

# CSCI-498

February 19, 2009

## JQuery

### Overview

In this worksheet you will work with JQuery and see how to integrate it into your pages to make them more interactive and intuitive. I have created an HTML page already for you that includes examples of most of the events in JQuery, but in this worksheet you will be working with everything else in JQuery.

I've created a template page for both pages that you will be creating today. You should be able to follow the directions and just fill in the blanks. Grab the file at:  
<http://myweb.csuchico.edu/~stodd3/jqueryTemplate.zip>

### Form Validation

Create a form that includes fields for first name, last name, phone number, street address, city, state, and zip code. Apply an appropriate ID attribute to each input tag. You should include a label tag for each input and wrap both the label and the input in a paragraph tag (you'll see why in a second). Assign the paragraph tag the class formInput.

Now add in a CSS style block to the header with the classes formHover and formError. Fill them in with an appropriate border and background.

For the Javascript create a new script block with a document ready function. In the document ready function use the hover to apply and remove the formHover class to the paragraph tags (hint: use \$(".formInput") as the selector). Add to the document ready function the blur event for each input (not paragraph) to call the function validateForm(this).

Create the validate form function to include several if statements (one for each input). Now use the Javascript match function and the following regular expressions to validate the inputs:

- ❖ Name (both) - `^[A-Z][a-z]+$`
- ❖ Phone Number - `^\([\]?[0-9]{3}[\)\-\.]?[0-9]{3}[\-\.]?[0-9]{4}$`
- ❖ Address - `^[0-9]+ [A-Za-z0-9#\.\ ]+$`
- ❖ City - `^[A-Za-z\ ]+$`
- ❖ State - `^[A-Z]{2}$`
- ❖ Zip Code - `^[0-9]{5}$`

If any of the inputs are errored, add the class formError. If they are successful, remove the class.

## **Selector Examples**

Use the divAdder.html template and add to the body three elements: two buttons and one div. The buttons should be labeled "Add Div" and "Count Divs". Now apply appropriate ID attributes to each. Apply the ID "divGrid" to the empty div.

For the Javascript, set the click event listener on the first button to append a new div to the divGrid. Inside of each div should be the count of the number of divs in place so far. Set the click event listener for the second button to display an alert with the number of divs that have the class "selectedDiv".

Now add in a live event listener for the click event on all divs contained by the divGrid. Toggle between adding and removing the selectedDiv class. You should use an if statement and the hasClass() function for this.

## **JQuery Animation**

Modify the above page to animate the selected divs when you press the count button. I'll leave it up to you on how you want to have them animated. Try to be creative and have a series of animations play. Some examples are having the background or border color fade to another color and then back to their original color. You will need to use the call back function of the animations in order to create a sequence of animations. See the countdown animation of the JQuery playground page for how to do this.