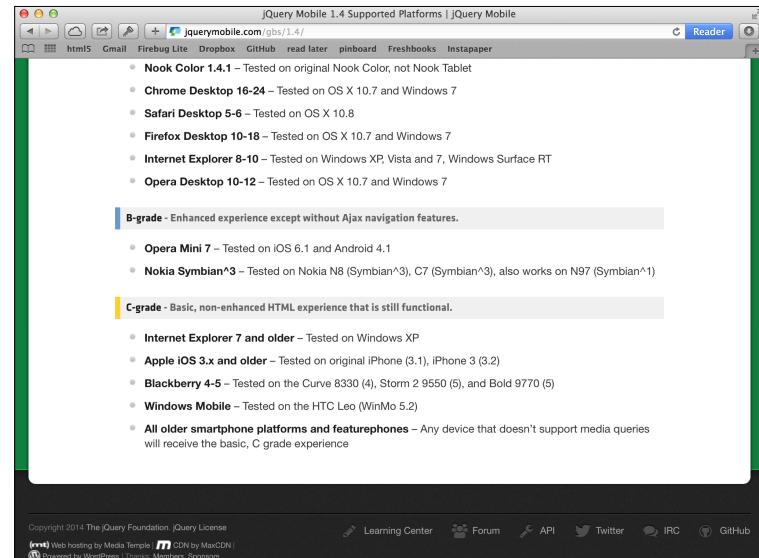
jQuery Mobile.

Why jQuery Mobile?

Unified UI.

Support for a wide variety
of hardware.

Support for vast majority
of mobile browsers.



Markup driven.

Build on clean HTML.

Progressive enhancement.

Responsive web design.

Includes responsive widgets.

Supports accessibility.

Built on jQuery core.

Familiar syntax.

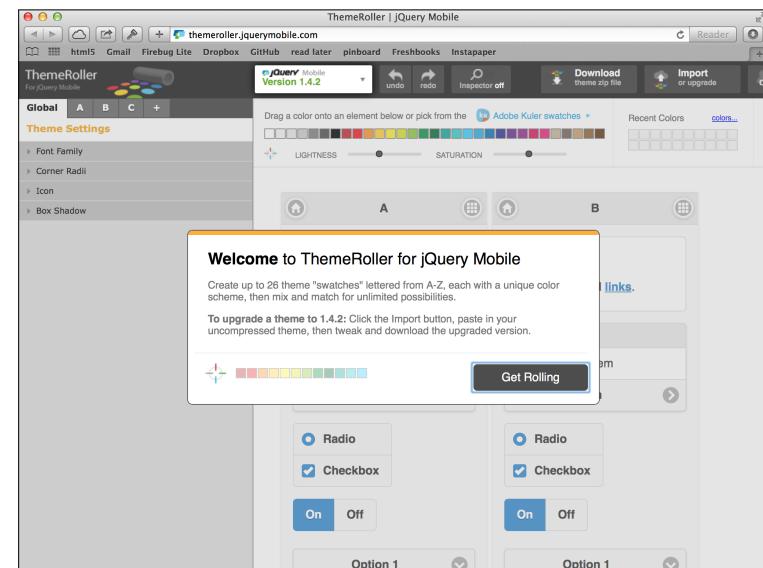
Relatively small...

Use Download Builder to
choose what is included.

HTML5.

Highly customizable UI.

Leverages ThemeRoller.



Widget auto-initialization.

Happens before
document.ready.

Touch events.

Plugins!

Page Structure.

HTML5.

Divs with data-roles.

And a little CSS...

