

How-To: Use jQuery to Enhance the Appearance and Usability of a Web Page in Netbeans

jQuery Overview

[jQuery](#) is a light-weight JavaScript library that allows programmers to easily and quickly add enhancements to the appearance and behaviors of their web pages. jQuery's syntax is concise and makes use of variables in the form of CSS selectors as a way of connecting an effect with any targeted element of the DOM, be it a unique element (`id`), or set of elements (`class`), or arbitrarily chosen. Because jQuery is JavaScript, it can be embedded in any project where JavaScript can be applied.

This tutorial demonstrates how to get started using jQuery in NetBeans projects, and take advantage of the IDE when working in any front-end project involving HTML, CSS, and JavaScript files. Primarily, you'll be shown how to invoke code completion on functions, and use the integrated API support. You'll also be introduced to key jQuery concepts, including the `$(document).ready` function call, the use of CSS-selector-like jQuery objects, and the chaining together of jQuery effects and behaviors. You'll also explore the benefits of the [jQuery UI library](#) by setting up a simple 'contacts list' example document, and applying the [jQuery accordion widget](#) to it.

Setting Up a NetBeans Project

1. Start by creating a new project. Select File > New Project (Ctrl-Shift-N; ⌘-Shift-N on Mac).
2. If you want to work in a PHP project, select the **PHP** category, then select or **PHP Application**.

If you want to work in a Java Web project, select the **Java Web** category, then select **Web Application**.

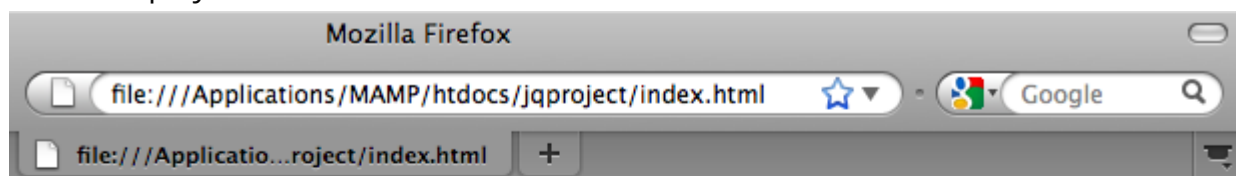
1. Click Next and name the project `jqproject`. Also, specify the directory on your computer where you want save the project. Click Next.
2. In Step 3, for purposes of this tutorial, accept default settings provided in the wizard.

Note: If you are creating a PHP project for the first time and need help, see Configuring Your Environment for PHP Development in the [PHP Learning Trail](#).

1. Click **Finish** to complete the wizard and create a new project. The `jqproject` opens in the Projects window, and the project welcome file opens in the editor.
2. Create a plain HTML file, which you can work in for the remainder of this tutorial. Because the jQuery code that we'll be adding does not require any communication with a back-end server, we'll just run the HTML file in a browser to view results.

Right-click the project node and choose New > HTML file (Ctrl-N).

1. Name the file `index`, then click **Finish**. In the Projects window, note that the new `index.html` file is listed within the project, and that the file opens in the editor.
2. Take a look at what the welcome page looks like in a browser. To do so, right-click the `index.html` node in the Projects window and choose View. (You can also choose View from the file's right-click menu in the editor.) The page displays in a browser window.



TODO write content

Figure 1. View the empty HTML template in a browser window

1. In the `index.html` file in the NetBeans editor, type in jQuery Test Project between the `<title>` tags, and create a pair of `<style>` tags within the page's `<head>` tags. (Changes in **bold**.)

```
<html>

<head>

  <title>*jQuery Test Project*</title>

  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

  *<style type="text/css">

    </style>*

</head>

<body>

  TODO write content

</body>

</html>
```

1. Configure your project so that the `index.html` file displays as the welcome file when the application is deployed and run. To do so, right-click the `jqproject` node in the Projects window and choose Properties.
 - **PHP projects:** Select the **Run Configuration** category, then type in `index.html` in the **Index File** field.
 - **Java Web projects:** Select the **Run** category, then type in `index.html` in the **Relative URL** field.
2. Click OK to close the Project Properties window and save changes.
3. At this stage, you can delete the original index file that was created with your project. In PHP projects, this is the `index.php` file; in Java Web projects, this is the `index.jsp` file.


To delete the file, right-click the file in the Projects window and choose Delete. In the confirmation dialog that displays, click **Yes**.

Adding the jQuery Library to the Project

Before we can begin working with jQuery, we must add the jQuery library to the project. If you haven't done so already, download the jQuery library from <http://jquery.com/>.

Choose the uncompressed version, i.e., 'Development', before downloading. Using the uncompressed version will allow you to examine the JavaScript code in the editor, and aid in any debugging processes.

To add the jQuery library to your NetBeans project, simply copy the library folder from its location on your computer, and paste it directly into your project in the IDE's Projects window. Details follow.

