

jQuery Mobile

Getting Started with jQuery

Lesson 1, Activity 2: Getting Started with jQuery Mobile

A Basic Page Template

To start using jQuery Mobile, you'll need to include three files in your page:

1. The jQuery JavaScript file.
2. The jQuery Mobile JavaScript file.
3. The jQuery Mobile CSS file.

Optionally, you might include a jQuery Mobile theme CSS file; more on this later.

You can either link to files you download from the [jQuery Mobile website](#) or link to files hosted on various Content Delivery Networks, or CDNs. To create a standalone jQuery web application, say with PhoneGap, you would need to download the files to your local site, as would also be true if you were hosting a site to be accessed by users without direct access to the internet. In other cases, using the CDN-hosted files makes the most sense: a network of high-speed servers delivers files very quickly and efficiently, and this strategy makes it easier for you to update to newer jQuery or jQuery Mobile file versions in the future.

Whichever strategy you choose, hosting the files on your own web server or linking to CDN-hosted files, be sure to choose the minified versions of the files for production use. The minification (the removal of spaces and line breaks) makes these files smaller and, thus, reduces download time.

Here is an example of the code that would live in the head section of your pages, showing links to CDN-hosted files:

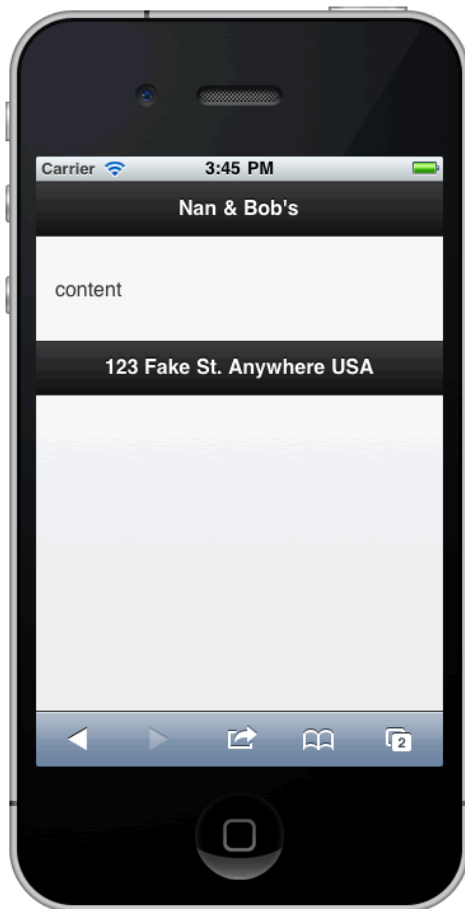
```
<link rel="stylesheet" href="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.css" />
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>
<script src="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.js"></script>
```

A Basic Page Template

You could, if you wished, link the JavaScript and CSS files shown above in any web page and start making use of jQuery Mobile's functionality. Most of the time, however, you will want to structure the markup of your pages to conform to jQuery Mobile's conventions

Let's look at a first demonstration of some of these concepts. Nan & Bob's is a (fictional) small bookstore and cafe, located right in the middle of downtown. We have developed a mobile website for them - using jQuery Mobile, of course - that we'll be extending and enhancing throughout this course. The site will give users information about their store, their books, and their cafe.

Open the file [GettingStartedjQuery/Demos/basicpagetemplate/index.html](#) in a mobile browser and also in a file editor to check out the code. Not much there yet - the page looks something like this when viewed on a mobile device:



Let's review the code:

Code Sample:

GettingStartedjQuery/Demos/basicpagetemplate/index.html

```

<!DOCTYPE html> <!-- HTML5 doctype -->
<html>
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" /> <!-- viewport tag - fit onto mobile screens -->
    <title>Nan & Bob's</title>
    <link rel="stylesheet" href="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.css" /> <!-- jQuery Mobile CSS from CDN -->
    <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script> <!-- jQuery JavaScript from CDN -->
    <script src="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.js"></script> <!-- jQuery Mobile JavaScript from CDN -->
  </head>
  <body>
    <div data-role="page"> <!-- the basic page unit -->
      <div data-role="header"> <!-- the page header for this page -->
        <h2>Nan & Bob's</h2>
      </div>
      <div data-role="content"> <!-- the content for this page -->
        <p>content</p>
      </div>
      <div data-role="footer"> <!-- the footer for this page -->
        <h3>
          123 Fake St. Anywhere USA
        </h3>
      </div>
    </div>
  </body>
</html>

```

The doctype at the top of the page identifies this page as an HTML5 page. As discussed above, we link to (and thus get to use all the power of) jQuery and jQuery Mobile JavaScript and CSS files in the head of the document.

In the body of the page, you'll note that the content of the page is marked up by a div with the data-role="page" attribute. jQuery Mobile makes extensive

use of HTML5 attributes, with `data-role` among the most important. This simple example displays only one page; we'll see soon that we can offer our users multiple pages as multiple html files or within one single file.

