



# Audience

This tutorial has been prepared for the beginners to help them understand basic jQuery Mobile framework. After completing this tutorial you will find yourself at a moderate level of expertise in jQuery Mobile framework from where you can take yourself to next levels.

## Prerequisite

jQuery Mobile framework is based on Javascript programming language, so if you have basic understanding on Javascript programming then it will be a fun to learn jQuery Mobile framework.

**What is jQuery Mobile?**

jQuery Mobile is a cross platform mobile framework designed to simplify and enhance the development of mobile web applications by integrating HTML5, CSS3, jQuery and jQuery UI into one framework that is not only robust, but maintainable and organized.



jQuery Mobile is built upon the existing jQuery core. The library is more like a framework which includes animations, transition effects, and automatic CSS styles for basic HTML elements. It was developed by the *jQuery project* team in the year 2010 and written in JavaScript.



jQuery Mobile works on all popular smartphones and tablets:



# Features

- ❖ Built on jQuery core
- ❖ Compatible with all major mobile, tablet, e-reader & desktop platforms
- ❖ Lightweight size
- ❖ Responsive design
- ❖ Powerful Ajax-powered navigation system
- ❖ Touch and mouse event support



# Adding jQuery Mobile to Your Web Pages

There are two ways to add jQuery Mobile to your web site. You can:

- ☐ Link to a jQuery Mobile library stored at a CDN (recommended)
- ☐ Link to a jQuery Mobile library stored at your computer

## jQuery Mobile CDN:

Just include the following style sheet (.css) and JavaScript libraries (.js) directly into your HTML page.

**<head>**

`<meta name="viewport" content="width=device-width, initial-scale=1">`

`<link rel="stylesheet" href="https://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">`

`<script src="https://code.jquery.com/jquery-1.11.3.min.js"></script>`

`<script src="https://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></script>`

**</head>**

Link to jQuery Mobile Stored in Your Computer:

If you want to host the jQuery Mobile library yourself, you must first download it from [jquerymobile.com](http://jquerymobile.com).

Then add the following code to your page:

**<head>**

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="jquery.mobile-1.4.5.css">

<script src="jquery.js"></script>

<script src="jquery.mobile-1.4.5.js"></script>

**</head>**

# Creating our first JQuery Mobile Page

There are three different ways of laying out a page in JQuery Mobile:

- ❖ **A single page**-A single page is created in HTML document using a standard way of writing a template.
- ❖ **Multiple pages**-Multiple pages can be included in the single HTML document which load together by adding multiple divs with *data-role="page"* or by linking with *href* attribute.
- ❖ **Dialog Page**-Modal dialogs open content in an interactive overlay above the page.

# A Single Page

```
<body>
  <div data-role="page">

    <div data-role="header">
      <h1>Welcome To My Homepage</h1>
    </div>

    <div data-role="main" class="ui-content">
      <p>I Am Now A Mobile Developer!!</p>
    </div>

    <div data-role="footer">
      <h1>Footer Text</h1>
    </div>

  </div>
</body>
```



## Example explained:

- ❑ The `data-role="page"` is the page displayed in the browser
- ❑ The `data-role="header"` creates header at the top of the page.
- ❑ The `data-role="main"` is used to define the content of the page.
- ❑ The `"ui-content"` class includes padding and margin inside the page content.
- ❑ The `data-role="footer"` creates footer at the bottom of the page.
- ❑ Inside these containers, you can add any HTML elements - paragraphs, images, headings, lists, etc.



