

# Jquery Mobile Overview

Author: Duy Nguyen

Release: June 2013

Revision: 1.1





## Course Objectives

At the end of the course, you will have acquired sufficient knowledge about:

- jQuery Mobile and its capabilities
- Mobile web application development using JQM



## Course Prerequisite

Trainees should have basic knowledge about following points:

- Web page designing with HTML/HTML5
- Basic level of using Javascript and CSS/CSS3



## Assessment Disciplines

- ❖ Class Participation : Required
- ❖ Assignment Completion : 100%
- ❖ Pass Score :  $\geq 70\%$

## Course Timetable

- ❖ Lecture Duration + Hands-on Labs: 3 hours
- ❖ Assignment Duration : One week



## Further References

### ❑ Document References:

- The material of this course
- Sams Teach Yourself JQM in 24 hours (pdf)

### ❑ Online References:

- <http://jquerymobile.com/demos>
- <http://learn.jquery.com/jquery-mobile/>
- <http://the-jquerymobile-tutorial.org/>



# Agenda

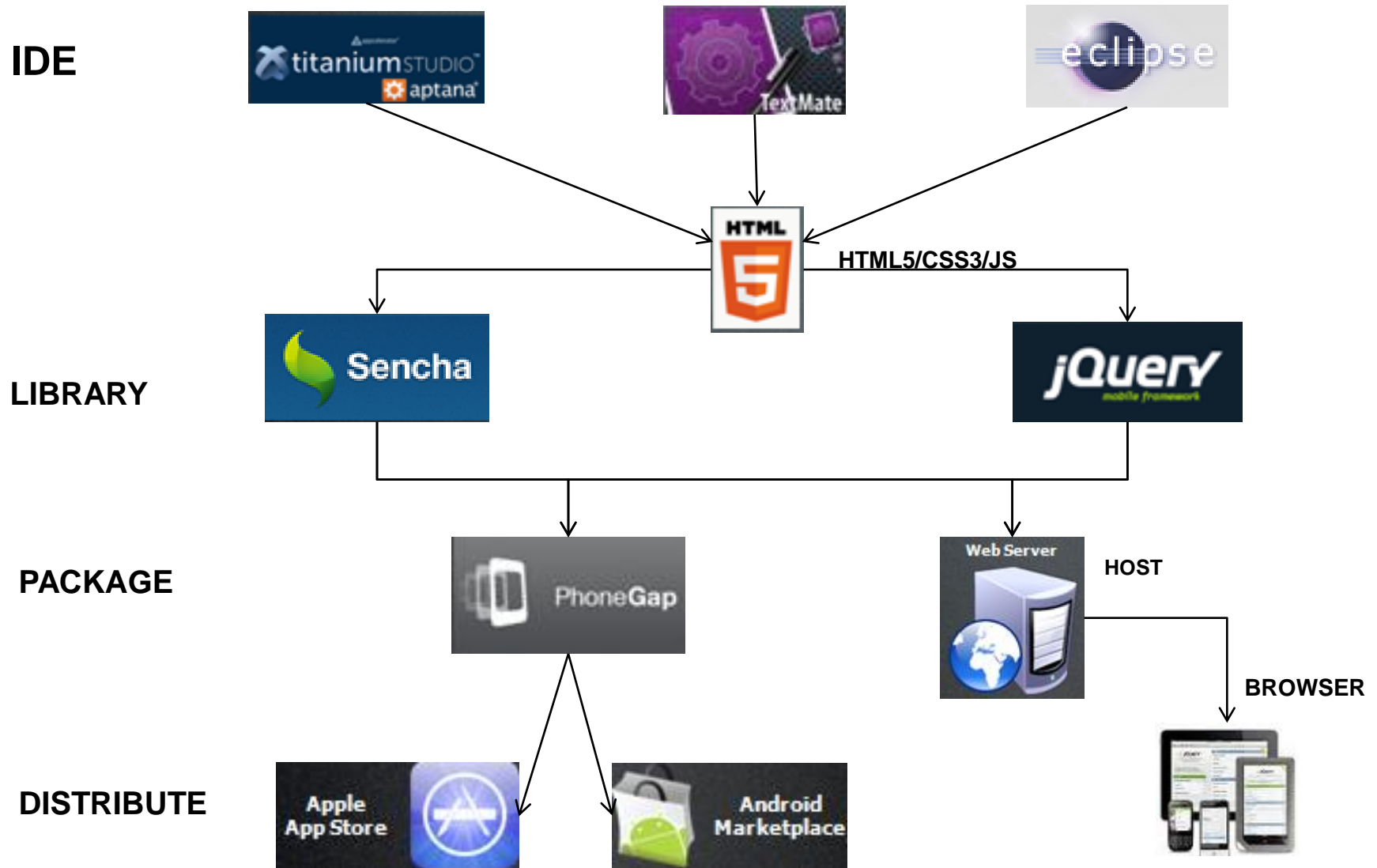
- ❑ jQuery Mobile Framework Overview
- ❑ Basic Page Structure
- ❑ Working with Page Layout
- ❑ Handling Events in jQuery Mobile
- ❑ Responsive Site Layout
- ❑ Customize Default Theme
- ❑ Comparison with Sencha Touch

