



Emerging Technologies

From Basic HTML to Accessible Web 2.0 Using Dojo – Part 1

Becky Gibson
Web Accessibility Architect
Dojo Committer

Agenda

- Brief Introduction to Dojo
- Dojo Accessibility
- Review a Basic HTML Form
 - Progressive enhancement
 - Client Side Validation
- Enhancing with Dojo
 - Include Dojo
 - Specify a Theme
 - Including Dojo User Interface Elements - Dijit
 - Converting UI Elements to Dijit
 - Dijit Validation
- Review & Questions

Dojo - What is it?

- Open Source JavaScript Toolkit
- “Easy” Ajax
- Data Binding
- Full event system
- Browser abstraction layer
- User Interface Widgets
- Dual Licensed
 - Academic Free License v2.1
 - BSD License
- Dojo 1.0 available in November, 2007
 - 1.1 in March, 2008
- Dojo 1.2 released October, 2008
 - 1.2.3 in December, 2008
- Dojo 1.3 planned for Q1 2009



Core Widget Set - dijit

- Accessible
- Internationalized
- Customizable Look and Feel
- Developer Documentation
 - API
 - User Manual
- Supported Browsers
 - Firefox 2, Firefox 3
 - IE 6, IE 7, IE 8 support in progress
 - Safari 3
- Data Binding
 - Tree, Grid*, Select

*Grid is in dojox (dojo extensions)

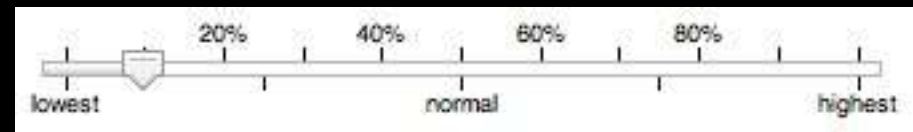
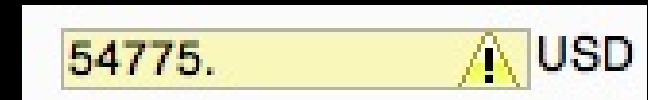
Dojo Core Widgets - Dijit

- Form Widgets
- Layout Widgets
- Advanced Widgets

Form Widgets

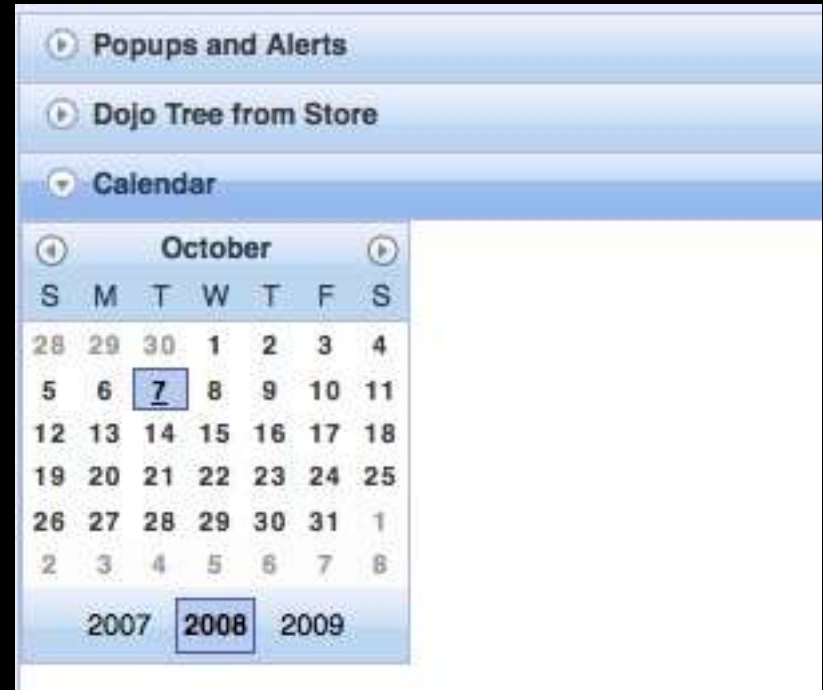


- Button, Dropdown Button, Combo Button
- Checkbox, Radio
- ComboBox, Filtering Select, Multi Select
- Textbox
- Currency & Integer Validation Textboxes
- Resizable Textarea
- Slider
- Integer Spinner
- Inline Editbox
- Dropdown Calendar