# Multiple Pages

## *Multiple pages linked internally*

```
<body>
  <div data-role="page" id="pageOne">
    <div data-role="main" class="ui-content">
      <a href="#pageTwo">Go to Page Two</a>
    </div>
  </div>

  <div data-role="page" id="pageTwo">
    <div data-role="main" class="ui-content">
      <a href="#pageOne">Go to Page One</a>
    </div>
  </div>
</body>
```

## ***Multiple pages linked externally***

```
<body>
  <div data-role="page" id="pageOne">
    <div data-role="main" class="ui-content">
      <a href="page2.php">Go to Page Two</a>
    </div>
  </div>
</body>
```

# Dialog Page

```
<body>
  <div data-role="page" id="pageone">
    <div data-role="header">
      <h1>Welcome To My Homepage</h1>
    </div>

    <div data-role="main" class="ui-content">
      <p>Welcome!</p>
      <a href="#pagetwo">Go to Dialog Page</a>
    </div>

    <div data-role="footer">
      <h1>Footer Text</h1>
    </div>
  </div>
```

```
<div data-role="page" data-dialog="true" id="pagetwo">
  <div data-role="header">
    <h1>I'm A Dialog Box!</h1>
  </div>

  <div data-role="main" class="ui-content">
    <p>The dialog box is different from a normal page, it is
displayed on top of the current page and it will not span the entire width
of the page. The dialog has also an icon of "X" in the header to close the
box.</p>
    <a href="#pageone">Go to Page One</a>
  </div>

  <div data-role="footer">
    <h1>Footer Text In Dialog</h1>
  </div>
</div>
</body>
```

# Creating a Button in jQuery Mobile

A button in jQuery Mobile can be created in three ways:

- ❖ Using the `<input>` element
- ❖ Using the `<button>` element with `class="ui-btn"`
- ❖ Using the `<a>` element with `class="ui-btn"`

Buttons in jQuery Mobile are automatically styled, making them attractive and useable on both mobile devices and desktop computers. **The `<a>` element with `class="ui-btn"` is better used to link between pages, and `<input>` or `<button>` elements for form submission.**

```
<body>
  <div data-role="main" class="ui-content">
    <input type="button" value="Input">
    <button class="ui-btn">Button</button>
    <a href="#anylink" class="ui-btn">Anchor</a>
    <div data-role="controlgroup" data-type="horizontal">
      <a href="#" class="ui-btn">Button 1</a>
      <a href="#" class="ui-btn">Button 2</a>
    </div>
    <div data-role="controlgroup" data-type="vertical">
      <a href="#" class="ui-btn">Button 1</a>
      <a href="#" class="ui-btn">Button 2</a>
    </div>
    <a href="#pagetwo" class="ui-btn ui-btn-inline">Inline Button</a>
    <a href="#" class="ui-btn" data-rel="back">Go Back</a>
  </div>
</body>
```



jQuery Mobile provides a set of icons that will make your buttons look more desirable. To add an icon to your button, use the **ui-icon class**, and position the icon with an icon position class (**ui-btn-icon-pos**):

```
<a href="#anylink" class="ui-btn ui-icon-search ui-btn-icon left">Search</a>
```

```
<a href="#anylink" class="ui-btn ui-icon-search ui-btn-icon-notext">Search</a>
```

**Note:** For other buttons, like buttons in list views and forms, you must use the data-icon attribute.

## Some available icons that jQuery Mobile provide

Icon Class	Description
ui-icon-arrow-l	Left Arrow
ui-icon-arrow-r	Right Arrow
ui-icon-search	Search
ui-icon-home	Home
ui-icon-back	Back
ui-icon-delete	Delete
ui-icon-audio	Audio
ui-icon-grid	Grid
ui-icon-info	Information



# jQuery Mobile Transition Effects

jQuery Mobile has a variety of effects for how to transition from one page to the next. The transition effect can be applied to any link or form submission by using the data-transition attribute:

**`<a href="#anylink" data-transition="slide">Slide to Page Two</a>`**

**Tip:** All the transition effects also support reverse/backward actions, e.g. if you want the page to slide from left to right, instead of right to left, use the data-direction attribute with value "reverse".