# Jquery Mobile Framework Overview





## Mobile Application Framework Overview







# What is Jquery Mobile?

- ❑ A unified, HTML5-based user interface framework for all popular mobile device platform.
- ❑ Built on jQuery and jQuery-UI
  - jQuery : Javascript library with cross browser support  
→ make web developer life easier.
  - jQuery-UI : Javascript library based on jQuery and CSS providing some effects, interaction and widget.  
→ give powerful features on widget.
- ❑ Touch-optimized framework to work across all popular mobile device platform and treat mobile web browsers exactly same as desktop web browsers.



# jQuery Mobile Key Features

- Built on jQuery core
- Compatible with all major mobile platforms
- Lightweight size
- HTML5 Markup-driven configuration
- Progressive enhancement
- Powerful Ajax-powered navigation system
- New plugins enhance native controls with touch-optimized, themable controls



## Supported Platforms

- Apple iOS (3.1+)
- Android (1.6+) all devices
- Blackberry (6+)
- Windows Phone (7+)
- Palm WebOS (1.4+)
- Opera Mobile (10.1+)
- Opera Mini (5.02+)



# Jquery Mobile Basic Page Structure





## Components Required

jQuery  
Core JS

jQuery  
Mobile JS

jQuery  
Mobile CSS

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
      href="http://code.jquery.com/mobile/1.2.0/jquery.mobile-1.2.0.min.css" />
<script src="http://code.jquery.com/jquery-1.8.2.min.js"></script>
<script src="http://code.jquery.com/mobile/1.2.0/jquery.mobile-1.2.0.min.js"></script>
</head>
<body>
...
</body>
```

HTML5 DOCTYPE

Making width fit to the screen

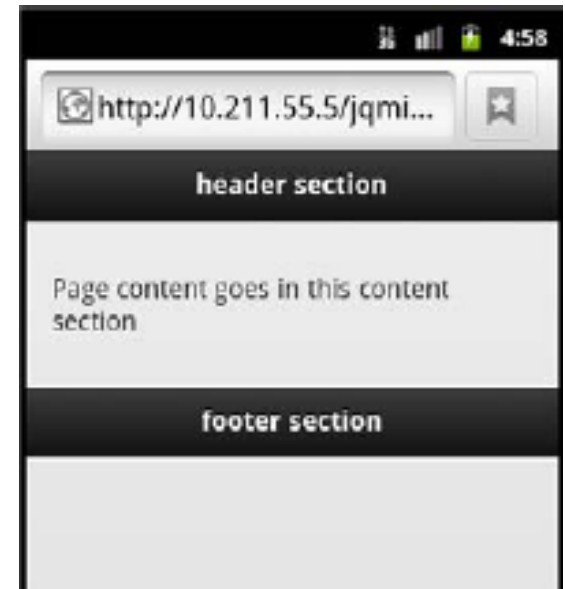
jQuery Mobile CSS & JS



## Single Page Structure

- Page is basically a div identified by data-role="page" and id.
- Each page can have there're some child <div> devided for header, content and footer part.
- Each element that represents a JQuery Mobile widget is using HTML attribute data-role.
  - ❖ *Note: "data-" attributes are wonderful new feature of HTML5 that let you add any attributes you like to an element.*

```
<body>
  <div data-role="page">
    <div data-role="header">
      <h1>header section</h1>
    </div>
    <div data-role="content">
      <p>Page content goes in this content section</p>
    </div>
    <div data-role="footer">
      <h4>footer section</h4>
    </div>
  </div>
</body>
```





## Multiple Pages Structure

- Multiple page layout is simply collection of single page layout.
- One HTML file can include multiple pages inside, or we can create multiple HTML file which contains one page, then link them together.
- Page navigation is using element a (anchor) with id name of page id as a value of href attribute