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="../javascript/jqueryMobile-1.4.2/jquery.mobile-1.4.2.min.css" />
<script src="../javascript/jquery-2.1.0.js"></script>
<script src="../javascript/jqueryMobile-1.4.2/jquery.mobile-1.4.2.min.js"></script>
</head>
<body>
<div data-role="page">
<div data-role="header">
<h1>Page Title</h1>
</div>
<div role="main" class="ui-content">
<p>Page content goes here.</p>
</div>
<div data-role="footer">
<h4>Page Footer</h4>
</div>
</div>
</body>
</html>
```

Could also reference the CDN for the jQuery files.

Viewport - prevents the “virtual” page width...

Without, many browsers go to 900 pixels.

Each “page” is a div.

Containing any valid HTML markup.

Divs for header and footer
with proper data-role.

The content used to be a
data-role (and it'll work!)

But now it's CSS class.

Lab time!

Page Structure

- Create a basic page with a header, content area and a footer
- [\\${extract}/jQuery_mobile_workshop/labs/page_structure/one_page.html](#)

One document can contain
multiple pages.

Stack the divs.

Just give each page a unique ID.

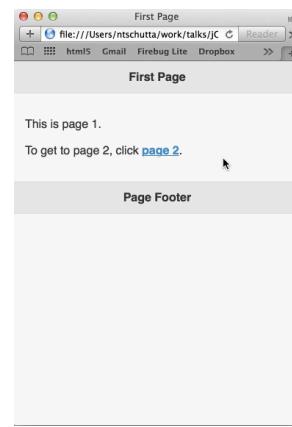
IDs must be unique across your entire site.

Take a look at the DOM...

Actually multiple “pages” at any given time.

Note: a lot of this is actually optional.

But it does provide some helpful structure.



```
<div data-role="page" id="page1">
  <div data-role="header">
    <h1>First Page</h1>
  </div>
  <div role="main" class="ui-content">
    <p>This is page 1.</p>
    <p>To get to page 2, click <a href="#page2" data-transition="flip">page 2</a>.</p>
  </div>
  <div data-role="footer">
    <h4>Page Footer</h4>
  </div>
</div>

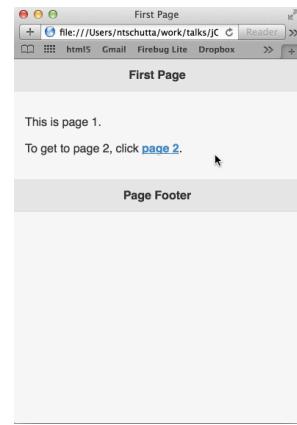
<div data-role="page" id="page2">
  <div data-role="header">
    <h1>Second Page</h1>
  </div>
  <div role="main" class="ui-content">
    <p>This is page 2.</p>
    <p>More stuff goes here.</p>
  </div>
  <div data-role="footer">
    <h4>Page Footer</h4>
  </div>
</div>
```

Page transitions.

CSS based.

- slide
- slideup
- slidedown
- slidefade
- pop
- fade
- flip
- turn
- flow
- none

Add a data transition attribute.



```
<div data-role="page" id="page1">
  <div data-role="header">
    <h1>First Page</h1>
  </div>
  <div role="main" class="ui-content">
    <p>This is page 1.</p>
    <p>To get to page 2, click <a href="#page2" data-transition="flip">page 2</a>.</p>
  </div>
  <div data-role="footer">
    <h4>Page Footer</h4>
  </div>
</div>

<div data-role="page" id="page2">
  <div data-role="header">
    <h1>Second Page</h1>
  </div>
  <div role="main" class="ui-content">
    <p>This is page 2.</p>
    <p>More stuff goes here.</p>
  </div>
  <div data-role="footer">
    <h4>Page Footer</h4>
  </div>
</div>
```

Lab time!

Page Structure - 2

- Add a second page
- Create a link from the first to the second
- Try different transitions
- \${extract}/jQuery_mobile_workshop/labs/page_structure/two_page.html

Why data-x?

HTML5.

data- is a storage area.

Markup is valid.

And you still get
the bits you need.

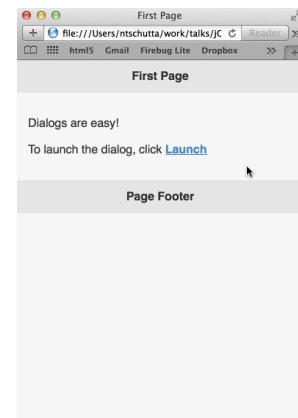
Dialogs.

Any page can be a dialog.

We modify the link
again...dialog relation.

Standard page
transitions work.

Pop, slidtdown and flip are
most “dialog like.”



