

Using SAS To Generate JavaScript For Dynamic Web Pages

Jim Hoffman – DBSI Corp.
SCSUG Conference October 6-8, 2002

Abstract:

This presentation outlines how to use SAS to generate JavaScript embedded within HTML. Example code will show how parallel JavaScript arrays and arrays within arrays can be used for dynamic web content. Sufficient code will be shown for the savvy user to be able to implement this concept.

Introduction:

Today's web pages need to be smarter. In order to "*user proof*" your web-based data entry; use Drop down selections as much as possible. Check Boxes and Radio Buttons are also excellent methods to limit user error. This presentation will outline a process that uses HTML to generate Drop down selection boxes, coupled with SAS generated JavaScript code that is embedded in your HTML. A description of a method called parallel arrays will be included. The process involves generating the arrays with valid values for a given variable. I am using an independent variable called State, and two dependent variables, one called City List and one called Widget List.

Process Overview:

Having two or more arrays that contain actual values and a final array that is an index to them creates a parallel array. The arrays containing the actual values to be displayed can be the same or different dimensions. The index array contains the beginning location of each of the valued arrays.

Process Detail:

The arrays are typically generated by SAS from an existing table of values. The code can be embedded completely in the HTML, which would assume that the entire web page was generated by SAS. For this presentation I have a static HTML page and use SAS to write the JavaScript array code to an external file. The code is introduced to the HTML via the `<script>` tag.

Example of a <script> tag for including external files

```
<script language="JavaScript"
  src="parallel_arrays.js"
  type="text/javascript">
</script>
```

This code is generated by SAS. The values for "State" are known, as is the order that they appear. What is unknown are the number of "Cities" and "Widget" values for each State. If the entire HTML code was generated by SAS, even the values for States could be generated dynamically from a table.

The arrays generated by SAS

```
var _a = new Array("Austin","Dallas","Houston","San
Antonio","El Paso","Corpus Christi");
var _b = new Array("Oklahoma City","Tulsa");
var _c = new Array("New Orleans","Monroe","Lafayette");
var _d = new Array("Albuquerque");
var _e = new Array("Little Rock","Fayetteville","Fort Smith");
var b1 = new Array("b-1","b-2","b-3","b-4","b-5","b-6","b-7");
var b2 = new Array("b-1");
var b3 = new Array("b-1","b-2","b-3","b-4");
var b4 = new Array("b-1","b-2","b-3","b-4");
var b5 = new Array("b-1","b-2","b-3","b-4");
var A = new Array(_a,_b,_c,_d,_e);
var Z = new Array(b1,b2,b3,b4,b5);
```

I have my HTML coded such that when the mouse crosses over the State value box or the value is changed, a JavaScript function is called to populate the values for City and Widget.

A subset of the JavaScript function to Populate the dependent variables

```
function Selection() {  
  
var colvar = document.myform.selection.options.selectedIndex;  
  
// Do the list box for the first one  
var listObj = document.myform.First_List;  
var newOpt;  
  
for (i in A[colvar]) {  
    var sel = A[colvar][i];  
    if (i < listObj.options.length) {  
        listObj.options[i].name=sel;  
        listObj.options[i].text=sel;  
    }  
    else {  
        newOpt = document.createElement("OPTION");  
        newOpt.text = sel;  
        newOpt.name = sel;  
        document.myform.First_List.add(newOpt);  
    }  
}  
newlen=i; newlen++;  
listObj.options.length = newlen;  
listObj.options.size = newlen;  
}
```

Acknowledgements:

Example code was retrieved from the JS-Examples archives <http://www.js-examples.com> where you will find hundreds of free ready to use scripts, tutorials, forums.

Author: JS-Examples - <http://www.js-examples.com/>

Programs/Comments:

A complete listing of all the programs is available for download at http://www.ccsi.com/~jshoffman/Parallel_Arrays.zip

For comments or questions please feel free to contact me via email at jshoffman@dbasicorp.com.