**`<a href="#anylink" data-transition="slide" data-direction="reverse">Slide</a>`**

## Some available transitions that jQuery Mobile provide

Transition	Description
fade	Fades to the next page
flip	Flips to the next page from back to front
flow	Throws the current page away and comes in with the next page
pop	Goes to the next page like a popup window
slide	Slides to the next page from right to left
slidefade	Slides from right to left and fades in the next page
slideup	Slides to the next page from bottom to top
slidedown	Slides to the next page from top to bottom
turn	Turns to the next page
none	No transition effect

# jQuery Mobile Navigation Bars

A navigation bar consists of a group of links that are aligned horizontally, typically within a header or footer. The bar is coded as an unordered list of links wrapped inside a `<div>` element that has the **`data-role="navbar"`** attribute. By default, links inside a navigation bar will automatically turn into a button (no need for **`class="ui-btn"`** or **`data-role="button"`**).

To add an icon to your navigation button, use the **`data-icon`** attribute:

```
<div data-role="header">  
  <div data-role="navbar">  
    <ul>  
      <li><a href="#anylink">Home</a></li>  
      <li><a href="#anylink">Page Two</a></li>  
      <li><a href="#anylink">Search</a></li>  
    </ul>  
  </div>  
</div>
```

# jQuery Mobile Tables

Tables in JQuery Mobile are responsive. With a simple class name, JQuery Mobile is aware of the user's available screen size and automatically resizes itself to show content that is relevant for that particular user.

Responsive tables allow us to display a large set of tabular data that will look attractive for both mobiles and desktops.

There are two types of responsive tables:

- ❑ **reflow** and
- ❑ **column toggle**.

# Reflow Tables

The reflow mode positions the table data horizontally until it reaches a minimum size, then all rows are grouped together vertically.

Create a table, add the data-role="table" and a class of "ui-responsive" on the <table> element.

# Example

```
<table data-role="table" class="ui-responsive">  
  <thead>  
    <tr>  
      <th>CustomerID</th>  
      <th>CustomerName</th>  
      <th>ContactName</th>  
      <th>Address</th>  
      <th>City</th>  
      <th>PostalCode</th>  
      <th>Country</th>  
    </tr>  
  </thead>
```



```
<tbody>
  <tr>
    <td>1</td>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Obere Str. 57</td>
    <td>Berlin</td>
    <td>12209</td>
    <td>Germany</td>
  </tr>
</tbody>
</table>
</div>
```



# Column Toggle Table

The "column toggle" mode will hide columns when there is not enough width to display the data.

To create a column toggle table, add the following to the <table> element:

```
<table data-role="table" data-mode="columnntoggle" class="ui-responsive"  
id="myTable">
```

By default, jQuery Mobile will hide columns from the right side of the table. However, you are allowed to specify which column that should be hidden or shown in a specific order. Add the **data-priority** attribute to the table's header (<th>) and specify a number between 1 (highest priority) to 6 (lowest priority):

If you do not specify a priority for a column, the column will be persistent and not available for hiding.

To change the toggle table button text, use the **data-column-btn-text** attribute:

# Example

```
<table data-role="table" data-mode="columntoggle" class="ui-responsive" data-column-btn-text="Click me to hide or show columns!" id="myTable">
```

```
<thead>
```

```
<tr>
```

```
<th data-priority="6">CustomerID</th>
```

```
<th>CustomerName</th>
```

```
<th data-priority="1">ContactName</th>
```

```
<th data-priority="2">Address</th>
```

```
<th data-priority="3">City</th>
```

```
<th data-priority="4">PostalCode</th>
```

```
<th data-priority="5">Country</th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
  <tr>
    <td>1</td>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Obere Str. 57</td>
    <td>Berlin</td>
    <td>12209</td>
    <td>Germany</td>
  </tr>
</tbody>
</table>
```

# jQuery Mobile Layout Grids

jQuery Mobile provides a set of CSS-based column layouts. However, column layouts are not generally recommended on a mobile device, due to the mobile's screen width.