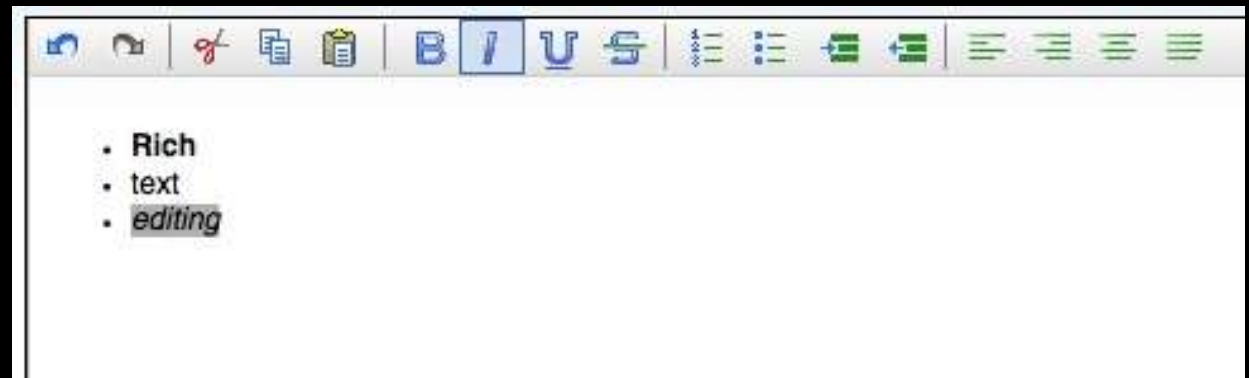
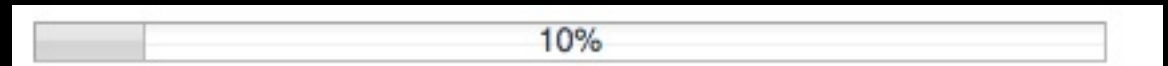
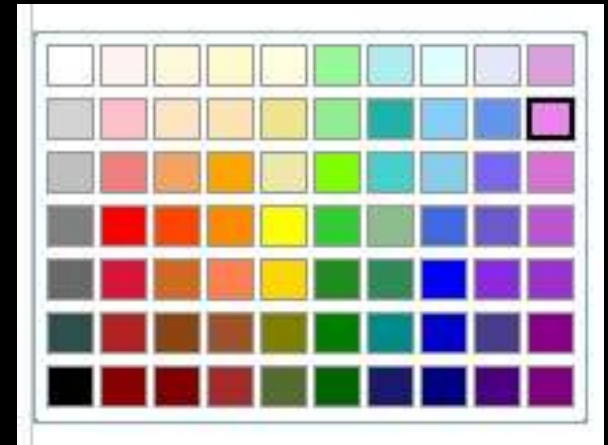
Layout Widgets

- Accordion Container
- Content Pane
- Dialog
- Border Container
- Tab Container
- Title Pane



Advanced Widgets

- Color Palette
- Context Menu
- Rich Text Editor
- Progress Bar
- Toolbar
- Tooltip, Tooltip Dialog
- Tree



Accessibility Support in Dojo Core Widgets - 1.0.x+

- Keyboard support in IE 6, IE 7, Firefox 2, Firefox 3
- Screen reader support in FF 2 & FF 3 with JAWS 9 or later
 - Implemented using Accessible Rich Internet Applications (ARIA)
- Low Vision Support
 - Windows high contrast support in IE 6, IE 7, FF2, FF3
 - No fixed font sizes
 - Support Images off in FF
- Low Vision support allows claim of “equivalent facilitation” for US Section 508 requirement to run with CSS turned off.

