

Using Javascript in your Apex Applications

Josh Millinger, President
Niantic Systems, LLC

Speaker Qualifications:

- Josh Millinger, President, Niantic Systems, LLC
- CS degrees from UW-Madison, Johns Hopkins
- Former Oracle Sales Consultant and Founder of the Oracle Partner Technology Center
- 11+ Years of Oracle Web Development Experience
- Have Been Developing with and Teaching Apex Since Well Before It Was Even Released as a Product! Started with Excel Migration as first project

- Oracle Consulting with a Focus on Application Express
- Application Express Training
- Oracle Forms/Reports
- Discoverer
- Mentoring
- Customers in the Federal, Commercial, Healthcare, Higher Education, Construction verticals

What is Javascript?

- Interpreted language with Object Oriented capabilities
- Not JAVA!
- Quite robust
- In most cases, it runs in in a browser (client side)
 - However, it can be run on a Web Server such as IIS, Netscape

Javascript Basics

- Case Sensitivity
 - JavaScript is very case sensitive
 - Functions, keywords, variables
 - For example:
 - Amount, AMOUNT, & AmounT are all treated differently
- Comments
 - Anything on a line after //
 - Anything between /* and */

Javascript Basics

- Variables
 - Must begin with a letter, _ , or \$
 - Following characters can be letter, digit, _ , \$
- Reserved words
 - break, case, catch, continue, default, delete, do, else, false, finally, for, function, if, in, instanceof, new, null, return, switch, this, throw, true, try, typeof, var, void, while, with

Javascript Basics

- Strings
 - Can use single or double quotes
 - `'Hello NYOUG Attendees'`
 - `"Hello NYOUG Attendees"`
 - `'\n'` is the newline character
 - `'\'` is the escape character
 - `' I won\'t use .NET'`
 - Concatenate with +
 - `var t = 'This is an ' + 'example.'`

- `if (expression)`
 `{statement1}`
`else`
 `{statement2}`
- `if (name == null)` // == is the equality operator
- `{name = 'Larry';}` // = is the assignment operator

- switch

```
switch(n) {  
    case 'a' :  
        return 1;  
    case 'b' :  
        return 2;  
    default:  
        return 0;  
}
```

Javascript Statement - for

- For

```
For ( var n=0 ; n < 10 ; n++)  
{  
    document.write (n + "<br>") ;  
}
```

- Functions
 - Block of defined code that can be called more than once.
 - Can return value
 - Common convention: start with lowercase and then uppercase all subsequent words
 - Has zero or more args passed in (arg1, arg2,...argn)

```
function writeText ( txt)
{
    document.write (txt) ;
}
```

Basic Functions

- Alert

```
alert(msg);
```

```
alert("You must supply a value for  
Name");
```

- Confirm

```
confirm('Are you sure you want to delete this');
```

- Array Operators

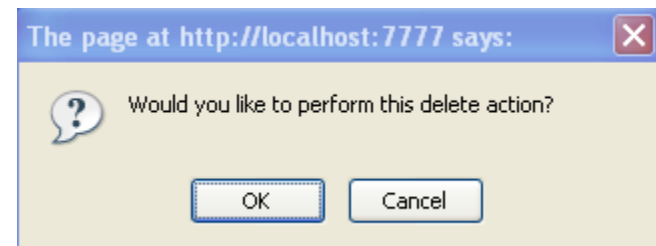
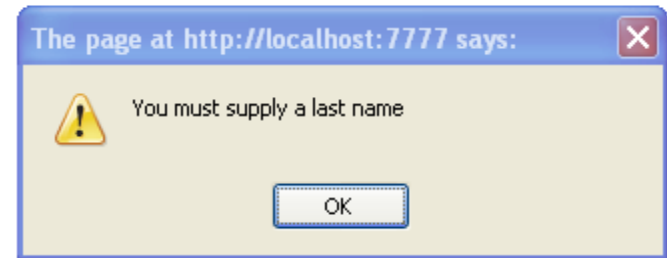
Join, Reverse, Sort, Concat

- String Operators

toLowerCase, concat, indexOf

- Math

abs, sin, min, pow, ceil



- Triggered when a particular action happens
- Very useful for capturing the timing of action and running appropriate script
- Examples:
 - When button is clicked
 - When select list value is changed
 - When focus leaves an item