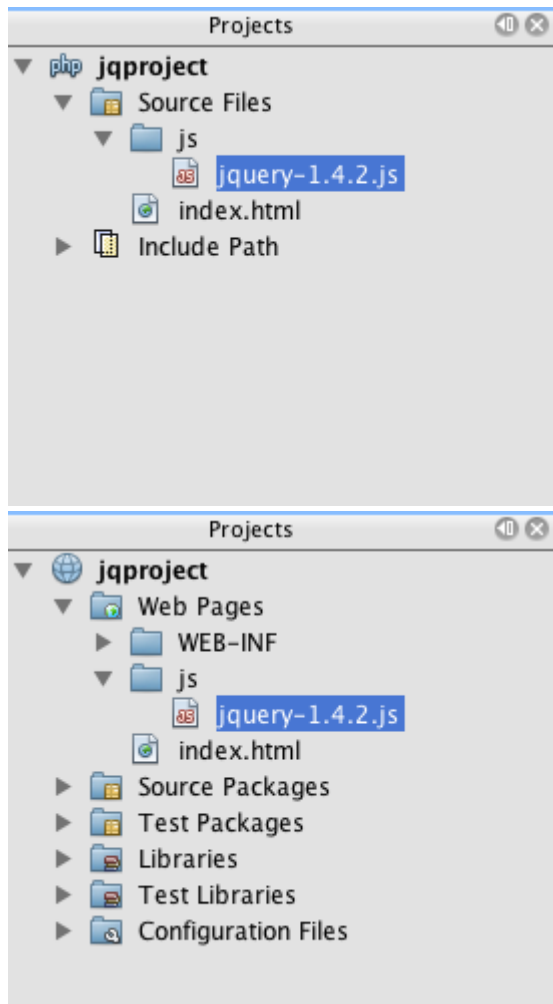
1. In the IDE, create a folder named `js`, and add it to your project. To do so, click the New File () button in the IDE's toolbar. (Alternatively, press Ctrl-N; ⌘-N on Mac.)
2. Select the **Other** category, then select **Folder**.
3. Name the folder `js`.

For Java Web projects, ensure that you place the `js` folder in the project's web root. To do so, enter `web` in the **Parent Folder** field.

1. Click **Finish** to exit the wizard.
2. Locate the jQuery library that you downloaded onto your computer. To date, the current library version is 1.4.2, so the file is typically named `jquery-1.4.2.js`. Copy the file to your clipboard (Ctrl-C; ⌘-C on Mac).
3. Paste the library file into the new `js` folder. To do so, right-click the `js` and choose Paste (Ctrl-V; ⌘-V on Mac). The `jquery-1.4.2.js` file node appears within the folder.

==== PHP project:

==== Java Web project:



1. In the editor, reference the jQuery library from the `index.html` file. To do so, add a pair of `<script>` tags and use the `src` attribute to point to the library location. (Changes in **bold**.)

```
<html>
  <head>
    <title>jQuery Test Project</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    *<script type="text/javascript" src="js/jquery-1.4.2.js"></script>*
    <style type="text/css">

  </style>
```

```
</head>
```

```
...
```

1. Save the file (Ctrl-S; ⌘-S on Mac).

The jQuery library is now included in the `jqproject` project, and referenced from our `index.html` file. We can begin adding jQuery functionality to the page.

Getting Acquainted with jQuery

jQuery works by connecting dynamically-applied JavaScript attributes and behaviors to elements of the DOM (Document Object Model). Let's add an element to the DOM and try to affect its properties. We'll create a heading that changes color from black to blue when we click on it.

1. We start by creating the heading, structurally an `<h1>` element. Remove the 'TODO write content' comment and enter the following between the `<body>` tags:

```
<h1>Test.</h1>
```

1. Now we'll create a CSS class that makes an element appear blue when it is applied. Enter the following between the `<style>` tags in the `<head>` of the document:

```
.blue { color: blue; }
```

1. Next we'll set up a place to put our jQuery commands. Add a new set of `<script>` tags to the `<head>` of the document, e.g., after the `<script>` tags linking to the jQuery library. (Changes in **bold.**)

```
<html>
  <head>
    <title>jQuery Test Project</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <script type="text/javascript" src="js/jquery-1.3.2.js"></script>
    *<script type="text/javascript">
```

```
</script>*

<style type="text/css">
    .blue { color: blue; }
</style>
</head>
...
```

You can tidy up your code by right-clicking in the editor and choosing Format.

The jQuery instructions that we will add must be executed only after all of the elements of the DOM have been loaded by the browser. This is important because jQuery behaviors connect to elements of the DOM, and these elements must be available to jQuery in order to get the results we expect. jQuery takes care of this for us through its built-in `(document).ready` function, which follows the jQuery object, represented by `$`.

1. Enter this construction between the script tags you just created:

```
$(document).ready(function(){

});
```

There is also an abbreviated version of this function that can alternately be used:

```
$(function(){

});
```

Our instructions for jQuery take the form of a JavaScript method, with an optional object literal representing an array of parameters, and must be placed between the curly braces `{ }` inside the `(document).ready` function in order to execute only at the proper time, which is after the DOM has completely loaded.