But there are times you want to position smaller elements, like buttons or navigation tabs, side-by-side as if it was in a table. Then, columns are perfect. Columns in a grid are of equal width (and 100% wide in total), with no border, background, margin or padding. On small screens, it is not recommended to have too many buttons with text side-by-side on one row - as the text might get shortened.

For responsive grids, add the **ui-responsive class** to the container

There are five layout grids that can be used:

Grid Class	Columns	Column Widths	Corresponds to
ui-grid-solo	1	100%	ui-block-a
ui-grid-a	2	50%/50%	ui-block-a   b
ui-grid-b	3	33%/33%/33%	ui-block-a   b   c
ui-grid-c	4	25%/25%/25%/25%	ui-block-a   b   c   d
ui-grid-d	5	20%/20%/20%/20%/20%	ui-block-a   b   c   d   e

# Example

```
<h2>Two Column Grid Example</h2>
```

```
<div class="ui-grid-a" class="ui-responsive">
```

```
  <div class="ui-block-a"><div class="ui-bar ui-bar-a">First Block</div></div>
```

```
  <div class="ui-block-b"><div class="ui-bar ui-bar-b">Second Block</div></div>
```

```
</div>
```

```
<h2>Three Column Grid Example</h2>
```

```
<div class="ui-grid-b" class="ui-responsive">
```

```
  <div class="ui-block-a"><div class="ui-bar ui-bar-a">First Block</div></div>
```

```
  <div class="ui-block-b"><div class="ui-bar ui-bar-b">Second Block</div></div>
```

```
  <div class="ui-block-c"><div class="ui-bar ui-bar-c">Third Block</div></div>
```

```
</div>
```



# jQuery Mobile List Views

List views in jQuery Mobile are standard HTML lists; ordered (<ol>) and unordered (<ul>).

To create a list, apply the `data-role="listview"` to the <ol> or <ul> element. To make the items tappable, specify a link inside each list item (<li>). By default, links inside a list item will automatically turn into a button (no need for **`ui-class="btn"`** or **`data-role="button"`**). If you have an alphabetically list, (for example a phone book) jQuery Mobile automatically adds appropriate dividers by setting the **`data-autodividers="true"`** attribute on the <ol> or <ul> element:



# Example

<h2>Ordered List:</h2>

```
<ol data-role="listview" data-role="listview" data-autodividers="true" >
```

```
<li><a href="#">List Item</a></li>
```

```
<li><a href="#">List Item</a></li>
```

```
<li><a href="#">List Item</a></li>
```

```
</ol>
```

<h2>Unordered List:</h2>

```
<ul data-role="listview" data-role="listview" data-autodividers="true" >
```

```
<li><a href="#">List Item</a></li>
```

```
<li><a href="#">List Item</a></li>
```

```
<li><a href="#">List Item</a></li>
```

```
</ul>
```

# jQuery Mobile List Content

The default icon for each list item containing a link is "**carat-r**" (right arrow). To change this to another icon, use the **data-icon** attribute on the list item you want to modify. To add a standard 16x16px icon to your list, add an `<img>` element inside the link with a class of "**ui-li-icon**". To create a split list with a vertical divider bar, place two links inside the `<li>` element. jQuery Mobile will automatically place the second link on the right side of the list with a right arrow-icon. And the text inside the link (if any) will be shown when a user hover over the icon:

# Example

```
<h2>Split Buttons</h2>
```

```
<ul data-role="listview" data-inset="true">
```

```
<li data-icon="delete">
```

```
<a href="#">
```

```

```

```
<h2>Mr. Cauchy</h2>
```

```
<p>Whoever thought that fat girl will turn into Oprah</p>
```

```
</a>
```

```
<a href="#">Onyebuchi</a>
```

```
</li>
```

```
</ul>
```

# jQuery Mobile Filters

All elements that have more than one child element, can be filtered.