```
<body>
<div data-role="page" id="foo">
  <div data-role="header"> <h1>First</h1> </div>
  <div data-role="content"> <p>I'm first Page</p>
  <p><a href="#bar">Link to second page</a></p>
</div>

<div data-role="page" id="bar">
  <div data-role="header" data-add-back-btn="true">
    <h1>Second</h1></div>
  <div data-role="content"> <p>I'm second Page</p>
  <p><a href="#foo">Link to first page</a></p>
</div>
</body>
```





## Page Transition

- Six CSS-based transition effects applied to any object or page change event.
  - right to left (by default)
  - others: slide, slideup, slidedown, pop, fade, flip
- Add the data-transition attribute to the link

```
<a href="#secondPage" data-transition="pop">to go second Page</a>
```

## Dialog

To create dialog window

- Specify the div id which the page reference to.
- Add to the anchor a data-rel="dialog" attribute with any data-transition type.

```
<a href="#dialogPage" data-rel="dialog" data-transition="fade">  
  Open dialog  
</a>
```



# Working with Page Layout





## Grid Content

- Grids in JQM allows to create table in which columns has the same width.
- Grid container use ui-grid- \* class to apply predefined style to the grid.  
*ui-grid-a/b/c/d → create a 2/3/4/5-column grid*
- Grid content uses ui-block-\* class to specify the block item of the grid  
*ui-block-a/b/c/d/e → specify the 1<sup>st</sup>/2<sup>nd</sup>/3<sup>rd</sup>/4<sup>th</sup>/5<sup>th</sup> block of the grid.*

```
<div data-role="content">
  <div class="ui-grid-a">
    <div class="ui-block-a">
      <p>This is first in the grid</p>
    </div>
    <div class="ui-block-b">
      <p>This element is second in a grid</p>
    </div>
    <div class="ui-block-a">
      <p>By reusing ui-block-a this wraps to a new row.</p>
    </div>
    <div class="ui-block-b">
      <p>This is the final element in this grid</p>
    </div>
  </div>
</div>
```





## Collapsible Content

- To create collapsible content, add data-role="collapsible" to the containing element <div>.
- To create multiple collapsible content, add data-role="collapsible-set" to the wrapper containing element <div>.
- To expand the content by default, set data-collapsed="false" to the content element.
- Set header text for each collapsible content by adding <h1>/<h2>/<h3>/...inside the collapsible container

```
<div data-role="content">
  <div data-role="collapsible">
    <h3>I'm a single collapsible element</h3>
    <p>I'm the content inside of the single collapsible element.</p>
  </div>
  <div data-role="collapsible-set">
    <div data-role="collapsible" data-collapsed="false">
      <h3>I'm expanded on page load</h3>
      <p>I am collapsible content that is visible on page load.</p>
      <p>That's because of the data-collapsed="false" attribute</p>
    </div>
    <div data-role="collapsible">
      <h3>Expand me I have something to say</h3>
      <p>I am closed on page load, but still part of an accordion.</p>
      <div data-role="collapsible">
        <h3>Wait, are you nested?</h3>
        <p>Yes! You can even nest your collapsible content!</p>
      </div>
    </div>
  </div>
</div>
```





# Practice

# Handling Events In JQuery Mobile





# Jquery Mobile Events

- ☐ *Page Events*
- ☐ *Touch Events*
- ☐ *Orientation Events*



### Page Initialization Events

- When page is initialized, it goes through 3 states:
  - Before page creation
  - Page creation
  - Page initialization
- jQuery Mobile auto-initializes plugins based on markup conventions found in a page
- *Auto-initialization is controlled by page plugins:*
  - *Dispatches events before and after it executes*
  - *Allow manipulation on page either by pre or post initialization*
  - *Provide your own initialization behavior and prevent the auto-initializations.*
- *Page initialization will only fire once per page, opposed to the show/hide events.*



## Page Initialization Events

### ❖ **pagebeforecreate event**

- *On page initialized, before initialization occurs*
- *Used when you want to modify the content before JQM lock and write data-roles and attributes of page element to DOM.*

```
<script>
  $(document).on('pagebeforecreate', function(event) {
    $(".modify").attr('data-inset','true');
  });
</script>

.....
<div data-role="content">
  <p>The following list will be inset during the pagebeforecreate event</p>
  <ul class="modify" data-role="listview">
    <li>A</li>
    <li>B</li>
    <li>C</li>
  </ul>
</div>
```







## Page Initialization Events

### ❖ **pagecreate event**

- *On page initialized, after initialization occurs*
- *Used when you want to apply your own widget or JQM pre-defined plugin controls*

```
<script>
  $(document).on('pagecreate', 'home', function(event) {
    $(".modify").attr('data-inset', 'true').listview();
  });
</script>
.....