At this stage, the `index.html` file should look as follows:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>jQuery Test Project</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <script type="text/javascript" src="js/jquery-1.3.2.js"></script>

    <script type="text/javascript">
      $(document).ready(function(){

        });
    </script>

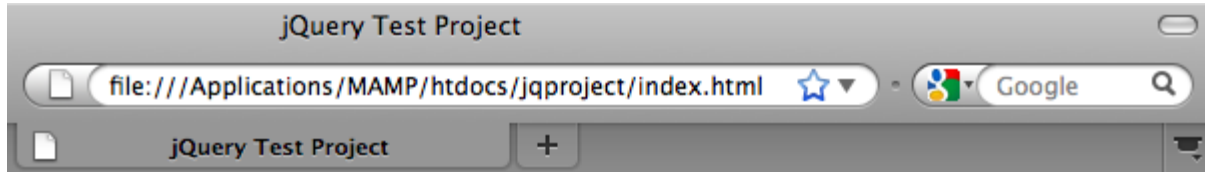
    <style type="text/css">
      .blue { color: blue; }
    </style>
  </head>
  <body>
    <h1>Test.</h1>
  </body>
</html>
```

1. To demonstrate how jQuery syntax works, let's try something simple. We'll add jQuery instructions to our page that will make the word 'Test' turn blue when we click on it. To accomplish this, we want jQuery to add the CSS class `.blue` to the `<h1>` element of the DOM when it receives a mouse click.

Enter the following code inside the `(document).ready` function, between the braces `{}`:


```
$("#h1").click(function(){
    $(this).addClass("blue");
});
```

1. Save the document (Ctrl-S; ⌘-S on Mac), then right-click in the editor and choose View to load it in your web browser. Test it to see if it works. When you click on the word 'Test', it should turn blue.



Test.

Figure 2. Text turns blue when clicked upon

This example uses the jQuery `click()` function to invoke the jQuery `addClass()` function when an element matching the CSS selector "h1" is encountered. The `$(this)` refers back to the calling element. If we were to add more `<h1>`'s to our page, the same behavior will be applied to all of them with this single set of rules, and each will interact with jQuery independently. (You can try this yourself as a quick exercise.)

1. Another important quality of jQuery is that functions can be simply chained together to create more complicated or even sequenced behaviors. To demonstrate this let's add a jQuery instruction for a `fadeOut()` to our `click()` function. Place a `fadeOut("slow")` jQuery function after the `addClass` function so that the line of code looks like this:

```
$(this).addClass("blue").fadeOut("slow");
```

The complete jQuery function should now look like this:

