CSCE 120: Learning To Code

Manipulating Data I
Hacktivity 7.2

Introduction

Prior to engaging in this hacktivity, you should have completed all of the pre-class activities as outlined in this module. At the start of class, you will be randomly assigned a partner to work with on the entirety of this hacktivity as a *peer programming* activity. Your instructor will inform you of your partner for this class.

One of you will be the driver and the other will take on the navigator role. Recall that a driver is in charge of the keyboard and computer while the navigator is in charge of the handout and directing the activity. However, you are *both* responsible for contributing and discussing solutions. If you were a driver/navigator in the prior activity, switch roles for this activity.

You will use the same project from the previous Hacktivity. If you need to, you can re-download it from GitHub using the URL, `https://github.com/cbourke/jQueryProject`.

1 Making Dynamic Elements

You will familiarize yourself further with jQuery and jQuery UI by adding dynamic elements to a web page. We have provided a starter webpage and some code in the `DynamicElements` folder. For the first few exercises you should place your code inside the `ready()` function in the `main.js` source file.

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

Recall that the code in this function is executed only after the page has fully loaded.

1.1 Page Load Animations

Make the page load more dynamic by animating some of the major elements. jQuery UI provides a `show()` function that displays hidden elements. A live demo of the various types of animation can be found here: https://jqueryui.com/show/ while the documentation for using this function can be found here: http://api.jqueryui.com/show/.

1. Animate the "jumbotron" element using the `show()` function. Experiment with different animations and options.
2. Animate the "jQuery Conference" information box using a different animation.
3. Make all the images fade in but do it more slowly than the other animations to give a visually distinctive animation.

1.2 Teaser Preview

Many webpages give a "teaser" of their content by displaying the first few words or lines of an article and hiding the rest. A link or button is displayed to allow a user to "read more." When clicked, the rest of the content is displayed. We'll implement this feature using jQuery.

In particular, when the page is loaded, we will change the content of the `mainArticle` element to provide a "teaser" and a "Read More" button. We have embedded a step-by-step outline in the code, but the steps below provide more details.

1. Start by using jQuery to get the HTML contents of the main article. Save the HTML contents into a variable so that we can "restore" them later on.

   Note: you should store the contents in a *global* variable so that it can be accessed in the `reveal()` function when the user presses the button. To save a value to a global variable, simply omit the usual `var` keyword.
2. Use jQuery to get the *text* content of the `leadParagraph` element and save it to a variable (this need not be a global variable).
3. *Truncate* the text variable to (say) the first 50 characters. You can truncate a string using the `substr` function. For example, `s.substr(0, 50)` would return a new string containing only the first 50 characters of the string `s`.
4. Create a new string variable that consists of the truncated lead paragraph, an ellipsis ("...") and an HTML formatted button (or link) with the text "Read More." In the button, specify an `onclick` attribute that calls the `reveal()` function.
5. Use jQuery to replace the HTML content of the article with the string you just created.
6. In the `reveal()` function, use jQuery to *restore* the original content of the main article (the content you saved in the variable in step 1).

7. Animate the restoration of the article's content to give visual feedback to the user.

1.3 Collapsable Comments

Notice that the page has a comments section. Many popular forums have *threaded* comments: users post and reply to comments with each reply being nested (indented) under the original comment. Inspired by Reddit, we have made it so that each comment and sub-thread can be collapsed or expanded for easier reading. Try it out with the "Collapse All" button (which changes to an "Expand All" button) to see what the intended effect is.

In this exercise, you will implement this functionality for each individual comment. The "collapse" links, `[-]` and the "expand" links `[+]` are already clickable and invoke a function `commentToggle()`. You will need to implement this function. Some starter example code has already been provided, but you will need to modify and add code to make it fully functional. Code has been provided to demonstrate the logic and operations for collapsing the first comment.

- Add code to handle the "expand" case (when a user clicks on `[+]`, it should show the comment's body).
- Modify the code to use the `commentNumber` parameter passed to the function so that it will work on any comment, not just the first one.

2 jQuery UI Widgets

The jQuery UI library provides several standard graphical user interface *widgets* that can be brought into your web apps and pages that provide various dynamic functionality. These widgets are also highly customizable.

For this exercise, you will familiarize yourself with the *accordion* widget that allows multiple sections of text or data to be collapsed so that the user only has to focus on one section at a time.

1. Open the `majors.html` page and note that we've provided several sections describing majors and minors offered by our department.
2. Read the basic information on the jQuery UI accordion widget:
   http://jqueryui.com/accordion/
3. Integrate the accordion widget and apply it to the content in the `majors.html` page
4. By default, the height of the accordion is the maximum height of any of the subsections. Change this by configuring your accordion with the "no auto height" option