```
<div data-role="page" id="page1">
  <div data-role="header">
    <h1>First Page</h1>
  </div>
  <div role="main" class="ui-content">
    <p>Dialogs are easy!</p>
    <p>To launch the dialog, click<br/>
      <a href="#" data-transition="pop" data-rel="dialog">Launch</a></p>
  </div>
  <div data-role="footer">
    <h4>Page Footer</h4>
  </div>
</div>
```

Any old link can
close a dialog...

A close button comes along for the ride.

If there's a header.

Move it to the right side:
`data-close-btn="right"`

In the dialog container.

Don't want a close button? No problem.

`data-close-btn="none"`

Want a cancel button? Link
to the launching page...

And add: `data-rel="back"`

Dialogs can be chained.

And themed.

Don't like the default CSS?

Override:
`.ui-dialog-contain`

Lab time!

Linked pages are loaded
via an Ajax call.

Allows for bookmarking!

Dialogs

- Create a basic page that links to the existing dialog.html
- Try different transitions
- Add a cancel link to the dialog
- `#{extract}jquery_mobile_workshop/labs/dialog/page_with_dialog.html`

Navigation modifies the
location hash.

Don't want an Ajax call?

Modify link attributes.

Add: rel=external or
specify a target.

Toolbars.

Headers and footers.

Header = page title and up
to two buttons.

Footer = text and buttons.

Nav bar widget.

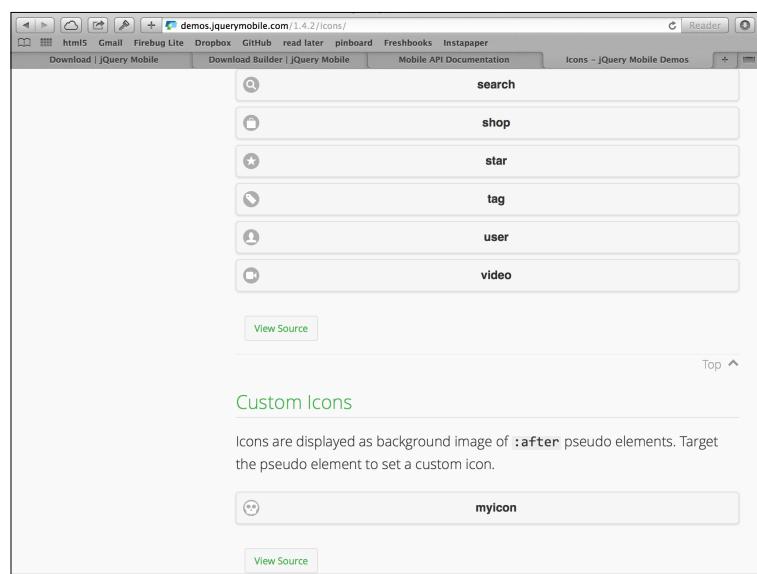
Where do the buttons
come from again?

First link - left button.
Second link - right button.

And we get icons!

Sprite.

Simply add an attribute.



Don't like one of those?

Make your own!

SVG icons work.

Default position is left.

But you can move it...

data-iconpos

Right, top, bottom...

Oh, **notext** gives
you just an icon.

And much more!

Navbars, persistent
footers, fullscreen...

Toolbars can be
external for reuse.

Quite powerful!

By default, jQuery won't add a back button.

But you can make it add one.

In the header...

`data-add-back-btn="true"`

Don't like "Back"?

`data-back-btn-text="Foo"`

Lab time!

Toolbars

- ➊ jQuery Mobile can add a back button
- ➋ Have jQuery Mobile create the back button
- ➌ Change the back button's text
- ➍ `#{extract}/jQuery_mobile_workshop/labs/toolbars/two_page_back_button.html`

Toolbars - 2

- ➊ Add a left and right button to the header
- ➋ Add a few buttons to the footer
- ➌ Experiment with different icons
- ➍ `#{extract}/jQuery_mobile_workshop/labs/toolbars/two_page_toolbar.html`

Forms!

Basically the same as any old web app.

Form tag, action, method.

One caveat...

Control ids must be unique
across site.

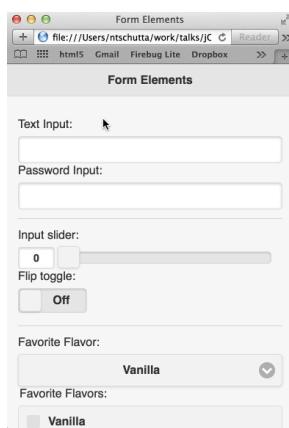
DOM can have multiple
pages present.

Field containers get
special styling.

Enhanced controls.

Degrades gracefully.

- text
- search
- slider
- flip toggle switch
- radio buttons
- checkbox
- multiselects



