**Assignment No 2**

**Web Application Development**

Original Assignment Link : <http://web.stanford.edu/~ouster/cgi-bin/cs142-winter14/project.php?project=6>

So, we’re doing yet another Stanford Assignment ☺

# CS142 Project #6: Javascript and the DOM

## Problem 1: Traversing a Table (15 points)

Create a file TableScan.js that implements a Javascript class named TableScan with a static method sumColumn. This method takes two arguments consisting of the id attribute for a <table> and a string identifying a column in the table. The method must search the topmost row of the table to find an entry matching the string argument. This selects a particular column in the table (if there is no matching column, the method returns 0). Then the method must find all of the values in that column (excluding the header value), treat them as numbers, and return the sum of all of those values. If a column contains any non-numeric values, you should treat them as if they were 0. Also, beware that browsers insert a <tbody> element around all of the <tr> elements in a table, if the table doesn't already contain a <tbody>.

Once you have created the Javascript class, **open the zip file in Assignment Helper Folder** and copy the files tableScan.html and tableScan.css into the project6 directory. These files represent an HTML page containing three sample tables. Modify tableScan.html to add the following elements:

* A form containing two text entries, which can be used to enter the id of a table and the string identifying a column within that table.
* A button or <a> element that the user can click after entering information in the form. When this control is clicked, the page must extract the values from the form elements and invoke the sumColumn method. You should not actually submit the form to the server: everything will happen on the browser.
* The page should also display a sentence of the following form: "Column *xxx* in table *yyy* sums to *zzz*". When the user clicks on the control, *xxx*, *yyy*, and *zzz* in this sentence should get filled in with the arguments and result of the sumColumn method.

You may add additional styles to tableScan.css if needed.

## Additional Requirements, Hints, etc.

* You may not use JQuery or any other Javascript library package in this project (or anywhere else in CS 142). If you were developing a real Web site then you should definitely take advantage of existing Javascript libraries, but for this class we want you to learn about the low-level Javascript/DOM features.
* You may find some of the following Javascript built-in functions useful in this project:
  + Methods of the Date class.
  + parseInt, parseFloat, isNan.
* You may find some of the following DOM element properties useful in this project:
  + cells
  + firstChild
  + nextSibling
  + rows
  + tagName
  + textContent
* The Javascript function console.log is very useful for debugging. It takes a string argument, which it prints to the Javascript log. You can display this log with "Tools -> JavaScript console" in the Chrome control menu. If you are having trouble figuring out what is happening in your Javascript, sprinkle console.log statements around your code so you can see which code is being executed.
* If nothing seems to be happening in your Javascript code, it's possible that your code contains an error that is causing it to be aborted. To find out if this is happening, open the Javascript console to see if there are any errors.

## Problem 2: Javascript Calendar (25 points)

Create a new directory project6 that will contain the solution for this project. Within that directory, create a file Calendar.js that implements a Javascript class named Calendar that can be used as in the following example:

var calendar = new Calendar("div1");

calendar.render(new Date("July 4, 1776"));

The constructor takes a single argument consisting of the id attribute for a div, and the render method takes one argument consisting of a Date object that selects a particular month (the object can refer to any time within the month). When render is invoked it replaces the contents of the calendar's div with HTML that displays a small one-month calendar such as those you might see in an airline reservation Web site:

* The calendar must display the days of the selected month in a grid with one line for each week and one column for each day of the week.
* Weeks run from Sunday on the left to Saturday on the right. The calendar must contain a header row displaying abbreviations for the days of the week, such as "Su", "Mo", etc.
* Each day of the month is displayed as a number.
* Some weeks may contain days not in the selected month. These days should be displayed as the number in their month, but in a dimmed fashion to indicate they are not part of the current month.
* The calendar must display the name of the month and year at the top of the calendar. In addition, it must display controls such as "<" and ">" that can be clicked to change the calendar's display to the previous or next month.

The calendar does not need to do anything if the user clicks on a particular date; all it needs to do is display one month and allow the month to be changed by clicking on the controls.

Once you have created the Javascript class, create an HTML file calendar.html that uses the Javascript class to display a calendar.