<div data-role="page" id="home">
  <div data-role="header"><h1>pagecreate event</h1></div>
  <div data-role="content">
    <p>The following list will be styled during the pagecreate event</p>
    <ul class="modify">
      <li>A</li>
      <li>B</li>
      <li>C</li>
    </ul>
  </div>
</div>
```



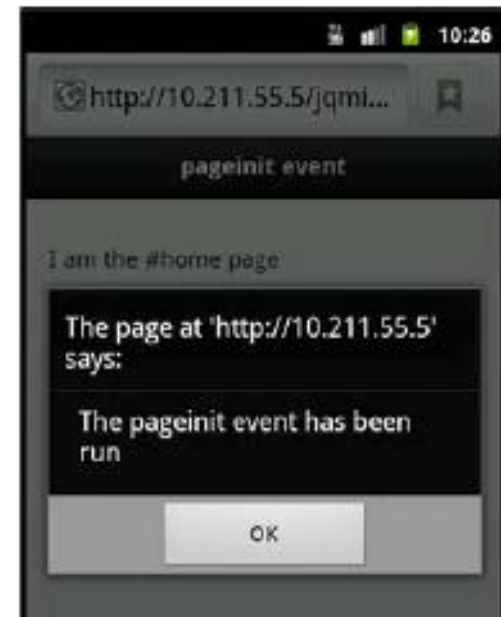


## Page Initialization Events

### ❖ pageinit event

- *This event is triggered after the DOM has been loaded and all widget/plugin have been run.*
- *This event is triggered whenever a page is loaded either directly or through Ajax call on another page.*
- *This event is triggered only once when included in the DOM*

```
<script>
  $(document).on('pageinit', 'away', function(event) {
    alert("The pageinit event has been run");
  });
</script>
.....
<div data-role="page" id="away">
  <div data-role="header"><h1>pageinit event</h1></div>
  <div data-role="content">
    <p>I am the #away page.
      The pageinit event runs on first page load.</p>
  </div>
</div>
```





# Page Show/Hide Events

- ❖ **pagebeforeshow**
  - *triggered on the page being shown, before its transition begins.*
- ❖ **pagebeforehide**
  - *triggered on the page being hidden, before its transition begins.*
- ❖ **pageshow**
  - *triggered on the page being shown, after its transition begins.*
- ❖ **pagehide**
  - *triggered on the page being hidden, after its transition begins.*



### Page Show/Hide Events

- All four events expose a reference to
  - The next page (ui.nextPage)
  - The previous page (ui.prevPage)
  - Whether the page is being shown or hidden
  - Whether the next page or previous page exist
- These events will be fired once each time the page is shown or hidden.

```
$( 'div' ).on( 'pageshow', |
    function( event, ui ) {
        alert( 'This page was just hidden: ' + ui.prevPage );
    } );
$( 'div' ).on( 'pagehide',
    function( event, ui ) {
        alert( 'This page was just shown: ' + ui.nextPage );
    } );
```



### Touch Events

- ❖ **tap**
  - *triggered after a quick tap on links, lists and other JQM plugins & widgets. Much faster and more responsive than click event.*
- ❖ **taphold**
  - *triggered after almost entire second tapping and holding*
- ❖ **swipe/swipeleft/swiperight**
  - *triggered after clicking and dragging element more than 30px distance, otherwise the event will be cancelled.*

#### Syntax:

```
$("#div_container").on('event_name', '#target_element', function(event, ui) {  
    // event handler goes here...  
});  
  
where event_name: tap/taphold/swipe/swipeleft/swiperight
```



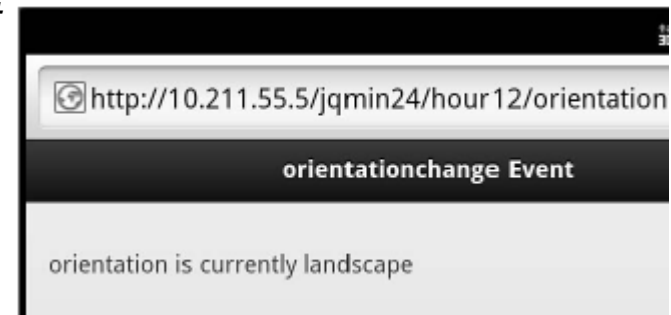
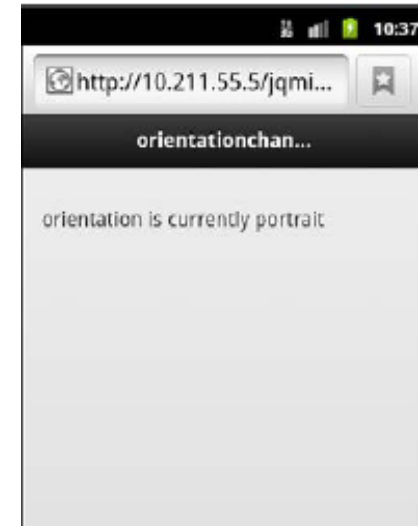
## Orientation Change Events

### ❖ **orientationchange**

- *triggered a device orientation changes by turning it vertically or horizontally.*
- *this event needs to run inside either `$(document).ready()` function or `pageinit()` event*
- *Use `orientation` property of the event object passed in call back function to get the portrait mode or landscape mode.*
- *For some device which doesnot support `window.orientation`, we should use `resize()` event instead.*

### **Syntax:**

```
$(document).on('pageinit', function() {  
    $(window).on('orientationchange', function(e) {  
        $("#mode").html('orientation is currently '+e.orientation);  
    });  
});
```





# Practice

# Responsive Site Layout







## Media Queries

- ❖ With a combination of CSS3 media queries and JQM framework, it's possible to achieve a responsive design for mobile, tablet and desktop environment.
- ❖ Media queries deliver a device-specific style sheet applied for mobile, tablet based on the screen resolution.
- ❖ JQuery Mobile add Media Queries style sheet class to the HTML element to apply the style for corresponding orientation.
- ❖ These classes are updated on load, resize and orientationchange event.
- ❖ It's also possible to use multiple media queries directly within the CSS.

```
<link
  rel="stylesheet"
  type="text/css"
  media="screen and (max-device-width: 799px)"
  href="mobile-tablet.css" />
```

```
@media all and (min-width: 800px) {
  #nav {
    width: 300px;
  }
}

@media all and (max-width: 799px) {
  #nav {
    width: 100%;
  }
}
```



## Orientation Classes

- ❖ The HTML always has a class of either “portrait” or “landscape”, depending on the orientation of browser or device.

```
.portrait {  
    /* portrait orientation changes go here! */  
}  
.landscape {  
    /* landscape orientation changes go here! */  
}
```

## Min-Max Width Breakpoint Classes

- ❖ By default we have min/max breakpoint classes with the width of 320, 480, 768, 1024
- ❖ We can make use of these classes like: “min-width-320px” or “max-width-480”, ...
- ❖ Many plugins in jQuery mobile leverage these width breakpoints.

```
.myelement {  
    float: none;  
}  
.min-width-480px .myelement {  
    float: left;  
}
```

```
label.ui-input-text {  
    display: block;  
}  
.min-width-480px label.ui-input-text {  
    display: inline-block;  
}
```



# Adding Custom Width Breakpoint

- ❖ jQuery mobile exposes the `$.mobile.addResolutionBreakpoints()` function, which helps to add custom width breakpoint by your own.

```
//add a min/max class for 1200 pixel widths  
$.mobile.addResolutionBreakpoints(1200);
```

```
//add min/max classes for 1200, and 1440 pixel widths  
$.mobile.addResolutionBreakpoints([1200, 1440]);
```

## Running Media Queries

- ❖ jQuery Mobile provides a function that allows you to test whether a particular CSS Media Query applies.
- ❖ Simple call `$.mobile.media()` and pass a media type or query. If the browser supports that type or query, and it currently applies, the function will return true. If not, it'll return false.