Within the `page div`, we see `data-roles` for `header`, `content`, and `footer` - which are, unsurprisingly, the header, main content, and footer of this particular page.

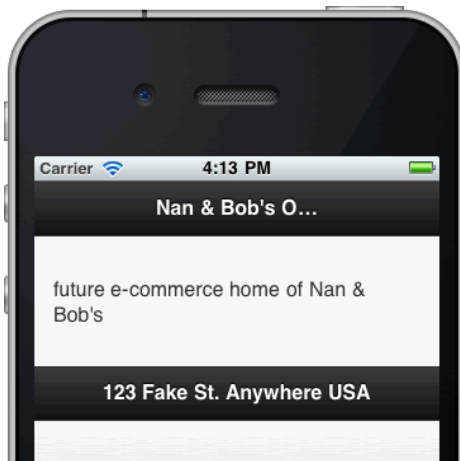
Let's now get you started writing some code!

Lesson 1, Activity 3: Nan & Bob's Online

Duration: 10 to 15 minutes.

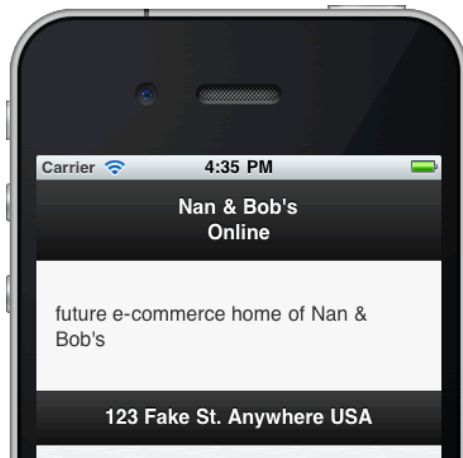
Nan & Bob want to boost their online presence with a new, also mobile-optimized, site to sell books online. You'll build a very basic shell - a first page - for that site now.

1. Open [GettingStartedjQuery/Exercises/nanandbobonline.html](#) in a file editor.
2. Add links to CDN-hosted jQuery Mobile CSS, jQuery JavaScript, and jQuery Mobile JavaScript files in the head of the file.
3. In the body of the document, add markup for the page and for the header, main content, and footer.
4. Test the file by viewing in a mobile browser; the results should look like this:



Challenge

You'll note that the title of the page ("Nan & Bob's Online") gets cut off - fix that by adding a CSS rule for `white-space: normal` for `h2` tags. This will overwrite the jQuery Mobile CSS rule `white-space: nowrap`. You'll need to use `!important` for the rule. It should look like this:



Solution:

[GettingStartedjQuery/Solutions/nanandbobonline.html](#)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>Nan & Bob's</title>
  <link rel="stylesheet" href="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.css" />
  <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>
  <script src="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.js"></script>
</head>
<body>
  <div data-role="page">
    <div data-role="header">
      <h2>Nan & Bob's Online</h2>
```

```

</div>
<div data-role="content">
  <p>future e-commerce home of Nan & Bob's</p>
</div>
<div data-role="footer">
  <h3>
    123 Fake St. Anywhere USA
  </h3>
</div>
</div>
</body>
</html>

```

We link to the various CSS and JavaScript files from the `code.jquery.com` and `ajax.googleapis.com` CDNs. We add markup for the canonical jQuery Mobile page structure: a `div` with `data-role="page"` and `divs` for the header, main content, and footer.

Challenge Solution:

[GettingStartedjQuery/Solutions/nanandbobonlineadvanced.html](#)

```

---- C O D E   O M I T T E D ----

  <style type="text/css">
    h2 {
      white-space: normal !important;
    }
  </style>
---- C O D E   O M I T T E D ----

```

The only difference in the challenge solution is the addition of a `style` rule in the head of the document, assigning `white-space: normal`. This allows the displayed title ("Nan & Bob's Online") to wrap and display without cutting off.

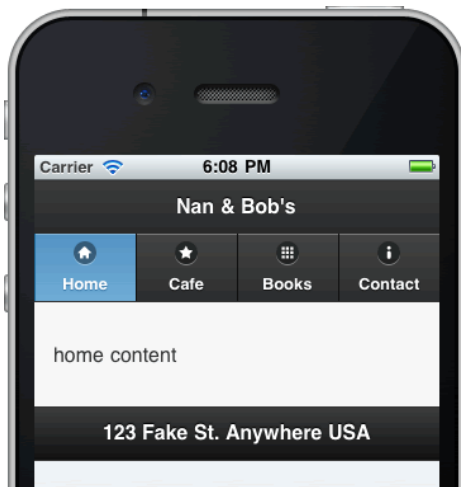
Lesson 1, Activity 5: **Linking between Pages**

The concept of a "page" in jQuery Mobile can be a slippery one. As users of a site, we know what pages are: we tap on a button or link and the stuff displayed on our phone changes - a new title and different content makes it clear that we have left one page for another page.

In traditional web development, this process of linking between pages would usually mean linking between two files: index.html might include a link (an a tag) to about.html. The user clicks the link for about.html, which sends a request from his/her browser to the server, which sends back a response with the contents of about.html.

With jQuery Mobile, "pages" might be multiple divs - each with the data-role="page" attribute - in the same file. Those pages might also be separate files. And any link on our jQuery Mobile pages could, of course, link to an external site like google.com.

Let's look at an example. Open up [GettingStartedjQuery/Demos/nanandbobs/index.html](#) in a mobile browser and in a file editor to review the code. The first page looks as follows:



Code Sample:

[GettingStartedjQuery/Demos/nanandbobs/index.html](#)