Starting with HTML

Becky's Setup

- MacBook Pro
- Eclipse with Aptana and Dojo 1.2 plugin
- XAMPP Apache Server with PHP
 - Configured to serve up CalWac directory
- Parallels Desktop to run Windows
- Firefox 3 in Windows
- JAWS 10 in Windows

Basic HTML Form – Scripting Off

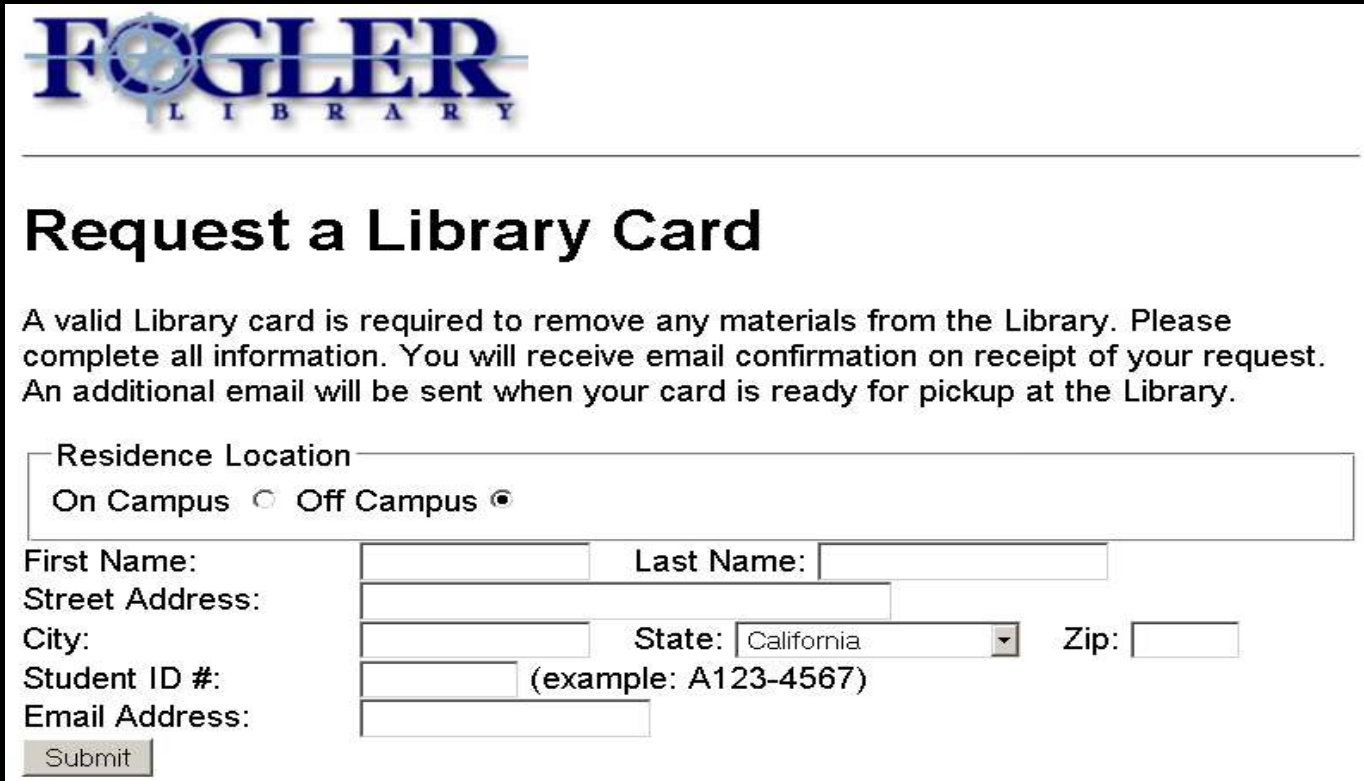


Request a Library Card

A valid Library card is required to remove any materials from the Library. Please complete all information. You will receive email confirmation on receipt of your request. An additional email will be sent when your card is ready for pickup at the Library.

First Name:	<input type="text"/>	Last Name:	<input type="text"/>		
Street Address:	<input type="text"/>				
City:	<input type="text"/>	State:	<input type="text" value="California"/>	Zip:	<input type="text"/>
Student ID #:	<input type="text"/> (example: A123-4567)				
Email Address:	<input type="text"/>				
<input type="button" value="Submit"/>					

Basic HTML Form – Scripting On



FOGLER
LIBRARY

Request a Library Card

A valid Library card is required to remove any materials from the Library. Please complete all information. You will receive email confirmation on receipt of your request. An additional email will be sent when your card is ready for pickup at the Library.