```
//test for screen media type  
$.mobile.media("screen");
```

```
//test a min-width media query  
$.mobile.media("screen and (min-width: 480px)");
```

```
//test for iOS retina display  
$.mobile.media("screen and (-webkit-min-device-pixel-ratio: 2)");
```



# Practice

# Customize Default Theme





# Apply Different Swatches

- ❖ Swatches are a set of defined styles that can be applied when using a theme.
- ❖ Swatches are labelled by alphabet letter. By default, only 5 swatches are included in jQuery Mobile (a-e)
- ❖ We can extend our own swatch using any alphabet from a-z by creating new CSS for new theme or override the defined CSS of pre-defined them.
- ❖ Change the default swatch used for an entire page by including an attribute of **data-theme** with the swatch value (a-z)
- ❖ When the page loads, every component except the header and footer will have that swatch color scheme applied to it.

```
<div data-role="page" data-theme="b">
  <div data-role="header"><h1>Single Swatch</h1></div>
  <div data-role="content">
    <p>Look at the button!</p>
    <a href="#" data-role="button">I am a button</a>
    <ul data-role="listview" data-inset="true">
      <li>List Item 1</li>
      <li>List Item 2</li>
      <li>List Item 3</li>
    </ul>
  </div>
</div>
```





# Customize the defined swatches

- ❖ Open the noncompressed or minified CSS file that is the part of jQuery Mobile framework.
- ❖ Change the default theme by overwrite the CSS of the base jQuery Mobile styles
  - i.e make changes on some classes like:
    - ui-bar-(a-z)
    - ui-btn-up-(a-z)
    - ui-btn-down-(a-z)....
- ❖ There're few options to create a swatch:
  - ✓ Modify an existing swatch
  - ✓ Code swatch from scratch
  - ✓ Copy one and change the swatch letter, then change the setting as you prefer

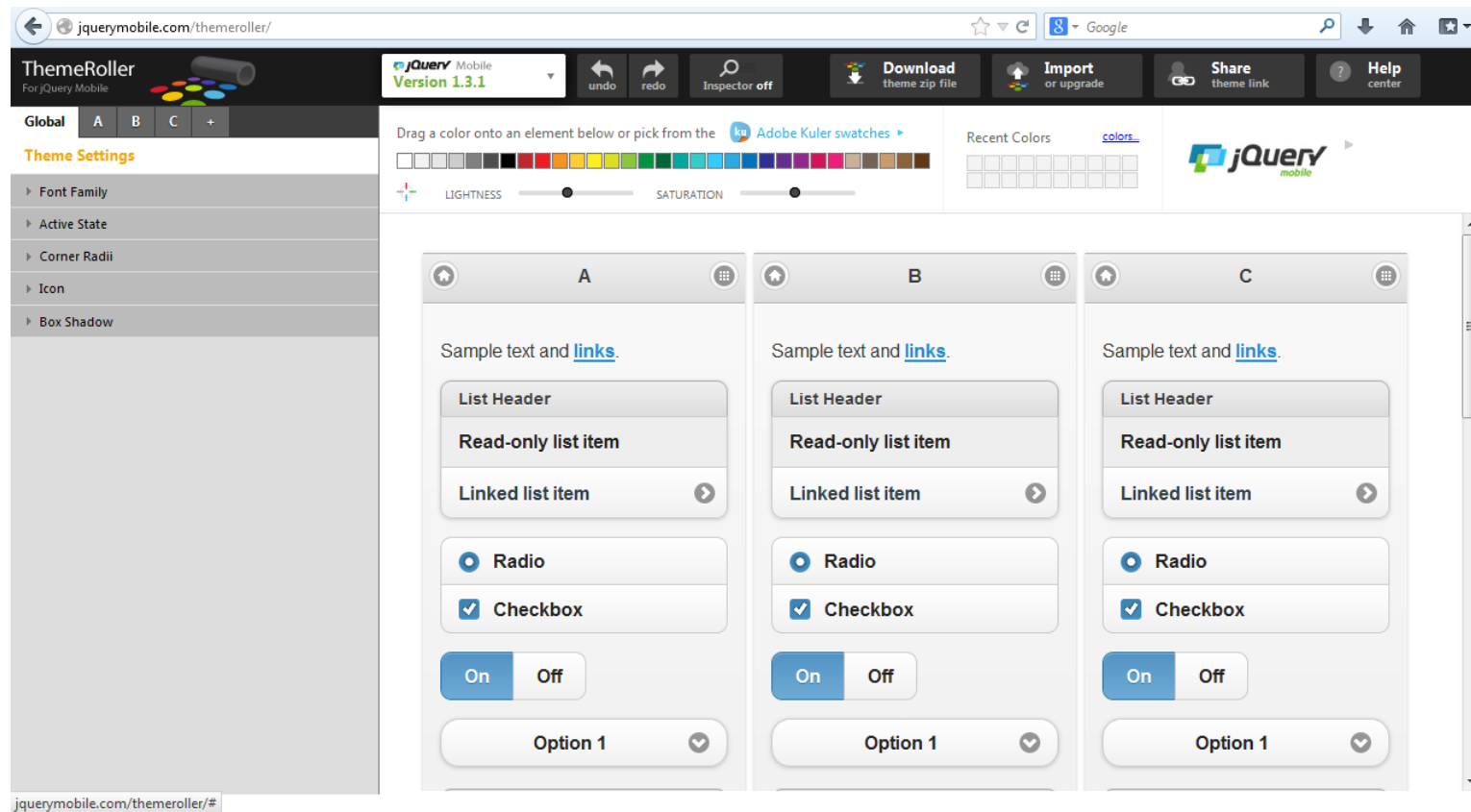