```
<form action="doFoo" method="post">
  <div data-role="fieldcontain">
    <label for="name">Text Input:</label>
    <input type="text" name="name" id="name" value="" />
    <label for="password">Password Input:</label>
    <input type="password" name="password" id="password" value="" />
  </div>
  <div data-role="fieldcontain">
    <label for="slider">Input slider:</label>
    <input type="range" name="slider" id="slider" value="0" min="0" max="100" />
    <label for="slider2">Flip toggle:</label>
    <input type="checkbox" id="slider2" checked="checked" />
    <select name="slider2" id="slider2" data-role="slider">
      <option value="off">Off</option>
      <option value="on">On</option>
    </select>
  </div>
  <div data-role="fieldcontain">
    <label for="select-choice-1" class="select">Favorite Flavor:</label>
    <select name="select-choice-1" id="select-choice-1" data-role="none">
      <option value="Vanilla">Vanilla</option>
      <option value="Chocolate">Chocolate</option>
      <option value="Neapolitan">Neapolitan</option>
      <option value="Coffee">Coffee</option>
    </select>
  </div>
</form>
```

Lab time!

Forms

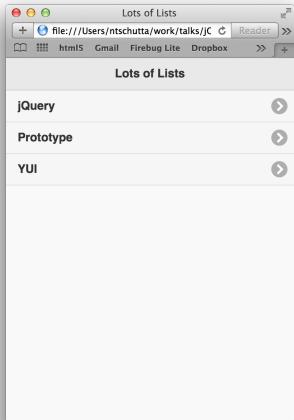
- Add a form
- Experiment with various form elements
- Add a field container
- Add a flip toggle switch
- \${extract}/jQuery_mobile_workshop/labs/forms/forms.html

Lists.

Tables of data.

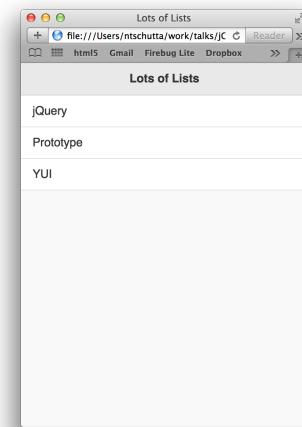
Very common.

Display, navigation, etc.



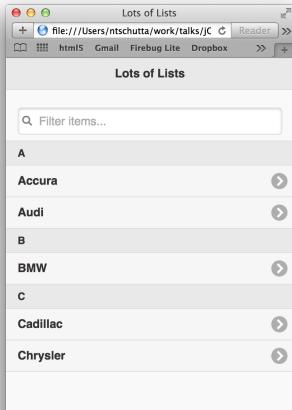
```
<div data-role="page">
  <div data-role="header">
    <h1>Lots of Lists</h1>
  </div>
  <div role="main" class="ui-content">
    <ul data-role="listview">
      <li><a href="jquery.html">jQuery</a></li>
      <li><a href="prototype.html">Prototype</a></li>
      <li><a href="yui.html">YUI</a></li>
    </ul>
  </div>
</div>
```

Tables can be read only.



```
<ul data-role="listview">
  <li>jQuery</li>
  <li>Prototype</li>
  <li>YUI</li>
</ul>
```

Or have dividers
and filters...