```
---- CODE OMITTED ----

<div data-role="page" id="home">
  <div data-role="header">
    <h2>Nan & Bob's</h2>
    <div data-role="navbar" data-iconpos="top">
      <ul>
        <li>
          <a href="#home" data-icon="home" class="ui-btn-active ui-state-persist">
            Home
          </a>
        </li>
        <li>
          <a href="#cafe" data-icon="star">
            Cafe
          </a>
        </li>
        <li>
          <a href="#books" data-icon="grid">
            Books
          </a>
        </li>
        <li>
          <a href="#contact" data-icon="info">
            Contact
          </a>
        </li>
      </ul>
    </div>
  </div>
  <div data-role="content">
    <p>home content</p>
  </div>
  <div data-role="footer">
    <h3>
```

```

        123 Fake St. Anywhere USA
    </h3>
</div>
</div>

<div data-role="page" id="cafe">
    <div data-role="header">
        <h2>N&B: Cafe</h2>
        <div data-role="navbar" data-iconpos="top">
            <ul>
                <li>
                    <a href="#home" data-icon="home">
                        Home
                    </a>
                </li>
                <li>
                    <a href="#cafe" data-icon="star" class="ui-btn-active ui-state-persist">
                        Cafe
                    </a>
                </li>
                <li>
                    <a href="#books" data-icon="grid">
                        Books
                    </a>
                </li>
                <li>
                    <a href="#contact" data-icon="info">
                        Contact
                    </a>
                </li>
            </ul>
        </div>
    </div>
    <div data-role="content">
        <p>cafe content</p>
    </div>
    <div data-role="footer">
        <h3>
            123 Fake St. Anywhere USA
        </h3>
    </div>
</div>

<div data-role="page" id="books">
    <div data-role="header">
        <h2>N&B: Books</h2>
        <div data-role="navbar" data-iconpos="top">
            <ul>
                <li>
                    <a href="#home" data-icon="home">
                        Home
                    </a>
                </li>
                <li>
                    <a href="#cafe" data-icon="star">
                        Cafe
                    </a>
                </li>
                <li>
                    <a href="#books" data-icon="grid" class="ui-btn-active ui-state-persist">
                        Books
                    </a>
                </li>
                <li>
                    <a href="#contact" data-icon="info">
                        Contact
                    </a>
                </li>
            </ul>
        </div>
    </div>
    <div data-role="content">
        <p>books content</p>
    </div>
    <div data-role="footer">
        <h3>
            123 Fake St. Anywhere USA
        </h3>
    </div>
</div>

```