```
.ui-bar-f,  
.ui-bar-f input,  
.ui-bar-f select,  
.ui-bar-f textarea,  
.ui-bar-f button {  
font-family: Helvetica, Arial, sans-serif;  
}  
.ui-bar-f .ui-link-inherit {  
color: #fff;  
}  
.ui-bar-f .ui-link {  
color: #F7E065;  
font-weight: bold;  
}  
.ui-bar-f .ui-link: hover {  
color: #FFFF7F;  
}  
.ui-bar-f .ui-link: active {  
color: #FFFF7F;  
}
```



## Using Theme Roller

- ❖ The jQuery UI Theme Roller is just a tool which helps to create widgets with common style easily
- ❖ <http://jquerymobile/themeroller> is the current online home of the jQuery Mobile ThemeRoller





# jQuery Mobile & Sencha Touch





## Comparison between Jquery Mobile & Sencha Touch

	Jquery Mobile	Sencha Touch
UI Widgets	<ul style="list-style-type: none"><li>• UI-only library, which relies on jQuery, jQuery UI.</li><li>• Just need to hand-code HTML and CSS to markup the element without write much on JS.</li></ul>	<ul style="list-style-type: none"><li>• Follow a JS-centric approach, where you're required to write little html.</li><li>• The DOM is generated based on the most Javascript</li></ul>
Code Structure	<ul style="list-style-type: none"><li>• Not impose a coding discipline or structure</li><li>• Hard to maintain for large application</li></ul>	<ul style="list-style-type: none"><li>• Well-structured and discipline code</li><li>• Ease to maintain with MVC codebase.</li></ul>
Device Compatibility	<ul style="list-style-type: none"><li>• Support large range of platforms and compatible with all of modern browsers</li></ul>	<ul style="list-style-type: none"><li>• Target webkit browsers (Chrome &amp; Safari), meaning that will work only with Android &amp; iOS</li></ul>
Ease of use	<ul style="list-style-type: none"><li>• Lightweight and take less time to implement</li><li>• Easy to find errors</li></ul>	<ul style="list-style-type: none"><li>• Take much time to study the framework</li><li>• Need to deep down to framwork to fix bugs</li></ul>



## Pros & Cons of Jquery Mobile & Sencha Touch

	Jquery Mobile	Sencha Touch
Pros	<ul style="list-style-type: none"><li>✓ Lightweight</li><li>✓ Full control of your application</li><li>✓ Easier to find errors</li><li>✓ Support all major browsers and platforms</li></ul>	<ul style="list-style-type: none"><li>✓ MVC Codebase</li><li>✓ Blazing fast in iOS</li></ul>
Cons	<ul style="list-style-type: none"><li>✓ Not given code structure (MVC pattern)</li><li>✓ Some problems in older iOS versions</li></ul>	<ul style="list-style-type: none"><li>✓ Very hard to impossible to debug and fix errors in the framework</li><li>✓ Support for only iOS and Android. But slow on Android</li><li>✓ The need of Webkit based browsers to view apps in your browser.</li></ul>

## When To Use?

- Use **Sencha Touch** if you are used to Java/C# and only want to support iPhone and Android
- Use **jQuery Mobile** if you are web developer, used to jQuery and HTML and want to support all kinds of devices and browsers.

# THANK YOU





BUSINESS SOLUTIONS  
TECHNOLOGY  
OUTSOURCING