```

$(document).ready(function(){
    $("h1").click(function(){
        $(this).addClass("blue").fadeOut("slow");
    });
});

```

1. In the browser, refresh the page and then click 'Test.' You will see that it turns blue, and then fades out, disappearing from the page. (To try it again, you must refresh the page.)

NetBeans Code Completion and API Support

Whenever you type in the editor, you can invoke code-completion by pressing Ctrl-Space. The IDE presents a list of suggestions which you can choose from, as well as an API documentation window that defines the listed items, provides code snippet examples, and shows target browser support.

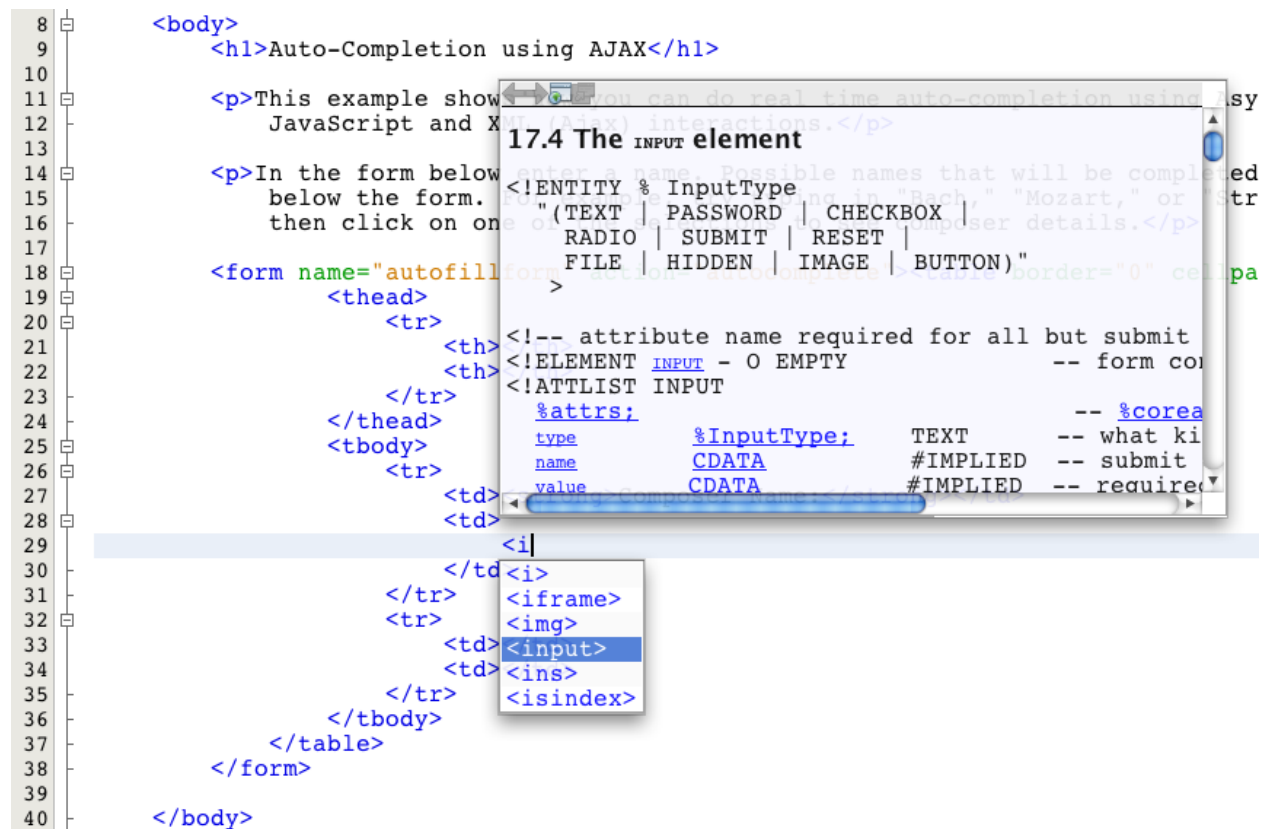


Figure 3. Press Ctrl-Space to view code completion and API documentation windows

You can specify the target browsers for code completion and API documentation by opening the IDE's JavaScript options window. Choose Tools > Options (NetBeans > Preferences on Mac), then choose Miscellaneous > JavaScript.

Adding the jQuery Accordion Widget to the Project

We created the simple test above by using JavaScript behaviors that are included in the core jQuery library. Now let's examine a more real-world example by setting up an employee contact list using basic HTML markup. We'll then apply the [jQuery accordion widget](#) to the contact list.

The accordion widget is part of the [jQuery UI library](#). The UI library is built on top of the core library, and provides a modular approach to enabling interactions, widgets and effects to your web pages. You can keep file sizes to a minimum and conveniently select only the components you need from the jQuery's download interface at <http://jqueryui.com/download>.

If you have not already done so, visit <http://jqueryui.com/download> and download the accordion navigation widget. Note that when you select the accordion widget, the UI Core library, and Widget Factory are also automatically selected. Also note that from the download page, the 'UI lightness' theme is selected by default, and is included in your download package. We'll be applying this theme to our contact list in the [following section](#).

1. Paste the following code into your document in place of `<h1>Test.</h1>`.

```
<div id="infolist">

  <h3><a href="#">Mary Adams</a></h3>

  <div>

    <ul>

      <li><h4>Vice President</h4></li>

      <li><b>phone:</b> x8234</li>

      <li><b>office:</b> 102 Bldg 1</li>

      <li><b>email:</b> m.adams@company.com</li>

    </ul>

  </div>

</div>
```

```
</ul>

<br clear="all">

</div>

<h3><a href="#">John Matthews</a></h3>

<div>

  <ul>

    <li><h4>Middle Manager</h4></li>

    <li><b>phone:</b> x3082</li>

    <li><b>office:</b> 307 Bldg 1</li>

    <li><b>email:</b> j.matthews@company.com</li>

  </ul>

  <br clear="all">

</div>

<h3><a href="#">Sam Jackson</a></h3>

<div>

  <ul>

    <li><h4>Deputy Assistant</h4></li>

    <li><b>phone:</b> x3494</li>

    <li><b>office:</b> 457 Bldg 2</li>

    <li><b>email:</b> s.jackson@company.com</li>

  </ul>

  <br clear="all">

</div>
```

```

<h3><a href="#">Jennifer Brooks</a></h3>
<div>
  
  <ul>
    <li><h4>Senior Technician</h4></li>
    <li><b>phone:</b> x9430</li>
    <li><b>office:</b> 327 Bldg 2</li>
    <li><b>email:</b> j.brooks@company.com</li>
  </ul>
  <br clear="all">
</div>
</div>

```

Observe that the overall enclosing `<div>` element is given an `id` attribute with a value of `infolist`. Within this `<div>` element, there are four sets of `<h3>` tags and `<div>` tags that contain an image and unordered list.

1. Add a few inline CSS rules to the above markup. Delete the `.blue` style rule you created for testing purposes above. In its place, add the following rules. (Changes in **bold**.)