```
<ul data-role="listview" data-autodividers="true" data-filter="true">
  <li><a href="accura.html">Accura</a></li>
  <li><a href="audi.html">Audi</a></li>
  <li><a href="bmw.html">BMW</a></li>
  <li><a href="cadillac.html">Cadillac</a></li>
  <li><a href="chrysler.html">Chrysler</a></li>
</ul>
```

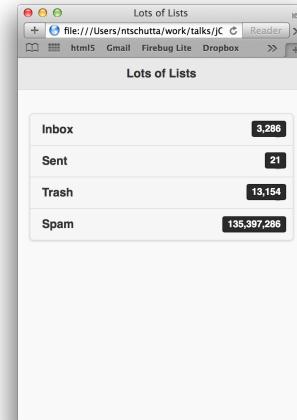
You can set the divider selector text if you want.

Defaults to uppercased first letter of the text.

You can also turn a list item into a divider.

That's not all!

We can add a count.



```
<ul data-role="listview" data-count-theme="b" data-inset="true">
  <li><a href="inbox.html">Inbox</a><span class="ui-li-count">3,286</span></li>
  <li><a href="sent.html">Sent</a><span class="ui-li-count">21</span></li>
  <li><a href="trash.html">Trash</a><span class="ui-li-count">13,154</span></li>
  <li><a href="spam.html">Spam</a><span class="ui-li-count">135,397,286</span></li>
</ul>
```

And on and on!

You can add icons,
thumbnails, split buttons.

Even formatted content.

Lab time!

Lists

- Create a page with a standard list
- Create a page with a read only list
- Add list dividers
- Add “counts”
- Convert a list to a numbered list
- \${extract}/jQuery_mobile_workshop/labs/lists/*.html

Custom events.

Adds a set of custom events.

Bind with `on()` or `bind()`.

Touch events.

- tap
- taphold
- swipe
- swipeleft
- swiperight

Orientation/Scrolling.

- orientationchange
- scrollstart
- scrollstop

Page show and hide.

- pagebeforeshow
- pagebeforehide
- pageshow
- pagehide

Page transitions involve 4 page events...

Two for each page.

Page events reference to next or previous page.

Page initialization events.

Markup “converted” to custom controls.

You can add custom code before/after init.

You can also add your own init code.

And you can stop the auto init if you want.

Page events only fire ONCE per page.

Not the same as hide/show.

Page initialization.

- pagebeforecreate
- pagecreate

To manipulate markup...

...bind to
`pagebeforecreate`

Do what you need to do.

Methods.

`$.mobile` exposes several
methods/properties.

`$.mobile.changePage`

Has been deprecated...

You can use the `change()` method on page instead.

`$.mobile.path.*`

Variety of methods to ask about current location, URL.

`$.mobile.silentScroll`

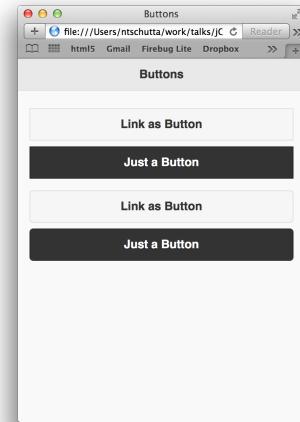
Scrolls without firing scroll events.

Odds and Ends.

Buttons.

It's just CSS!

Add it to a link,
create a button.

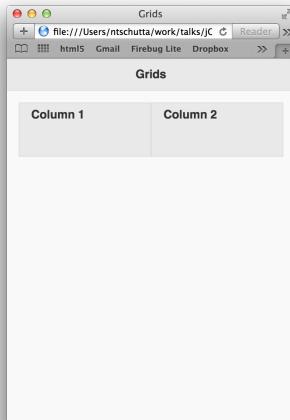


```
<div data-role="page">
  <div data-role="header">
    <h1>Buttons</h1>
  </div>
  <div role="main" class="ui-content">
    <a href="#" class="ui-btn">Link as Button</a>
    <button class="ui-btn ui-btn-b">Just a Button</button>
    <br>
    <a href="#" class="ui-btn ui-corner-all">Link as Button</a>
    <button class="ui-btn ui-corner-all ui-btn-b">Just a Button</button>
  </div>
</div>
```

Grids.

2 column: ui-grid-a
3 column: ui-grid-b

Grids are 100% width,
invisible, no margins/padding.