```

<div data-role="page" id="contact">
  <div data-role="header">
    <h2>N&B: Contact</h2>
    <div data-role="navbar" data-iconpos="top">
      <ul>
        <li>
          <a href="#home" data-icon="home">
            Home
          </a>
        </li>
        <li>
          <a href="#cafe" data-icon="star">
            Cafe
          </a>
        </li>
        <li>
          <a href="#books" data-icon="grid">
            Books
          </a>
        </li>
        <li>
          <a href="#contact" data-icon="info" class="ui-btn-active ui-state-persist">
            Contact
          </a>
        </li>
      </ul>
    </div>
  </div>
  <div data-role="content">
    <p>contact content</p>
  </div>
  <div data-role="footer">
    <h3>
      123 Fake St. Anywhere USA
    </h3>
  </div>
</div>
---- C O D E   O M I T T E D ----

```

The four pages - "Home", "Cafe", "Books", and "Contact" - correspond to the four divs, each with `data-role="page"`, in the file `index.html`. Each page div has its own header, content, and footer. We've used jQuery Mobile's `navbar` UI widget to display the navigation links, and chosen simple icons to represent each of the four pages. We'll get more into more detail about this (and other) UI widgets, as well as custom icons, down the road.

Note the correspondence between the `id` value of each page div and the links:

1. `` links to the Home page, which has `id "home"`.
2. `` links to the Cafe page, which has `id "cafe"`.
3. `` links to the Books page, which has `id "books"`.
4. `` links to the Contact page, which has `id "contact"`.

Note that the appropriate title (from the `data-role="header"` div) shows for each page and that all other content - the navbar items, main content, etc - displays for each page. There is just one file (`index.html`) here, but the multiple divs act just like four separate pages as far as the user is concerned.

We use the `data-iconpos` attribute to position the icons, in this case to "top", above the text. Other valid values are "bottom", "right", or "left". We apply a class of `"ui-btn-active ui-state-persist"` to show the relevant icon as highlighted for each given page.

For devices that support it, jQuery Mobile links between pages via Ajax, without an extra (and more costly, in terms of speed) http request. The beauty of the framework is that you don't need to worry about the low-level stuff: you link between pages as you normally would, and jQuery Mobile automatically handles page requests in a single-page model, using Ajax when possible. If Ajax isn't possible, or if you specify that you don't want Ajax, a normal http request is used instead.

Some links won't be loaded with Ajax but instead will cause a full page refresh (see table below):

Links without Ajax

Description	Example
Links that point to other domains	<code>google.com</code>
Links with <code>rel="external"</code>	<code>somepage.html</code>
Links with <code>data-ajax="false"</code>	<code>aboutus.htm</code>
Links with <code>target</code> attribute	<code>books.php</code>

Page Transitions

Depending upon the device you are using, you might have noticed a slight fade-in/fade-out effect when transitioning between pages in the last example. By default, jQuery Mobile applies a "fade" transition between Ajax-loaded pages. Pages that aren't loaded via Ajax aren't loaded with an animated transition. The framework offers a variety of page transitions from which to choose:

jQuery Mobile Transitions

Transition	Description
fade	Cross-fade animation; the default transition
pop	New page grows from small to big
flip	A rotation animation, spinning on center vertical axis
turn	A rotation animation, like a page turn
flow	Current page shrinks off to left; new page slides in from right
slidefade	New page slides and fades in from right
slide	New page slides in from right
slideup	New page slides up from bottom
slidedown	New page slides down from top

You can override the default (fade) transition for all transitions in the given file by adding a global configuration directive - add the following to the head of the file, after linking the jQuery JavaScript file but before linking the jQuery Mobile file:

```
<script>
$(document).bind("mobileinit", function() {
    $.mobile.defaultPageTransition = 'flow';
});
</script>
```

You can also specify a particular transition for a given link with the data-transition attribute:

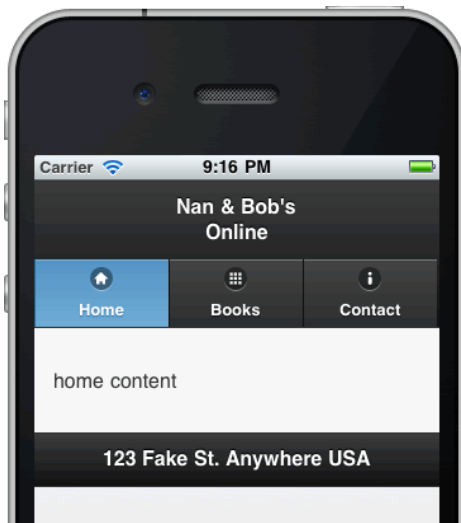
```
<a href="index.html" data-transition="flip">I will flip</a>
```

Let's work on these concepts by building out some pages for the Nan & Bob's Online site.

Lesson 1, Activity 6: More Pages & Transitions for Nan & Bob's Online

Duration: 20 to 30 minutes.

In this exercise, you will add a top navbar to the Nan & Bob's Online site, build new pages for "Books" and "Contact", and set specific page transitions. The solution should look something like this:



1. Open [GettingStartedjQuery/Exercises/transitions.html](#) in a file editor.
2. Add a div with data-role="navbar" to the header of the page div in the exercise file.
3. Add links within the navbar to #home, #books, and #contact.
4. From the [jQuery Mobile docs](#), choose appropriate icons for each link; use the data-icon attribute on the a tag to specify the icon.
5. Copy the existing page div and paste two new copies; change one pasted copy to become the "Books" page and the other to become the "Contact" page. Be sure to set the id of each page div appropriately.
6. Set the transition from "Books" to "Home" and from "Contact" to "Home" to be the slideup transition.
7. Use \$.mobile.defaultPageTransition in the head of the document to set the default page transition to flip.
8. Test your work by viewing the pages in a mobile browser.

Solution:

[GettingStartedjQuery/Solutions/transitions.html](#)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>Nan & Bob's</title>
  <link rel="stylesheet" href="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.css" />
  <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>
  <script>
    //set default transition
    $(document).bind("mobileinit", function(){
      $.mobile.defaultPageTransition = 'flip';
    });
  </script>
  <script src="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.js"></script>
  <style type="text/css">
    h2 {
      white-space: normal !important;
    }
  </style>
</head>
<body>
  <div data-role="page" id="home">
    <div data-role="header">
      <h2>Nan & Bob's Online</h2>
      <div data-role="navbar" data-iconpos="top">
        <ul>
          <li>
            <a href="#home" data-icon="home" class="ui-btn-active ui-state-persist">
              Home
            </a>
          </li>
          <li>
```

```

    <a href="#books" data-icon="grid">
      Books
    </a>
  </li>
</li>
<li>
  <a href="#contact" data-icon="info">
    Contact
  </a>
</li>
</ul>
</div>
</div>
<div data-role="content">
  <p>home content</p>
</div>
<div data-role="footer">
  <h3>
    123 Fake St. Anywhere USA
  </h3>
</div>
</div>

<div data-role="page" id="books">
  <div data-role="header">
    <h2>N&B: Books</h2>
    <div data-role="navbar" data-iconpos="top">
      <ul>
        <li>
          <a href="#home" data-icon="home" data-transition="slideup">
            Home
          </a>
        </li>
        <li>
          <a href="#books" data-icon="grid" class="ui-btn-active ui-state-persist">
            Books
          </a>
        </li>
        <li>
          <a href="#contact" data-icon="info">
            Contact
          </a>
        </li>
      </ul>
    </div>
  </div>
  <div data-role="content">
    <p>books content</p>
  </div>
  <div data-role="footer">
    <h3>
      123 Fake St. Anywhere USA
    </h3>
  </div>
</div>

<div data-role="page" id="contact">
  <div data-role="header">
    <h2>N&B: Contact</h2>
    <div data-role="navbar" data-iconpos="top">
      <ul>
        <li>
          <a href="#home" data-icon="home" data-transition="slideup">
            Home
          </a>
        </li>
        <li>
          <a href="#books" data-icon="grid">
            Books
          </a>
        </li>
        <li>
          <a href="#contact" data-icon="info" class="ui-btn-active ui-state-persist">
            Contact
          </a>
        </li>
      </ul>
    </div>
  </div>
  <div data-role="content">
    <p>contact content</p>
  </div>