## How To Create a Search Field:

- ❖ The element you want to be filterable, must include the **data-filter="true"** attribute.
- ❖ Create an `<input>` element with a specified id and add the `data-type="search"` attribute. This will create a basic search field. Wrap the `<input>` in a form, and add the "ui-filterable" class to the `<form>` element - this will adjust the margin between the search field and the filterable element.
- ❖ Then, add the `data-input` attribute to the filterable element. Its value must match the id of the `<input>` element.

# Example

```
<form class="ui-filterable">
```

```
  <input id="myFilter" data-type="search" placeholder="Search for names..">
```

```
</form>
```

```
<ul data-role="listview" data-filter="true" data-input="#myFilter">
```

```
  <li><a href="#">Adele</a></li>
```

```
  <li><a href="#">Billy</a></li>
```

```
  <li><a href="#">Calvin</a></li>
```

```
</ul>
```

# jQuery Mobile Forms

jQuery Mobile automatically style HTML forms to make them look engaging and touch-friendly. jQuery Mobile uses CSS to style HTML form elements, making them attractive and easy to use.

In jQuery Mobile you can use the following form controls:

- ❖ Text Inputs
- ❖ Search Inputs
- ❖ Radio Buttons
- ❖ Checkboxes
- ❖ Select Menus
- ❖ Sliders
- ❖ Flip Toggle Switches

When working with jQuery Mobile forms you should know:

The <form> element must have a method and an action attribute

Each form element must have a unique "id" attribute. The id must be unique across the pages in the site. This is because jQuery Mobile's single-page navigation model allows many different "pages" to be present at the same time

Each form element must have a label. Set the **for** attribute of the label to match the id of the element.



To make labels and form elements look properly on wider screens, wrap a `<div>` element with the "ui-field-contain" class around the label/form element. jQuery Mobile will automatically handle the form submission via AJAX, and will attempt to integrate the server response into the DOM of the application.

## Example

```
<form method="post" action="process.php">  
  <div class="ui-field-contain">  
    <label for="lname" class="ui-hidden-accessible" >Last name:</label>  
    <input type="text" name="lname" id="lname" placeholder="First name">  
    <label for="bday">Date of Birth:</label>  
    <input type="date" name="bday" id="bday">  
    <label for="search">Search:</label>  
    <input type="search" name="search" id="search">  
    <label for="info">Additional Information:</label>  
    <textarea name="addinfo" id="info"></textarea>
```

```
<fieldset data-role="controlgroup" data-type="horizontal">
  <legend>Choose your gender:</legend>
  <label for="male">Male</label>
  <input type="radio" name="gender" id="male" value="male">
  <label for="female">Female</label>
  <input type="radio" name="gender" id="female" value="female">
</fieldset>
<label for="points">Points:</label>
  <input type="range" data-show-value="true" name="points" id="points"
value="50" min="0" max="100">
</div>
<div data-role="controlgroup" data-type="horizontal">
  <input type="button" value="Button">
  <input type="reset" value="Reset Button">
  <input type="submit" value="Submit Button">
</div>
</form>
```

# jQuery Mobile Themes

jQuery Mobile provides two different style themes, "a" and "b" - each with different colors for buttons, bars, content blocks, and so on.

To customize the look of your application, use the data-theme attribute, and assign the attribute with a letter:

```
<div data-role="page" data-theme="a | b">
```

For buttons with class="ui-btn", use the "ui-btn-a | b" class to style the button either gray (default) or black:

```
<a href="#" class="ui-btn ui-btn-a | b">Button</a>
```

# Themes in jQuery Mobile and their descriptions

Value	Description
a	Black text on a light gray background for page content Black text on a gray background for headers and footers Black text on a light gray background for buttons White text on a blue background for active buttons Blue text on links Light gray text (placeholder) or black text (value) on a white background for input fields
b	White text on a dark gray background for page content White text on a dark gray background for headers and footers White text on a charcoal background for buttons White text on a "cyan" blue background for active buttons "Cyan" blue text on links Gray text (placeholder) or white text (value) on a black background for input fields