```

<style type="text/css">
  *ul {list-style-type: none}
  img {padding-right: 20px; float:left}

  #infolist {width:500px}*
</style>

```

When you type within `<style>` tags, take advantage of the IDE's built-in CSS code-completion by pressing `Ctrl-Space`.

1. Save the file (Ctrl-S; ⌘-S on Mac).
2. Now we'll add the the JPG portraits that are referenced in the above code fragment to our project. Retrieve the pix directory from the [project resources you downloaded earlier](#) and copy the entire directory to your project folder, placing it at the same level as index.html. After a brief moment, NetBeans automatically updates the Projects window to reflect that a new directory has been manually added to the project.
3. Switch to your browser and refresh the page.

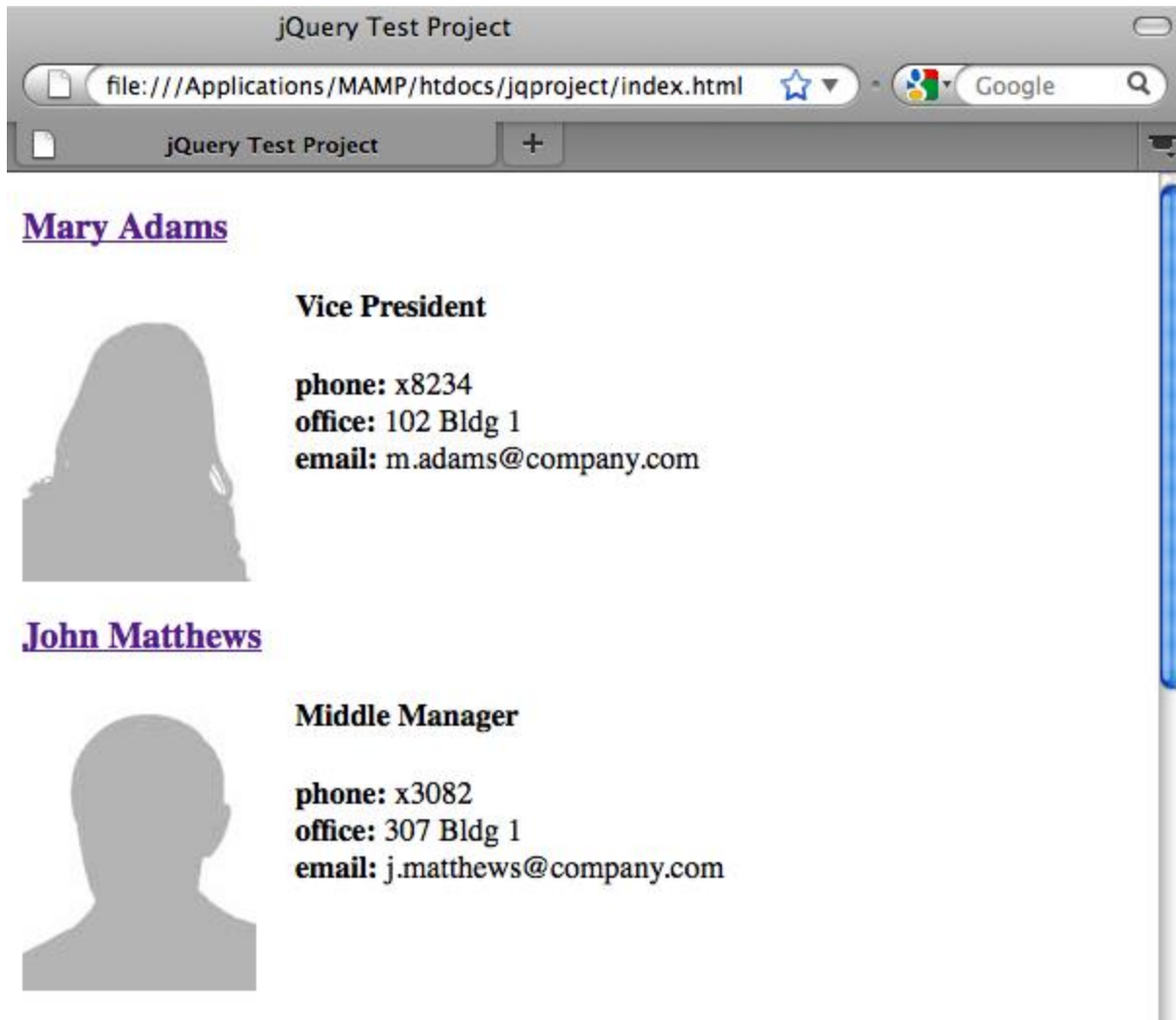


Figure 4. Structured list displays in a browser

There are a number of problems with this document that we will address. Firstly, it is more difficult than it needs to be to scan the list quickly to find the person you're looking for: one must scroll the page and visually inspect a lot of information that may not be of immediate interest. Four contacts in a list might be manageable, but

if the number grew to say, 50, then the list would become much more difficult to use. Secondly, the document is visually plain, and is unlikely to blend in esthetically with most web site designs, particularly designs that have a strong graphic identity. We will address these issues by using the jQuery accordion widget, in combination with jQuery UI's default theme.

1. To produce the accordion effect, navigate to the location on your computer where you downloaded the accordion widget. Within the downloaded folder, you'll find a folder named 'development-bundle'. Within the `development-bundle` folder, expand the `ui` folder and locate the following three scripts:
 - `jquery.ui.core.js`
 - `jquery.ui.widget.js`
 - `jquery.ui.accordion.js`

Development versions of toolkit scripts are *unminimized*, meaning that their code is human-readable when viewed in an editor. Normally, you would want to switch to the compressed, minimized versions for a production-ready application in order to conserve download times.

1. Copy (Ctrl-C; ⌘-C on Mac) the three scripts and, back in the IDE, paste them in the `js` folder you [created earlier](#) in your `jqproject` folder.

You can paste by either pressing Ctrl-V (⌘-V on Mac), or right-clicking the `js` folder and choosing Paste.

The `development-bundle > ui` folder also contains a file named `jquery-ui-1.8.1.custom.js`. This file combines the three scripts listed above into a single script. You could equally paste this file into your project in place of the three individual scripts.

1. Reference the scripts in your `index.html` page by entering three `<script>` tags linking to these new JavaScript files. You can add the `<script>` tags immediately after the `<script>` tags that refers to the core jQuery library `jquery-1.4.2.js`. Use the existing `<script>` tags as a model.
2. Delete the test code we created inside the `(document).ready` function. You no longer need it.

The `<head>` tags of your file should now look as follows.