Events

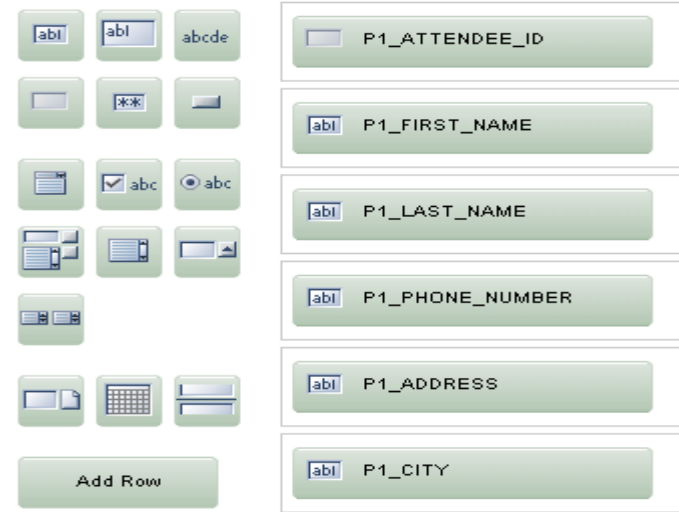
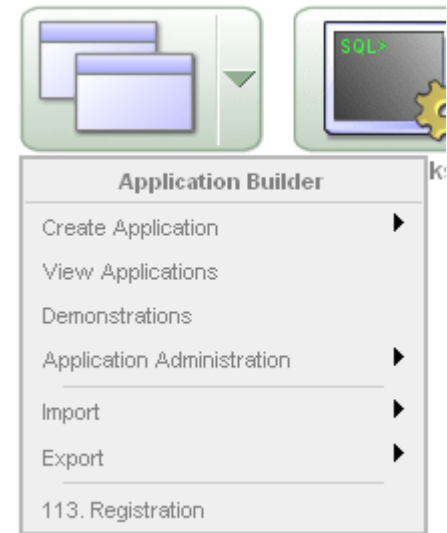
Attribute	The event occurs when...
<u>onabort</u>	Loading of an image is interrupted
<u>onblur</u>	An element loses focus
<u>onchange</u>	The user changes the content of a field
<u>onclick</u>	Mouse clicks an object
<u>ondblclick</u>	Mouse double-clicks an object
<u>onerror</u>	An error occurs when loading a document or an image
<u>onfocus</u>	An element gets focus
<u>onkeydown</u>	A keyboard key is pressed
<u>onkeypress</u>	A keyboard key is pressed or held down
<u>onkeyup</u>	A keyboard key is released
<u>onload</u>	A page or an image is finished loading
<u>onmousedown</u>	A mouse button is pressed
<u>onmousemove</u>	The mouse is moved
<u>onmouseout</u>	The mouse is moved off an element
<u>onmouseover</u>	The mouse is moved over an element
<u>onmouseup</u>	A mouse button is released
<u>onreset</u>	The reset button is clicked
<u>onresize</u>	A window or frame is resized
<u>onselect</u>	Text is selected
<u>onsubmit</u>	The submit button is clicked
<u>onunload</u>	The user exits the page

- Common syntax you will find in Apex applications

```
<script type="text/javascript">  
  function confirmDelete()  
  {  
    var x = confirm("Are you sure you wish to delete this  
    record?");  
    if (x)  
      doSubmit('DELETE');  
    else  
      return false;  
  }  
</script>
```

Apex and Javascript

- APEX wouldn't be APEX without Javascript
- Delete Confirmation
- Menus
- Query Builder
- Drag and Drop Layout



JavaScript Uses

- To accomplish tasks before page processing
 - Validation data
 - Compute new values
 - Confirmations/Alerts
 - Dynamically change look and feel of application through DHTML
 - Ajax for database communication
 - Overall, make the application more like a client/sever application
- Set values as page is loading

Validations

- Very useful to increase the interactive nature of applications
 - Must be a number or number < 100%
- Should always be used in conjunction with APEX validations.
- Javascript is easy to disable
 - Example is Firefox->Tools->Options->Content

Computations

- Javascript can compute values for items on the page
- Assign values through various methods
 - Static Text
 - `$x('P1_SEARCH').value= 'Jones' ;`
 - `Document.getElementById('REGION_ID').innerHTML= 'Jones' ;`
 - Arithmetic/String Functions
 - `Document.getElementById('P1_SEARCH').value=someString.toUpperCase() ;`
 - `$x('P1_SAL').value=someNumber+1.1 ;`