```

```
</div>
<div data-role="footer">
  <h3>
    123 Fake St. Anywhere USA
  </h3>
</div>
</div>
</body>
</html>
```

We use `$.mobile.defaultPageTransition = 'flip'`; to set the default page transition to flip.

We add a navbar to the `#home` div with links to `#home`, `#books`, and `#contact`.

We add two new divs with `data-role="page"`, one each for the `#books` and `#contact` pages.

We set, using `data-transition="slideup"`, a custom page transition between the `#books` and `#home` pages and between the `#contact` and `#home` pages

Lesson 1, Activity 8: **Buttons**

Simple inline text links can be a challenge for users on a mobile device. Buttons, easier to see and easier to tap, work better for most links. jQuery Mobile makes it easy to present a-tag links as buttons in a variety of styles.

To present a link as a button, simply add the `data-role="button"` attribute to the a tag:

```
<a href="#" data-role="button">Button</a>
```

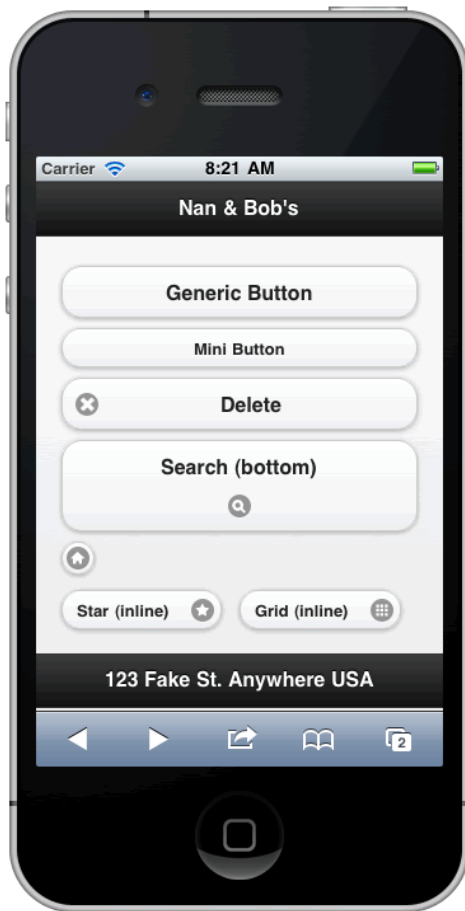
You can style buttons as `mini`, presenting them slightly smaller, with attribute `data-mini="true"`.