```
<head>
```

```
<title>jQuery Test Project</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<script type="text/javascript" src="js/jquery-1.4.2.js"></script>
<script type="text/javascript" src="js/jquery.ui.core.js"></script>
<script type="text/javascript" src="js/jquery.ui.widget.js"></script>
<script type="text/javascript" src="js/jquery.ui.accordion.js"></script>

<script type="text/javascript">
    $(document).ready(function(){

    });
</script>
</head>
```

1. To make our static, unstyled list take on the accordion behavior is as simple as adding a single line of jQuery code. Enter this line into the `(document).ready` function. (Changes in **bold**.)

```
$(document).ready(function(){
    *$("#infoList").accordion({
        autoHeight: false
    });*
});
```

In this line of code, `#infoList` is a CSS selector connected to a unique DOM element that has an `id` attribute with the value `infoList`; in other words, our contacts list. It is connected using typical JavaScript dot notation (`.`) to the jQuery instruction that uses the `accordion()` method to display this element.

You've also specified `'autoHeight: false'` in the above snippet. This prevents the accordion widget from setting the height of each panel based on the highest

content part contained within the markup. For more information, consult the [accordion API documentation](#).

1. Save the file (Ctrl-S; ⌘-S on Mac).
2. Go back to the web browser and refresh. Click on one of the names (other than the top one) to see the accordion effect in action. The jQuery accordion widget handles all the details of handling the DOM and responding to user mouse clicks.

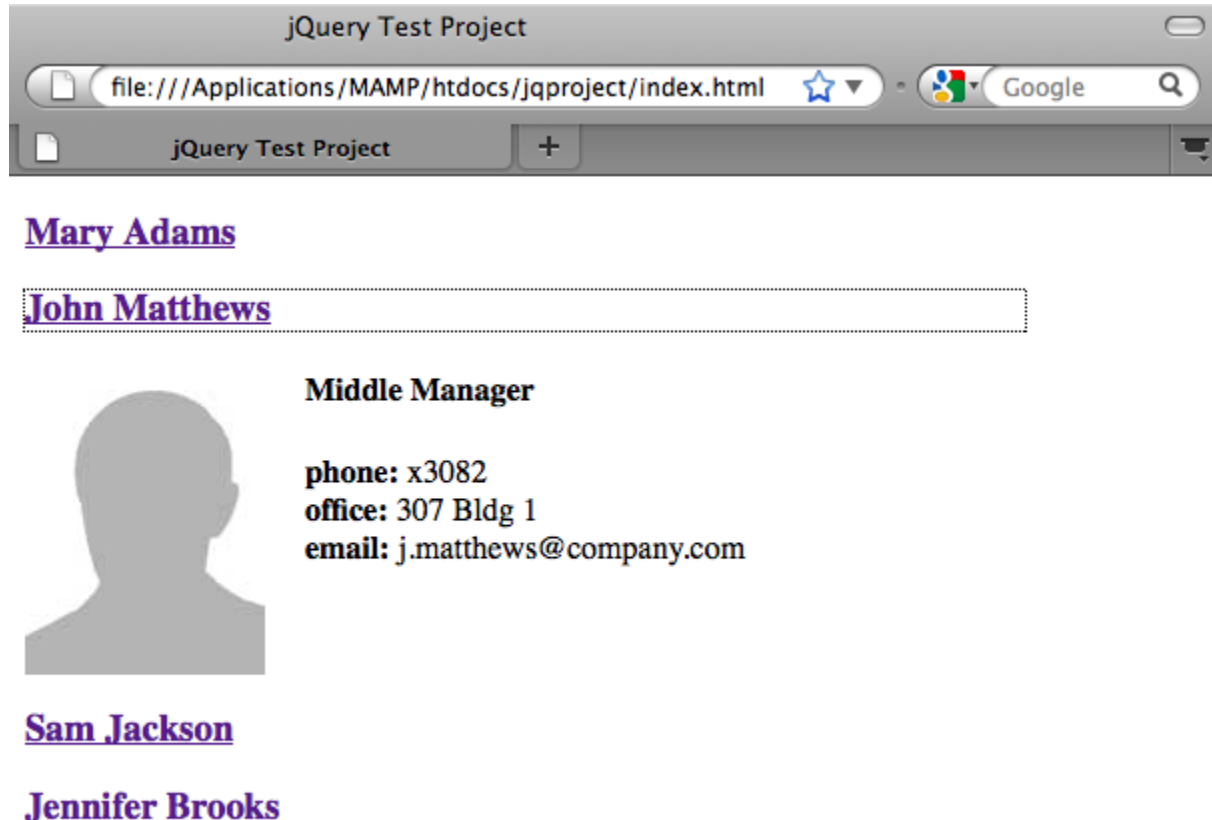