APEX JavaScript Files

- /i/javascript
 - apex_3_1.js
 - apex_get_3_1.js
 - plugins.js
 - apex_ns_3_1.js
 - htmldb_validate.js

```
function $x(pNd){
  try{
    var node;
    switch(typeof (pNd)){
      case 'string':node =
document.getElementById(pNd);break;
      case 'object':node = pNd;break;
      default:node = false;break;
    }
    if(node.nodeType == 1){return node;}else{return false;}
  }catch(e){return false;}
}
• * nodeType 1 is Element
```

Custom JavaScript Functions

- JavaScript Functions are collections of commands which execute when triggered to do so by an event
- Can be included in your APEX application in five places:
 - .js File
 - PL/SQL Region on Page Zero
 - Page Header
 - Page Region
 - Page Template
- Best approach is usually a combination of the above methods

- You can put all or some of your JavaScript functions in a static text file and then include that file as part of your page template

```
<script src="/c/myFunctions.js"  
type="text/javascript"></script>
```

- Benefits:
 - Easier to manage, as you can use a text editor with syntax highlighting
 - Browser will cache the JS file
- Drawbacks
 - Cannot dynamically build JavaScript code
 - Browser will cache the JS file
 - You have to remember to include the .js file when you promote your application to production

- Can call a procedure or be an anonymous PL/SQL block
- Allows you to refer to session state items and use them in your JavaScript
- Most flexible way to implement JavaScript in Apex

```
var pThis = 'P'+$v('pFlowStepId')+'_CAR_HAVE';
var pThat = 'P'+$v('pFlowStepId')+'_CAR_RIDERS';
$f_DisableOnValue(pThis,'N',pThat);
if ($v(pThis)=='N' || $v(pThis)=='')
{ $s(pThat,'');
  $x(pThat).disabled=true;
}
```

PL/SQL Region on Page Zero

- Benefits
 - You can dynamically control what code is generated and include session state variables
 - Good example is APP_ID
- Drawbacks
 - Each page view will execute the PL/SQL procedure
 - Which isn't really a bad thing, as it will be pinned in the SGA before long and take almost no time to execute
 - If you refer to a named PL/SQL procedure, you must include that procedure when you promote your application to production
 - write_javascript procedure

- Can call a JavaScript function when the page loads
- Contents are substituted in the Page Template #ONLOAD# token

On Load ↑

Page HTML Body Attribute

```
initPage ();
```

◆ Use this to add onload events such as calls to JavaScripts

- Can associate to an item
 - Elements > HTML Form Element Attributes
- Once the event occurs on the Item, it will execute

Element

Width Maximum Width Height

Horizontal / Vertical Alignment

HTML Table Cell Attributes


HTML Form Element Attributes


Form Element Option Attributes

Button & Column Links

- When a button or Link is clicked, it can either submit the page or redirect to a URL


Optional URL Redirect ↑


Target is a 

Page  (reset pagination for this page)

Request

Clear Cache (comma separated page numbers)

Set These Items  (comma separated name list)

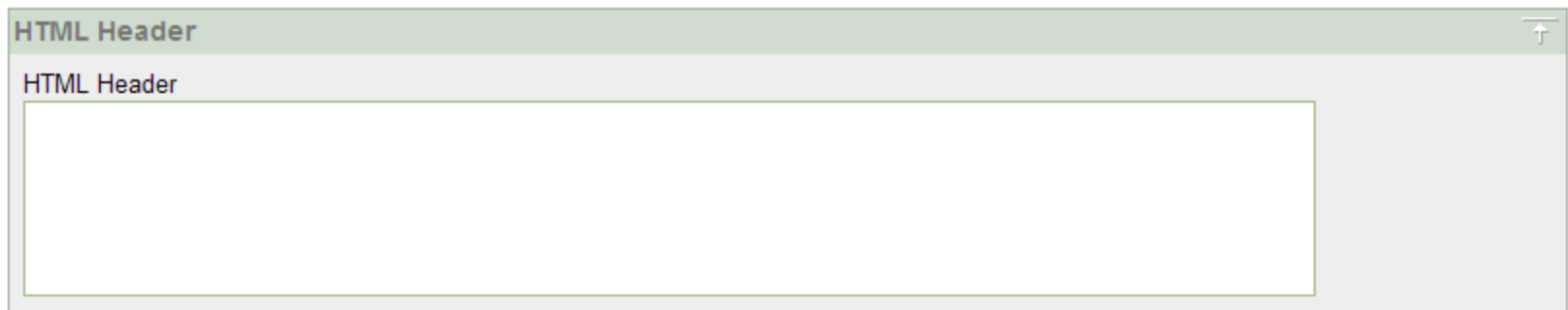
With These Values  (comma separated value list)

URL Target

```
javascript:confirmDelete(htmldb_delete_message, 'MULTI_ROW_DELETE');
```

Page Header

- You can also include JavaScript on the Page Header
- Only use for JavaScript which will be page-specific



A screenshot of a software interface for editing the HTML header. The window has a title bar that says "HTML Header" and a small upward-pointing arrow icon in the top right corner. Below the title bar, the text "HTML Header" is displayed above a large, empty rectangular text area for editing.