# Example

```
<body>  
  <div data-role="page" data-theme="b">  
  
    <div data-role="header">  
      <h1>Welcome To My Homepage</h1>  
    </div>  
  
    <div data-role="main" class="ui-content">  
      <p>I Am Now A Mobile Developer!!</p>  
    </div>  
  
    <div data-role="footer">  
      <h1>Footer Text</h1>  
    </div>  
  
  </div>  
</body>
```



# jQuery Mobile Events

You can use any standard jQuery events in jQuery Mobile.

In addition, jQuery Mobile also offers several events that are tailor-made for mobile-browsing:

- ❖ Touch events - triggers when a user touches the screen (tap and swipe)
- ❖ Scroll events - triggers when a user scrolls up and down
- ❖ Orientation events - triggers when the device rotates vertically or horizontally
- ❖ Page events - triggers when a page is shown, hidden, created, loaded and/or unloaded



# Initializing jQuery Mobile Events

In jQuery Mobile, we use the `pagecreate` event, which occurs when the page has been created in the DOM, but before enhancement is complete to initialize events.

The second parameter ("`#pageone`") points to the id of the page to specify the event(s) for:

```
<script>
```

```
$(document).on("pagecreate", "#page_id", function(){
```

```
    // jQuery events go here...
```

```
});
```

```
</script>
```

The jQuery `on()` method is used to attach event handlers.



# jQuery Mobile Touch Events

Touch events are triggered when the user touches the screen (page).

Some JQuery Mobile touch events include:

Event	Description
jQuery Mobile Tap	The tap event is triggered when the user taps on an element.
jQuery Mobile Taphold	The taphold event is triggered when the user taps on an element and hold for one second:
jQuery Mobile Swipe	The swipe event is triggered when the user swipes over an element horizontally by more than 30px:
jQuery Mobile Swipeleft	The swipeleft event is triggered when the user swipes over an element in the left direction by more than 30px:

# Example

```
<body>  
  <div data-role="page" id="pageone">  
    <div data-role="header">  
      <h1>The tap Event</h1>  
    </div>  
  
    <div data-role="main" class="ui-content">  
      <p>Tap me away!</p>  
      <p>Tap me too!</p>  
    </div>  
  
    <div data-role="footer">  
      <h1>Footer Text</h1>  
    </div>  
  </div>  
</body>
```

```
<script>
```

```
$(document).on("pagecreate", "#pageone", function(){
```

```
    $("p").on("tap", function(){
```

```
        $(this).hide();
```

```
    });
```

```
});
```

```
</script>
```

# jQuery Mobile Scroll Events

jQuery Mobile provides two scroll events: when scrolling starts and when scrolling stops. Some JQuery Mobile scroll events include:

Event	Description
jQuery Mobile Scrollstart	The scrollstart event is triggered when the user starts to scroll the page:
jQuery Mobile Scrollstop	The scrollstop event is triggered when the user stops to scroll the page:

```
<script>
```

```
$(document).on("scrollstop", function(){
```

```
    alert("Stopped scrolling!");
```

```
});
```

```
</script>
```

# jQuery Mobile orientationchange Event

The orientationchange event is triggered when the user rotates the mobile device vertically(portrait) or horizontally(landscape). The window.orientation property returns 0 for portrait and 90 or -90 for landscape view. Because the orientationchange event is bound to the window object, we can use the window.orientation property to, for example, set different styles to distinguish between portrait and landscape views:



## Example

```
$(window).on("orientationchange",function(){  
    if(window.orientation == 0) // Portrait  
    {  
        $("p").css({"background-color":"yellow"});  
    }  
    else // Landscape  
    {  
        $("p").css({"background-color":"pink"});  
    }  
});
```

# jQuery Mobile Page Events

Events for handling pages in jQuery Mobile are divided into four categories:

- ❖ Page Initialization - Before page creation, and when the page has been created
- ❖ Page Load/Unload - When an external page is loading, unloading or encounters a failure
- ❖ Page Transition - Before and after page transitions
- ❖ Page Change - When pages are changed to or from, or encounters a failure

# jQuery Mobile Initialization Events

When a typical page in jQuery Mobile is initialized, it goes through two stages:

- ❖ Before page creation
- ❖ Page creation

Each stage has an event that can be used to insert or manipulate code before or when jQuery Mobile enhances the page.

jQuery Mobile initialization events include:

Event	Description
pagebeforecreate	Triggered when the page is about to be initialized, and before jQuery Mobile has started enhancing the page
pagecreate	Triggered when the page has been created, but before enhancement is complete

## Example

```
$(document).on("pagebeforecreate", function(event){  
    alert("pagebeforecreate event fired!");  
});  
$(document).on("pagecreate", function(event){  
    alert("pagecreate event fired!");  
});
```

# jQuery Mobile Load Events

Page load events are for external pages.

Whenever an external page is loaded into the DOM, 2 events fire. The first is `pagecontainerbeforeload`, and the second will either be `pagecontainerload` (success) or `pagecontainerloadfailed` (fail).

jQuery Mobile load events include:

Event	Description
pagecontainerbeforeload	Triggered before any page load request is made
pagecontainerload	Triggered after the page has been successfully loaded and inserted into the DOM
pagecontainerloadfailed	Triggered if the page load request fails. By default, it will show the "Error Loading Page" message



## Example

```
<body>
<div data-role="page">
  <div data-role="header">
    <h1>Header Text</h1>
  </div>

  <div data-role="main" class="ui-content">
    <a href="externalpage.php">External page</a>

    <a href="externalnotexist.php">External page that does not exist</a>
  </div>

  <div data-role="footer">
    <h1>Footer Text</h1>
  </div>
</div>
</body>
```

## Example (externalpage.php)

```
<body>
<div data-role="page">
  <div data-role="header">
    <h1>I am the external page</h1>
  </div>

  <div data-role="main" class="ui-content">
    <a href="#" data-rel="back">Back</a>

  </div>

  <div data-role="footer">
    <h1>Footer Text</h1>
  </div>
</div>
</body>
```

# Example

```
<script>
```

```
$(document).on("pagecontainerload",function(event, data){  
    alert("pageload event fired!\nURL: " + data.url);  
});
```

```
$(document).on("pagecontainerloadfailed", function(event, data){  
    alert("Sorry, requested page does not exist.");  
});
```

```
</script>
```

# jQuery Mobile Transition Events

We can also use events for when we transition from one page to the next. Page transitions involve two pages: a "from" page and a "to" page - these transitions animate the change from the current active page (fromPage) to a new page (toPage).

jQuery Mobile transition events include:

Event	Description
pagebeforeshow	Triggered on the " <b>to</b> " page, before the transition animation starts
pageshow	Triggered on the " <b>to</b> " page, after the transition animation completes
pagebeforehide	Triggered on the " <b>from</b> " page, before the transition animation starts
pagehide	Triggered on the " <b>from</b> " page, after the transition animation completes

## Example

```
$(document).on("pagebeforeshow", "#pagetwo", function(){ //
```

When entering pagetwo

```
    alert("pagetwo is about to be shown");
```

```
});
```

```
$(document).on("pageshow", "#pagetwo", function(){ // When  
entering pagetwo
```

```
    alert("pagetwo is now shown");
```

```
});
```

```
$(document).on("pagebeforehide", "#pagetwo", function(){ //
```

When leaving pagetwo

```
    alert("pagetwo is about to be hidden");
```

```
});
```

```
$(document).on("pagehide", "#pagetwo", function(){ // When
```

leaving pagetwo

```
    alert("pagetwo is now hidden");
```

```
});
```



# Questions