Adding and positioning icons is quite easy with the `data-icon` and `data-iconpos` attributes. You can create and use your own custom icons; jQuery Mobile offers a set of stock icons:

jQuery Mobile Button Icons

Icon	Code	
Left arrow	<code>data-icon="arrow-l"</code>	◀
Right arrow	<code>data-icon="arrow-r"</code>	▶
Up arrow	<code>data-icon="arrow-u"</code>	▲
Down arrow	<code>data-icon="arrow-d"</code>	▼
Delete	<code>data-icon="delete"</code>	✕
Plus	<code>data-icon="plus"</code>	⊕
Minus	<code>data-icon="minus"</code>	⊖
Check	<code>data-icon="check"</code>	✓
Gear	<code>data-icon="gear"</code>	⚙
Refresh	<code>data-icon="refresh"</code>	↺
Forward	<code>data-icon="forward"</code>	⏭
Back	<code>data-icon="back"</code>	↶
Grid	<code>data-icon="grid"</code>	⌘
Star	<code>data-icon="star"</code>	★
Alert	<code>data-icon="alert"</code>	⚠
Info	<code>data-icon="info"</code>	ℹ
Home	<code>data-icon="home"</code>	🏠
Search	<code>data-icon="search"</code>	🔍

Buttons can display no text (`data-iconpos="notext"`) and display inline. Let's look at some examples of these features:



Code Sample:

GettingStartedjQuery/Demos/buttons/index.html

```

---- CODE OMITTED ----

<a href="#" data-role="button">Generic Button</a>
<a href="#" data-role="button" data-mini="true">Mini Button</a>
<a href="#" data-role="button" data-icon="delete">Delete</a>
<a href="#" data-role="button" data-icon="search" data-iconpos="bottom">Search (bottom)</a>
<a href="#" data-role="button" data-icon="home" data-iconpos="notext">Home (no text)</a>
<a href="#" data-role="button" data-icon="star" data-iconpos="right" data-mini="true" data-inline="true">Star (inline)</a>
<a href="#" data-role="button" data-icon="grid" data-iconpos="right" data-mini="true" data-inline="true">Grid (inline)</a>

---- CODE OMITTED ----

```

1. The "Generic Button" is simply a link (an `a` tag) with the `data-role="button"` attribute.
2. The "Mini Button" has attribute `data-mini="true"`, which causes the button to display with less top and bottom padding.
3. The "Delete" button's `data-icon="delete"` attribute adds the stock jQuery Mobile "delete" icon, positioned by default at left.
4. The "Search (bottom)" button presents the jQuery Mobile "search" icon; `data-iconpos="bottom"` positions the icon at bottom.
5. The attribute `data-iconpos="notext"` on the next button hides all label text, showing just a stock "home" icon.
6. The two icons at bottom make use of the `data-inline="true"` attribute, preventing them from filling the horizontal width of the page; both are also "mini".

Grouped Buttons

Containing a set of buttons in a `div` element with the attribute `data-role="controlgroup"` groups the contained buttons visually, either stacked vertically (the default) or horizontally (with the additional attribute `data-type="horizontal"` on the container `div`):



Code Sample:

GettingStartedjQuery/Demos/buttons/grouped.html

```

---- CODE OMITTED ----

<div data-role="controlgroup">
  <a href="#" data-role="button">Yes</a>
  <a href="#" data-role="button">No</a>
  <a href="#" data-role="button">Maybe</a>
</div>
<div data-role="controlgroup" data-type="horizontal">
  <a href="#" data-role="button">Yes</a>
  <a href="#" data-role="button">No</a>
  <a href="#" data-role="button">Maybe</a>
</div>
<div data-role="controlgroup" data-type="horizontal">
  <a href="#" data-role="button" data-icon="arrow-l">Left</a>
  <a href="#" data-role="button" data-icon="arrow-r" data-iconpos="right">Right</a>
</div>

---- CODE OMITTED ----

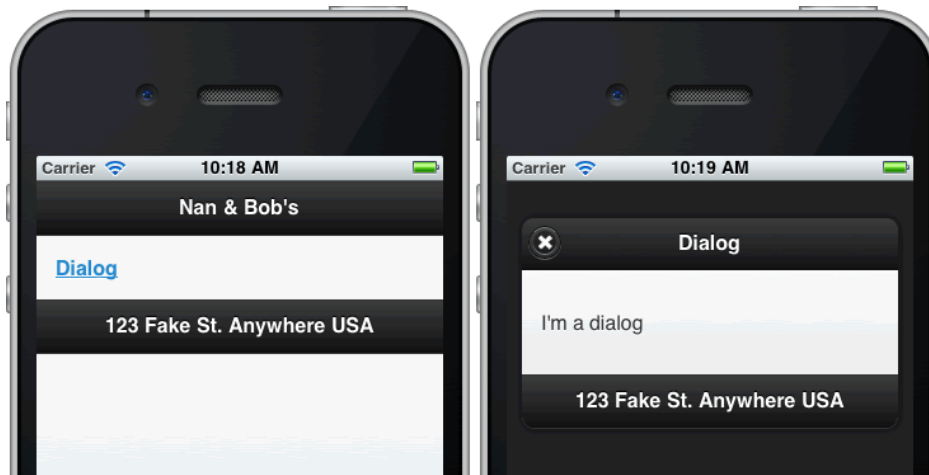
```

The first set of buttons stacks vertically because of the `<div data-role="controlgroup">` wrapping the three buttons. The `data-type="horizontal"` on the second set of buttons aligns the buttons horizontally. The third set of buttons ("Left" and "Right") include appropriately labeled and positioned icons.