- You can also put JavaScript in a Page Region of type HTML
- Best Practices:
 - Put at the top of your page (Page Position 1, Sequence 1)
 - Set Template to **No Template**
- To share among more than one page, consider putting the Region on Page 0

Page Template

- Used when you have some static JavaScript that you want on every single page, and you don't want to include a .js file
- Drawbacks
 - Browser will never cache like it would a .js file, as each page is different
 - Hard to edit, as edit window in the web page is quite small

doSubmit()

- Part of the Apex JavaScript libraries
- Causes the page to be submitted
- Can also pass in the REQUEST parameter
 - `doSubmit (' SAVE ') ;`

Ajax in APEX

Ajax in Apex

- Ajax = Asynchronous JavaScript & XML
- Not a language, but rather an implementation of a suite of technologies which have been around for some time
- “Extends Client-Server like behavior to web applications”
- Apex 2.0 and later makes use of Ajax in the Application Builder & SQL Workshop
- You can take advantage of some of these Ajax libraries in your own applications

Ajax in Apex

- Web 1.0 Applications
 - Navigate to a Page with a Form
 - Fill out the Form
 - Click Submit
 - Page will process some server-side program and return the results back
- Still the safest way to build a web site!
 - Apex validations work this way

Ajax in Apex

- Web 2.0 Applications
 - Navigate to a Page with a Form
 - Fill out the Form
 - Get instantaneous feedback on validation errors, dynamically data-driven select lists, etc.
 - Submit the form only when data is correct
- While it may be cooler, this method introduces more “moving parts” which have to be adequately secured & managed
 - Web 2.0 forms should always have server-side validations

Firebug is your Friend

- Firebug

Employees

Emp#	Name	Job	Manager	Hiredate	Salary	Commission	Dept	Deleted	Add Field	Seqfield
	SMITH	CLERK	FORD	12/17/1980	\$800.00		RESEARCH	<input type="checkbox"/>	add field	<input type="text"/>
	ALLEN	SALESMAN	BLAKE	02/20/1981	\$1,600.00	\$300.00	SALES	<input type="checkbox"/>	add field	<input type="text"/>
	WARD	SALESMAN	BLAKE	02/22/1981	\$1,250.00	\$500.00	SALES	<input type="checkbox"/>	add field	<input type="text"/>
	JONES	MANAGER	KING	04/02/1981	\$2,975.00		RESEARCH	<input type="checkbox"/>	add field	<input type="text"/>
	MARTIN	SALESMAN	BLAKE	09/28/1981	\$1,250.00	\$1,400.00	SALES	<input type="checkbox"/>	add field	<input type="text"/>



The screenshot shows the Firebug console with the following JavaScript code:

```

28     $x_RowHighlight (ltr, '#CCCCFF' );)
29     else
30     {
31         //alert('turn off');
32         $x_RowHighlightOff (ltr);)
33     }
34 -->
35 </script>
36 <meta http-equiv="Content-Type" content="text/html; charset=windows-1252" />
37 </head>
38 <body ><form action="wwv_flow.accept" method="post" name="wwv_flow" id="wwvFlowForm">
39     <input type="hidden" name="p_flow_id" value="137" id="pFlowId" /> <input type="hidden" name="p_flow
40     <tr>
41         <td valign="top" class="t1Logo"></td>
42         <td valign="top" width="100%"></td>
43         <td align="right" valign="top"><div class="t1NavigationBar"><a href="wwv_flow_custom_auth_std.log

```

Apex 4.0

- Coming in Apex 4.0 is Dynamic Action
- JQuery
- Shown at ODTUG 2009
- Declaratively create javascript functionality from builder.
 - Hide Items
 - Change styles
 - Change values

- Using JavaScript in Apex Applications:
 - http://www.oracle.com/technology/products/database/htmldb/howtos/htmldb_javascript_howto2.html
- Javascript: The Definitive Guide, Third Edition
 - <http://www.oreilly.com/catalog/jscript3/index.html>

Thank You!

- If you're so inclined, send me questions & comments directly:
 - Josh Millinger, Niantic Systems, LLC
 - Phone: 609.945.3151
 - Email: jmillinger@nianticsystems.com