```
<div class="ui-grid-a">
  <div class="ui-block-a"><div class="ui-bar ui-bar-a" style="height:60px">Column 1</div></div>
  <div class="ui-block-b"><div class="ui-bar ui-bar-b" style="height:60px">Column 2</div></div>
```

Can be combined
with themes.

Works with any container.

Also has support for 3, 4
an 5 column grids.

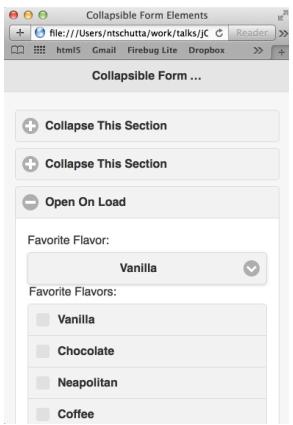
You can even have
multi row grids.

Collapsible content.

Wait for it...

`data-role="collapsible"`

Add any header element.



```
<div data-role="collapsible">
  <h3>Collapse This Section</h3>
  <div data-role="fieldcontain">
    <label for="slider">Input slider:</label>
    <input type="range" name="slider" id="slider" value="0" min="0" max="100" />
    <label for="slider2">Flip toggle:</label>
    <select name="slider2" id="slider2" data-role="slider">
      <option value="off">Off</option>
      <option value="on">On</option>
    </select>
  </div>
</div>
<div data-role="collapsible" data-collapsed="false">
  <h3>Open On Load</h3>
  <div data-role="fieldcontain">
    <label for="select-choice-1" class="select">Favorite Flavor:</label>
    <select name="select-choice-1" id="select-choice-1">
      <option value="Vanilla">Vanilla</option>
      <option value="Chocolate">Chocolate</option>
      <option value="Neapolitan">Neapolitan</option>
    </select>
  </div>
</div>
```

Lab time!

Can also create an accordion.

Collapsible Sets

- Add collapsable sets to the form
- Have one (or more) sections open on load
- \${extract}/jQuery_mobile_workshop/labs/collapsible/collapsible.html

data-role=
"collapsibleset"

Themes.

Strength of jQuery.

Unified visual design.

Uses CSS3 properties.

No images for rounded corners, gradients, etc.

Themes use color swatches.

Can mix and match.

Themes are open ended,
allows for variety.

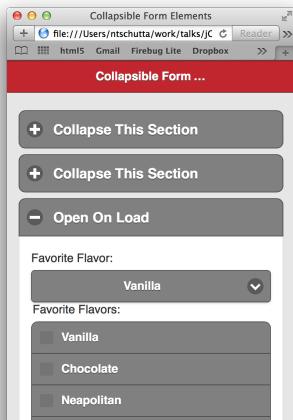
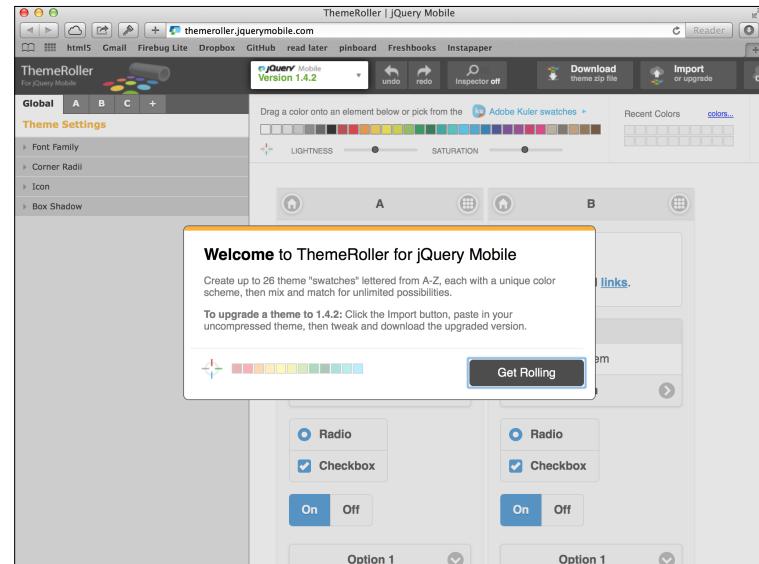
CSS3 gradients for
background.

Icon set.

Separation of color
and texture...

Default theme 2
swatches: a, b.

ThemeRoller For jQuery Mobile.