Residence Location

On Campus Off Campus

First Name: Last Name:

Street Address:

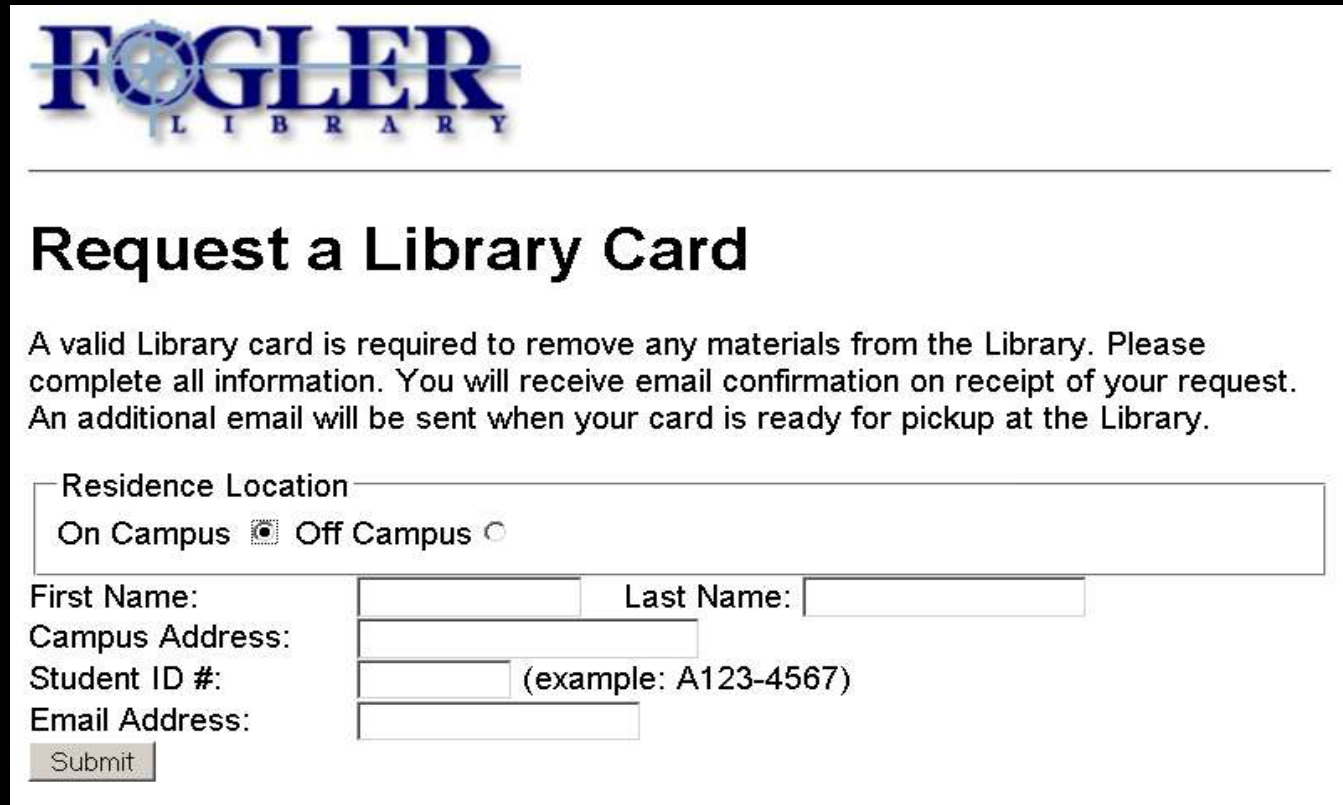
City: State: Zip:

Student ID #: (example: A123-4567)

Email Address:

RadioButton for selection of on or off campus location. Default is off campus and street, city, state and zip fields are displayed.

Basic HTML Form – Scripting On



FOGLER
LIBRARY

Request a Library Card

A valid Library card is required to remove any materials from the Library. Please complete all information. You will receive email confirmation on receipt of your request. An additional email will be sent when your card is ready for pickup at the Library.

Residence Location

On Campus Off Campus

First Name: Last Name:

Campus Address:

Student ID #: (example: A123-4567)

Email Address:

On Campus radioButton is selected and Campus Address field is displayed.