Dialogs

Any page in jQuery Mobile can be presented as a modal dialog - content popped up in front of the current page, like a dialog box - by adding `data-rel="dialog"` to the link to that page. When the "dialog" attribute is applied, the framework adds styles to the linked-to page (rounded corners, margins around the page, a dark background) to make it look like it is above the current page. We can use dialogs to present modal information (i.e., "confirm delete"), offer associated content, present sign-up forms, etc.

The left screenshot shows the calling page; the right screenshot shows the popped-up dialog the results from tapping the "Dialog" link:



Code Sample:

GettingStartedjQuery/Demos/dialogs/index.html

```

---- CODE OMITTED ----

<div data-role="page">
  <div data-role="header">
    <h2>Nan & Bob's</h2>
  </div>
  <div data-role="content">
    <a href="#dialog" data-rel="dialog">Dialog</a>
  </div>
  <div data-role="footer">
    <h3>
      123 Fake St. Anywhere USA
    </h3>
  </div>
</div>

<div data-role="page" id="dialog">
  <div data-role="header">
    <h2>Dialog</h2>
  </div>
  <div data-role="content">
    <p>I'm a dialog</p>
  </div>
  <div data-role="footer">
    <h3>
      123 Fake St. Anywhere USA
    </h3>
  </div>
</div>
---- CODE OMITTED ----

```

The `data-rel="dialog"` attribute on the link presents the linked-to content - the `div` with `id "dialog"` - as a dialog. The popped up content ("I'm a dialog") thus opens with a "close" button at upper left and drop shadowing to make it appear above the existing page content.

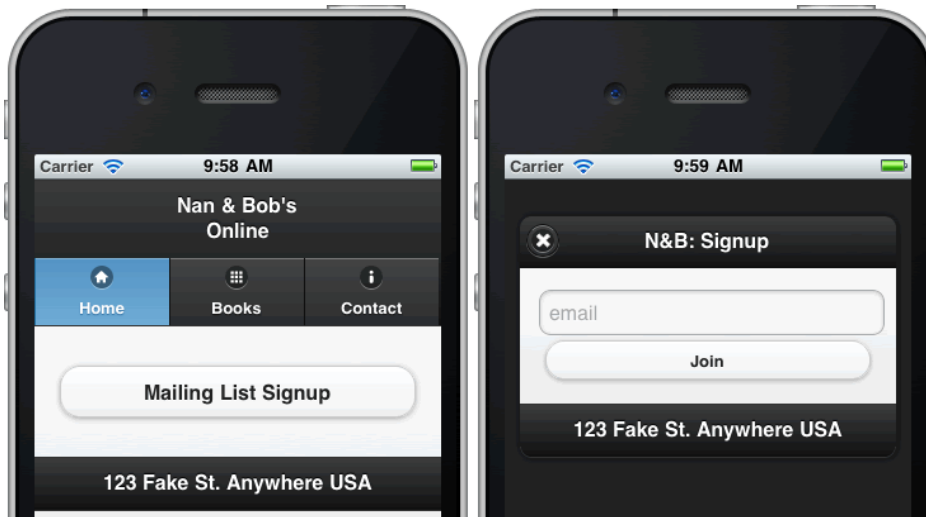
In the example above, it is the link to the content that determines that this is a dialog rather than a page: the attribute `data-rel="dialog"` on the `a` tag says "open this stuff up as a dialog rather than as a page". Alternately, we can let the content itself "know" that it is a dialog: if we replace `data-role="page"` with `data-role="dialog"` on the `div` that wraps the content, then the "page" will behave as a dialog regardless of how it is linked to.

Lesson 1, Activity 10: Buttons & Dialogs for Nan & Bob's Online

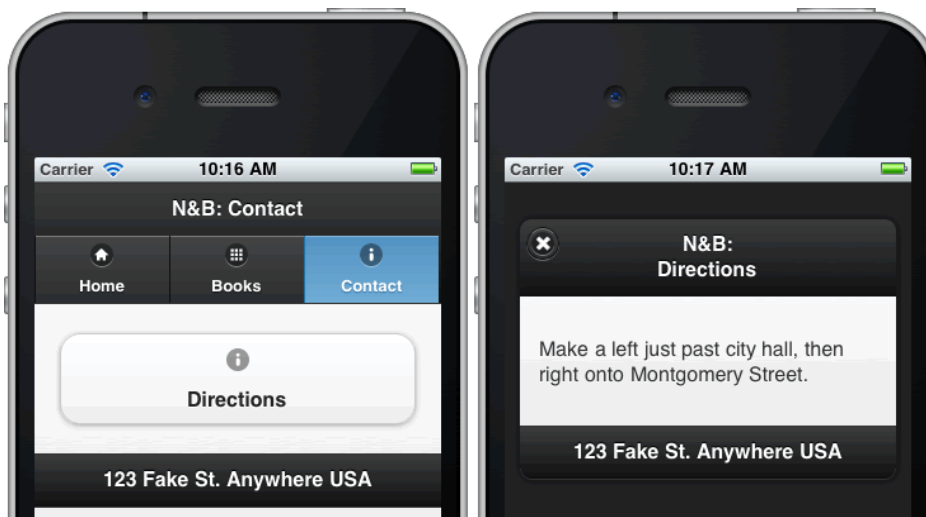
Duration: 20 to 30 minutes.

