What is jQuery?

jQuery is a concise and fast JavaScript library that can be used to simplify event handling, HTML document traversing, Ajax interactions and animation for speedy website development. jQuery simplifies the HTML's client-side scripting, thus simplifying Web 2.0 applications development.

jQuery is a free, open-source and dual-licensed library under the GNU General Public License. It is considered one of the favorite JavaScript (JS) libraries available today. As of 2012, it is used by more than half of the Web's top sites.

(Credits: Techopedia)

Table of Contents

Agenda

1. Define jQuery

2. Useful links for information/Getting Started (in References/Resources)

3. Bootstrap JS section Overview (How tabs work)

4. Cloning the Repo

5. Add to GAE (if needed)

6. Overview of completed exercise (Expected behavior)

Exercise

Overview:

Clone the repository at https://github.com/KaranKamath/Orbital-jQuery, and edit the jquery-rss.js within the js directory.

Implement the TODO sections following the hints if necessary.

Steps:

- Reviewing existing code in the js file
- Handling the submit event (Notice how the tab pane for the feed is added to the page in on calling getFeedWithURL but is currently hidden as the tab control is missing).
- Adding the tab control on submit (look at how the addTabPaneForFeed function is implemented).
- Implementing the show/hide paragraphs effect
- Viewing some docs for the bootstrap-progressbar plugin
- Handling the animations for the progress bar using the plugin (Note that you can use the completion callback of fadeOut too, but we prefer you try $.when and its chained done in this case for learning value)

Some Example RSS Feeds:

http://www.xkcd.com/rss.xml

http://rss.cnn.com/rss/edition.rss

http://feeds.wired.com/wired/index

http://feeds.feedburner.com/CrackedRSS

Homework

1. Extend the functionality of the the rss feed reader. This is an open ended challenge, and does not need to include too much code. What it should try to be though, is relevant and interesting.

2. Using this application or otherwise, showcase your use of jQuery for making an AJAX call. You will have to read up on how to do this.

Post your answers to piazza including [jQuery-Mission Control] in the title. It can be (formatted) code snippets and screenshots. You can also issue a pull request so I can try to add your extension to the next release :P.

What this exercise covers

Some basic animations, DOM manipulation and event handling.

An important thing this exercise does not cover

AJAX. Chances are, you may need it in your application. Look it up later and try using jQuery for it.

Common Problems

1. jQuery Code not wrapped in $(document).ready(), leads to attempts at manipulating objects that are not loaded yet. See this. As a rule, enclose all your code within it, like so:

```
$(document).ready(function() {
    // Your code here..
});
```

2. If you have multiple script files, make sure to import them in the right order. The independent ones should come first.

3. If you are using a submit button to perform a custom task, be sure to prevent it's default action. It may refresh the page if you don't.

```
$("#myButton").click(function(event){
    event.preventDefault();
    // Your code here...
});
```

4. If you need events to happen in order, check out the documentation for $.deferred and $.when, amongst other things. These may be especially useful for animations.

5. In case you run into other common problems, let us know on Piazza!

Reference

Official jQuery site : It has fantastic documentation of the API, and addresses a range of common issues. This should be your first stop in anything jQuery.

jQuery (Stack Overflow) : Turns out your new best friend knows a fair bit about jQuery too. I'd suggest not just looking for answers but asking questions too (in case you can't find your problem). People are willing to help debug YOUR problem. How cool is that.

Tags: jquery (usually in relation with html, css, ajax, etc.)

Plugins

One of the advantages of using jQuery is the huge number of third-party plugins available at your disposal. You can easily enhance your app using these, especially for common and expected features. It is often as easy as including a script and calling a function.

Here is the offical plug-in registry: http://plugins.jquery.com/

Resources

1. http://flukeout.github.io/ :Go from 0 to CSS selector pro in 26 levels.

2. http://www.codecademy.com/tracks/jquery : Really basic and compulsory if you are new to jQuery. They estimate you will take 3 hours. But really, you should be done *much* sooner than that.

3. http://learn.jquery.com/using-jquery-core/faq/ : Read through. It's not really long and worth reading.

4. http://try.jquery.com/ : Comprehensive set of tutorials.

Technical Terminology

1. Software Library:

   A software library generally consists of pre-written code, classes, procedures, scripts, configuration data and more. Typically, a developer might manually add a software library to a program to achieve more functionality or to automate a process without writing code for it. For example, when developing a mathematical program or application, a developer may add a mathematics software library to the program to eliminate the need for writing complex functions. All of the available functions within a software library can just be called/used within the program body without defining them explicitly. Similarly, a compiler might automatically add a related software library to a program on run time.

   Context: jQuery is a JavaScript library

   (Credits: Techopedia)

2. API:

   API, an abbreviation of ***a****pplication* ***p****rogram* ***i****nterface*, is a set of routines, protocols, and tools for building software applications. The API specifies how software components should interact and are used when programming graphical user interface (GUI) components. A good API makes it easier to develop a program by providing all the building blocks. A programmer then puts the blocks together.

   Context: 99% of the time, all you need to worry about is how to use jQuery's API.

   (Credits: Webopedia)

3. DOM (Document Object Model): All you need to know is jQuery manipulates this by using "jQuery objects" which contain the relevant parts of the DOM.
4. AJAX:

5. Callback Functions: In computer programming, a callback is a piece of executable code that is passed as an argument to other code, which is expected to call back (execute) the argument at some convenient time. The invocation may be immediate as in a

synchronous callback or it might happen at later time, as in an asynchronous callback. In all cases, the intention is to specify a function or subroutine as an entity that is, depending on the language, more or less similar to a variable. (Wikipedia)

Context: When we define event handlers, we are essentially using callbacks, i.e. passing our functions as arguments to a jQuery function. This ensures our function is executed by jQuery when an event occurs.

A Word of Warning

I'm sure most of you are aware of the hammer and nail conundrum. Since we get jQuery for free while working with Twitter Bootstrap, it makes sense to leverage it for our web app. However, if you didn't need Bootstrap, you should think about whether you really need jQuery. The online community is quite divided. If we had to play Devil's advocate:

1. Vanilla JS is way faster and capable of doing everything jQuery does. Although jQuery often makes it much simpler to do.

2. We should consider whether we really need to import a (relatively) huge file into our webpage to do something simple.

Follow Up

If you have any doubts, errors or questions, feel free to bug me at karankamath26@gmail.com, and I'll help you google the solutions :P.