Using Progressive Enhancement

RadioButtons are within a div styled with display:none

```
<div id="location" style="display:none">
<fieldset>
<legend>Residence Location</legend>
<label for="on">On Campus </label><input
type="radio" id="on" name="resides" value="true"
onchange="setResidenceType(this);">
<label for="off">Off Campus</label><input
type="radio" id="off" name="resides" value="false"
onchange="setResidenceType(this);" checked>
</fieldset>
</div>
```


Display Radio Buttons if Scripting Enabled

RadioButtons are displayed on page load if scripting is enabled

```
<script type="text/javascript">
function setup() {
    var loc = document.getElementById('location');
    loc.style.display="block";
    ....
}
</script>
</head>
<body onload="setup();">
.....
```


RadioButton onclick to display appropriate fields

```
function setResidenceType(theRadio) {
    var on = document.getElementById('onCampus');
    var off = document.getElementById('offCampus');
    var show, hide;
    if (theRadio.id == "on" && theRadio.checked) {
        show = on; // show onCampus
        hide = off;
    } else if (theRadio.id == "off" && theRadio.checked) {
        show = off;
        hide = on;
    }
    show.style.display="block";
    hide.style.display="none";
...}.
<input type="radio" id="on" name="resides" value="true"
onclick="setResidenceType(this);">
<input type="radio" id="off" name="resides" value="false"
onclick="setResidenceType(this);" checked>
```

Client Side Validation of Form

Add validation on page load if scripting is enabled

```
<script type="text/javascript">
function setup() {
var loc = document.getElementById('location');
loc.style.display="block";
document.getElementById('cardReq').onsubmit=validate;
}
</script>
```

Client Side Validation Script

```
function validate() {
var message="Please correct the following fields:"
var isClean = true;
var bContinue = true;
var theForm = document.getElementById('cardReq');
var test;
for (var i=0; i<theForm.length; i++){
field = theForm.elements[i];
switch (field.id) {
case 'fname':
case 'lname':
case 'street':
case 'city':
case 'campusAddr':
var labels = document.getElementsByTagName('label');
if (field.clientHeight > 0 && field.value == ""){
for (var j = 0; j < labels.length; j++) {
if (labels[j].htmlFor == field.id) {
// a bit dangerous assuming the first node of label is the one I want.....
message += '\n' + labels[j].firstChild.nodeValue;
break;
}
}
isClean = false;
}
break;
}
```

continued..

Client Side Validation Script (continued)

```
case 'studentId':
    //var idExp = new RegExp("[A-Z]+\d{3}+\-\+\d{4}", "g");
    var idExp = /^[A-Z][0-9]{3}\-[0-9]{4}$/;
    if (!idExp.test(field.value)) {
        message += '\n' + "Student ID #:";
        isClean = false;
    }
break;
case 'email':
    // not well tested obtained from http://www.regular-expressions.info/email.html
    var testExp= new RegExp("^[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}$", "i");
    if (!testExp.test(field.value)){
        message += '\n' + "Email:";
        isClean = false;
    }
break;
case 'zip':
    var zipExp = /^[0-9]{5}$/;
    if (!zipExp.test(field.value)){
        message += '\n' + "Zip:";
        isClean = false;
    }
break;
} // end switch
} // end for loop
if (isClean == false) {
    alert(message);
}
return isClean;
}
```

Testing with JAWS 10 and Firefox 3

Move to Web 2.0 Using Dojo

- Improve Look and Feel
 - Prompts about data entry format
 - Visual error indication
 - Error messages
- Improve Accessibility
- Simplify Validation
- Learn Dojo and Dijit Basics

Move to Web 2.0 with Dojo

- Get Dojo
- Add Dojo
- Add Dijit Theme
- Use `Dojo.addOnLoad()`
- Add Dijit Requires
- Use `dijit.form.RadioButton`
- Use `dijit.form.TextBox`
- Use `dijit.form.FilteringSelect`
- Use `dijit.form.ValidationTextBox`
- Use `dijit.form.Form`

Move to Web 2.0 with Dojo

- **Get Dojo**
- Add Dojo
- Add Dijit Theme
- Use `Dojo.addOnLoad()`
- Add Dijit Requires
- Use `dijit.form.RadioButton`
- Use `dijit.form.TextBox`
- Use `dijit.form.FilteringSelect`
- Use `dijit.form.ValidationTextBox`
- Use `dijit.form.Form`