Figure 5. Accordion widget handles user clicks and produces the accordion effect

Using jQuery's Default Theme for Style Enhancement

Our project now has the behavior we want, but it looks quite plain and still lacks a well-organized appearance. Let's address this by incorporating jQuery's default 'UI lightness' theme.

1. Navigate to the location on your computer where you downloaded the accordion widget. Within the downloaded folder, expand the `development-bundle > themes > ui-lightness` folder.
2. Within the `ui-lightness` folder, copy (Ctrl-C; ⌘-C on Mac) the `jquery-ui-1.8.1.custom.css` file, and the `images` folder, which contains all of the images necessary for the theme to render properly.
3. In the IDE, create a new folder within your project named `css`. This folder will contain the 'UI lightness' theme for the accordion widget.

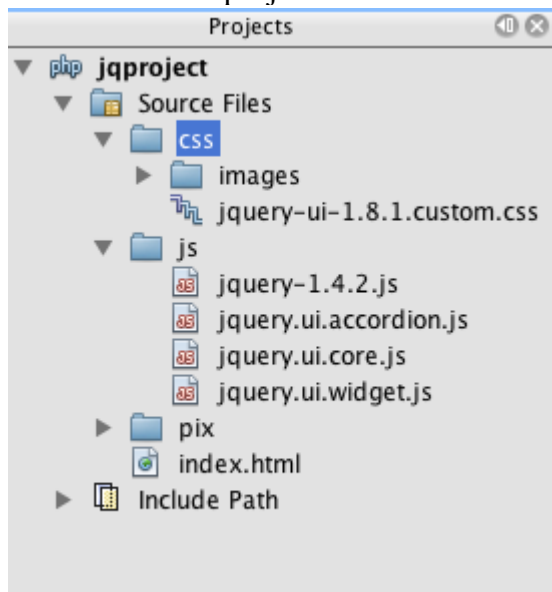
To do so, right-click the project node and choose `New > Folder`. (If `Folder` doesn't appear as an option, click the `New File` () button in the IDE's toolbar, then choose `Other > Folder` in the `New File` wizard.) Name the folder `css` and place it within the same directory as your `index.html` file.

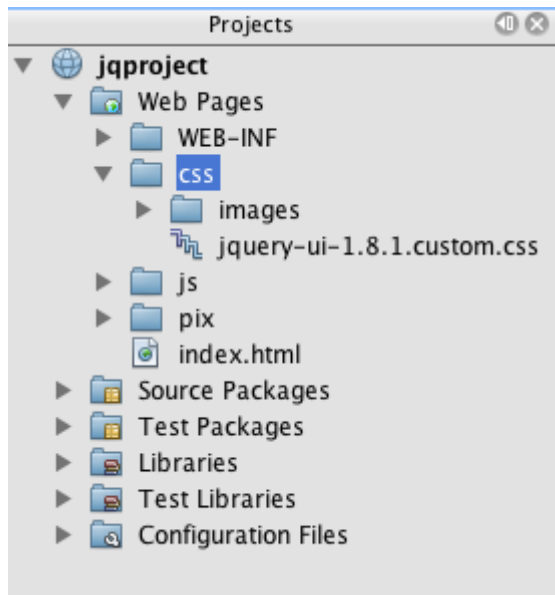
For Java Web projects, ensure that you place the `css` folder in the project's web root. To do so, enter `web` in the **Parent Folder** field.

1. Paste the two items directly into the new `css` folder. To do so, right-click the `css` folder node and choose `Paste`. Your project folder should look as follows.

==== PHP project:

==== Java Web project:





1. Reference the `jquery-ui-1.8.1.custom.css` file from within your `index.html` web page. Add the following `<link>` tag within the page's head.