```
<title>Collapsible</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="../javascript/jqueryMobile-1.4.2/jquery.mobile-1.4.2.min.css" />
<link rel="stylesheet" href="../javascript/jqueryMobile-1.4.2/test_theme.min.css" />
<link rel="stylesheet" href="../javascript/jqueryMobile-1.4.2/jquery.mobile.icons.min" />
<script src="../javascript/jquery-2.1.0.js"></script>
<script src="../javascript/jqueryMobile-1.4.2/jquery.mobile-1.4.2.min.js"></script>
</head>
<body>
<div data-role="page" data-theme="c">
<div data-role="header">
<h1>Collapsible Form Elements</h1>
</div>
<div role="main" class="ui-content">
```

Lab time!

Themes

- ➊ Wifi willing...
- ➋ <http://themeroller.jquerymobile.com>
- ➌ Using jQuery ThemeRoller, create your own theme
- ➍ Try your theme out on various labs
- ➎ \${extract}/jQuery_mobile_workshop/labs/collapsible/collapsible.html

Swatches default styles
for buttons et al.

Buttons in a bar
automatically match.

Changes cascade down.

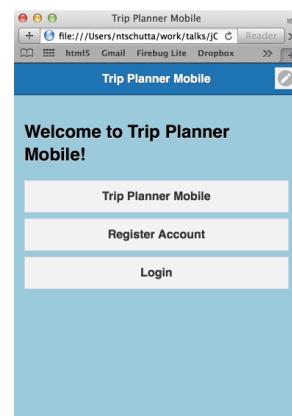
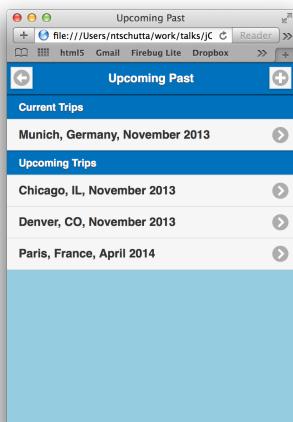
Putting it all together!

I love TriplIt!

<http://www.tripit.com>

Incredibly useful for
anyone that travels.

Great website, also has
a mobile app.



Lab time!

Trip Planner

- Using what we've learned, create a jQuery Mobile based "trip planner"
- Splash page
- Trip list
- Trip details
- \${extract}/jQuery_mobile_workshop/labs/trip_planner/trip_planner.html

Is jQuery Mobile
“the” answer?

Pros...and cons.

“Web” vs. “Native”.

Many native apps could be
written in HTML5.

Should they be the
interesting question.

Depends on what you
want to do...

Publishers - don't like apps?

<http://www.technologyreview.com/news/427785/why-publishers-dont-like-apps/>

Maybe they're doing it wrong...