In this exercise, you will use what we have learned about buttons and dialogs to further flesh out the content on the Nan & Bob's Online site.

1. Open [GettingStartedjQuery/Exercises/dialogs.html](#) in a file editor.
2. Add a button from the home page to a join-mailing-list dialog.
3. Add a div (with `data-role="page"`) for the mailing list signup form.
4. Add a simple form, with input of type `email`, to the mailing list dialog page; you need not handle the processing of the form submission.
5. The home page and mailing list dialog should look like this:



6. Add a button from the contact page to a dialog showing driving directions.
7. Add a div (with `data-role="dialog"`) for the directions dialog; note that this means you won't need the `button` link opening this dialog to use `data-rel="dialog"`.
8. The contact page and directions dialog should look like this:



9. Test your work from a mobile browser.

Solution:

[GettingStartedjQuery/Solutions/dialogs.html](#)

```

---- CODE OMITTED ----

<div data-role="page" id="home">
  <div data-role="header">
    <h2>Nan & Bob's Online</h2>
    <div data-role="navbar" data-iconpos="top">
      <ul>
        <li>
          <a href="#home" data-icon="home" class="ui-btn-active ui-state-persist">
            Home
          </a>
        </li>
      </ul>
    </div>
  </div>
  <div data-role="main">
    <div data-role="button">Mailing List Signup</div>
  </div>
  <div data-role="page-footer">
    123 Fake St. Anywhere USA
  </div>
</div>

```

```

        </li>
        <li>
            <a href="#books" data-icon="grid">
                Books
            </a>
        </li>
        <li>
            <a href="#contact" data-icon="info">
                Contact
            </a>
        </li>
    </ul>
</div>
</div>
<div data-role="content">
    <a href="#signup" data-rel="dialog" data-role="button">Mailing List Signup</a>
</div>
<div data-role="footer">
    <h3>
        123 Fake St. Anywhere USA
    </h3>
</div>
</div>
---- C O D E   O M I T T E D ----

<div data-role="page" id="contact">
    <div data-role="header">
        <h2>N&B: Contact</h2>
        <div data-role="navbar" data-iconpos="top">
            <ul>
                <li>
                    <a href="#home" data-icon="home" data-transition="slideup">
                        Home
                    </a>
                </li>
                <li>
                    <a href="#books" data-icon="grid">
                        Books
                    </a>
                </li>
                <li>
                    <a href="#contact" data-icon="info" class="ui-btn-active ui-state-persist">
                        Contact
                    </a>
                </li>
            </ul>
        </div>
    </div>
    <div data-role="content">
        <a href="#directions" data-role="button" data-icon="info" data-iconpos="top">Directions</a>
    </div>
    <div data-role="footer">
        <h3>
            123 Fake St. Anywhere USA
        </h3>
    </div>
</div>

<div data-role="page" id="signup">
    <div data-role="header">
        <h2>N&B: Signup</h2>
    </div>
    <div data-role="content">
        <form action="dialogs.html" method="post">
            <input type="email" name="email" id="email" placeholder="email">
            <input type="submit" value="Join" data-mini="true">
        </form>
    </div>
    <div data-role="footer">
        <h3>
            123 Fake St. Anywhere USA
        </h3>
    </div>
</div>

<div data-role="dialog" id="directions">
    <div data-role="header">
        <h2>N&B: Directions</h2>
    </div>
    <div data-role="content">

```

```
<p>Make a left just past city hall, then right onto Montgomery Street.</p>
</div>
<div data-role="footer">
  <h3>
    123 Fake St. Anywhere USA
  </h3>
</div>
</div>
```

---- C O D E O M I T T E D ----

We add two buttons to the content of the existing pages: a button from the home page to the mailing list dialog and a button (with icon) from the contact page to the directions dialog.

The link to the mailing list includes the attribute `data-rel="dialog"`, which presents the mailing list content as a dialog. The link to the directions content does not include the `data-rel="dialog"` attribute; however, the directions content does show as a dialog, because its `data-role` attribute has value `dialog`.

The mailing list dialog presents a simple email-list signup form, with an `input` of type `email`. The directions dialog offers the user some driving directions as simple text content.