```
<link rel="stylesheet" href="css/jquery-ui-1.8.1.custom.css" type="text/css">
```

1. Save the file (Ctrl-S; ⌘-S on Mac).
2. Return to the web browser and refresh the page. Notice that the list now displays using jQuery's default theme, which is an esthetic improvement over the plain, unstylized version.

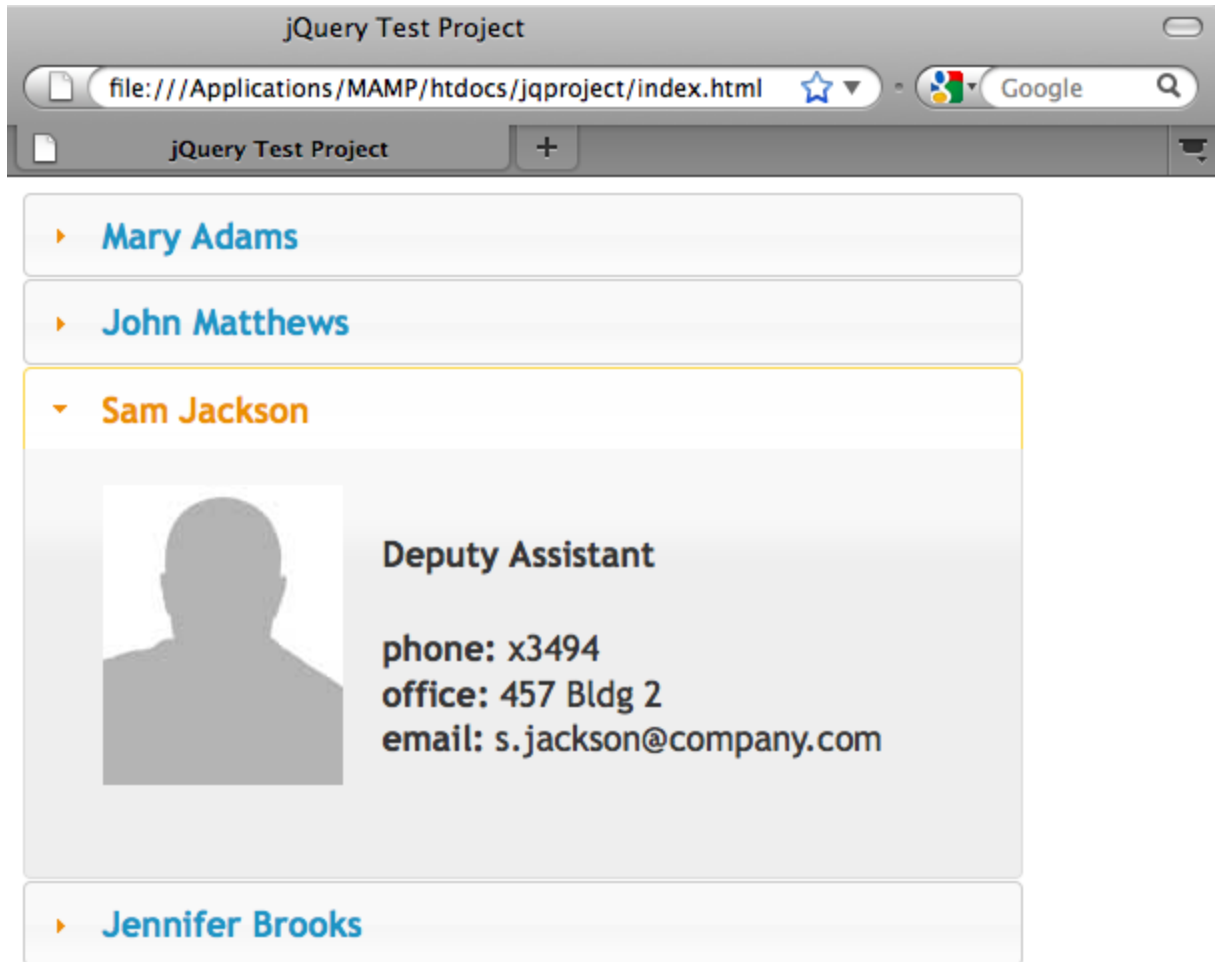


Figure 6. jQuery default theme enhances the appearance of the accordion widget

Summary

In this tutorial, you have learned how to add jQuery libraries to your project, as well as how to write some basic instructions using the jQuery syntax. You also learned how jQuery interacts with the DOM (Document Object Model) using variables that resemble CSS selectors to affect the appearance and behavior of elements on a web page.

Finally, you briefly explored the capabilities of the jQuery UI library by applying the accordion widget to a simple contact list. After implementing the accordion effect, you applied jQuery's default style theme to the list. You should now be better able

to appreciate how jQuery can be used to create dynamic web pages, while improving overall appearance and usability.

Courtesy: <http://netbeans.apache.org/kb/docs/web/js-toolkits-jquery.html>

Modified: 2021.10.04.1.50.PM

Dököll Solutions, Inc.