<http://www.marco.org/2012/10/11/the-magazine>

Uncanny valley...

http://en.wikipedia.org/wiki/Uncanny_valley



<http://www.flickr.com/photos/usarak/5431814673/>

By U.S. Army Alaska



Applies to interfaces too...

<http://billhiggins.us/blog/2007/05/17/the-uncanny-valley-of-user-interface-design/>

CrossPlatformMobile.

<http://martinfowler.com/bliki/CrossPlatformMobile.html>

Pretenders: Why mobile Web apps should stop trying to act like native apps.

<http://cvi.ly/2011/06/19/pretenders-why-mobile-web-apps-should-stop-trying-to-act-like-native-apps/>

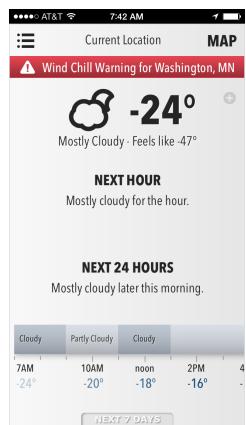
Why Mobile Web Apps Should Stop Trying to Act Like Native Apps.

<http://daringfireball.net/linked/2011/06/20/pretenders>

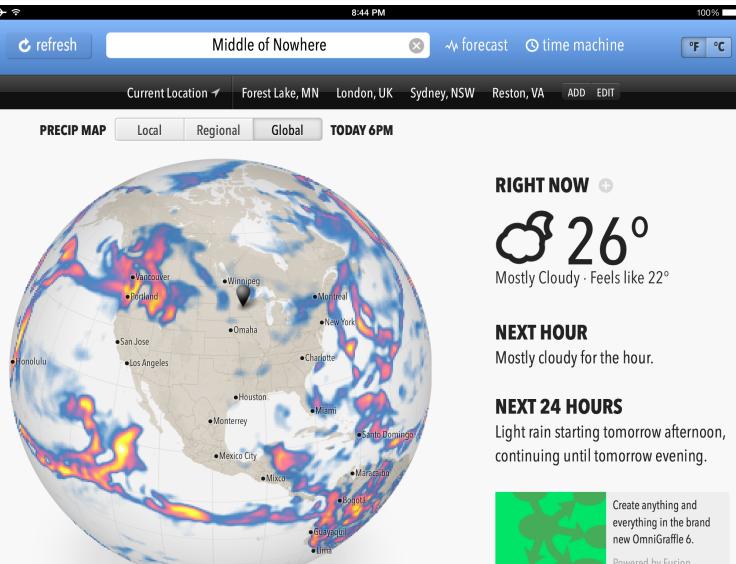
Does the coding language matter that much?

Or is it about “getting your icon on their home screen”?

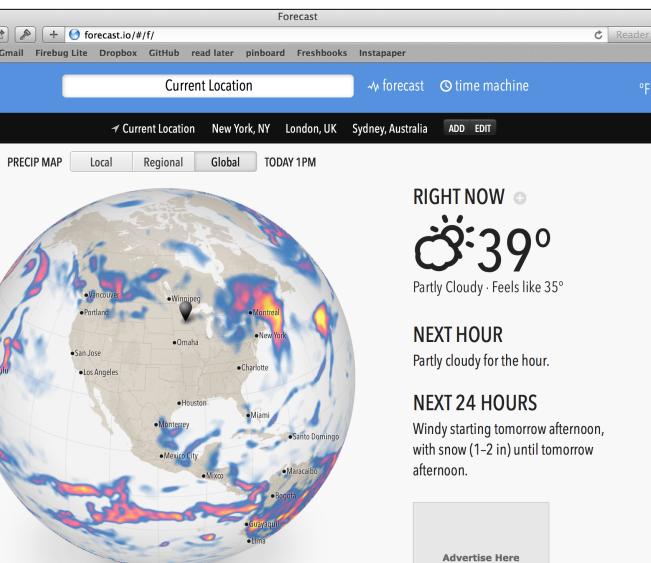
<http://ben-evans.com/benedictevans/2014/3/18/cards-code-and-wearables>



Is that an “app” or a website?



How about now?



Does it matter?

“It’s an app you install from the web”

[http://blog.forecast.io/its-not-a-web-app-
its-an-app-you-install-from-the-web/](http://blog.forecast.io/its-not-a-web-app-its-an-app-you-install-from-the-web/)

“I couldn’t find it in the app store”...

Benedict Evans
@BenedictEvans

The company most likely to kill native apps is Apple

Reply Retweet Favorite More

RETWEETS 21 FAVORITES 36

4:53 PM - 18 Mar 2014

<https://twitter.com/BenedictEvans/statuses/446071876956082176>

jQuery Mobile has changed considerably.

More CSS, less data-.

But a lot of the data- works.

May not matter much for prototyping though.

Evolving rapidly.

Expect changes.

Questions?

Thanks!

Nathaniel T. Schutta
@ntschutta