Get Dojo

- Install Dojo on your server
 - Download from <http://dojotoolkit.org/downloads>
 - Once development is complete consider creating your own optimized build <http://dojocampus.org/content/tag/build/>
- Use Dojo from your IDE
 - Example: Eclipse with Aptana plugin
from Apanta page select create new project link
adds projectName/lib/dojo/dojo
 projectName/lib/dojo/dijit
 projectName/lib/dojo/dojox
- Use Dojo from CDN
 - AOL: `<SCRIPT TYPE="text/javascript" SRC="http://o.aolcdn.com/dojo/1.2/dojo/dojo.xd.js"></SCRIPT>`
 - Google: `<SCRIPT TYPE="text/javascript" SRC="http://ajax.googleapis.com/ajax/libs/dojo/1.2/dojo/dojo.xd.js"></SCRIPT>`

Move to Web 2.0 with Dojo

- Get Dojo
- **Add Dojo**
- Add Dijit Theme
- Use `Dojo.addOnLoad()`
- Add Dijit Requires
- Use `dijit.form.RadioButton`
- Use `dijit.form.TextBox`
- Use `dijit.form.FilteringSelect`
- Use `dijit.form.ValidationTextBox`
- Use `dijit.form.Form`

Adding Dojo

```
<style type="text/css">  
    @import "lib/dojo/dojo/resources/dojo.css";  
    @import "resources/calwac.css";  
</style>  
  
<!-- required: dojo.js -->  
<script type="text/javascript" src="./lib/dojo/dojo/dojo.js"  
    djConfig="isDebug: true, parseOnLoad: true">  
</script>
```

Useful Dojo Methods

- `dojo.require()` - add dojo and dijit resources to your page
- `dojo.addOnLoad()` - defers script execution until all the HTML is loaded
- `dojo.byId()` - shortcut to `document.getElementById()`
- `dijit.byId()` - get a reference to a dijit widget by id
- `dojo.connect()` - connecting event handlers
- `dojo.attr()` - standard mechanism for getting/setting attributes on a node

Move to Web 2.0 with Dojo

- Get Dojo
- Add Dojo
- **Add Dijit Theme**
- Use `Dojo.addOnLoad()`
- Add Dijit Requires
- Use `dijit.form.RadioButton`
- Use `dijit.form.TextBox`
- Use `dijit.form.FilteringSelect`
- Use `dijit.form.ValidationTextBox`
- Use `dijit.form.Form`

Dijit Themes

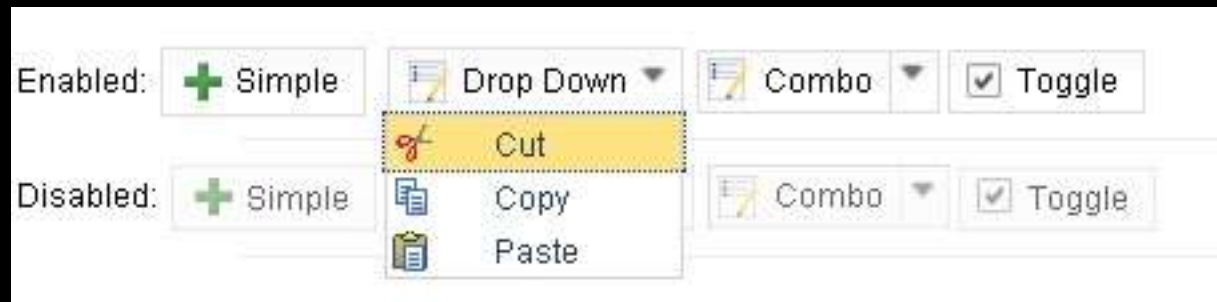
tundra



soria



nihilo



Add Dijit Theme

```
<style type="text/css">  
    @import "lib/dojo/dojo/resources/dojo.css";  
/* ADD DIJIT THEME */  
    @import "lib/dojo/dijit/themes/soria/soria.css";  
    @import "resources/calwac.css";  
  
</style>
```

Move to Web 2.0 with Dojo

- Get Dojo
- Add Dojo
- Add Dijit Theme
- Use `Dojo.addOnLoad()`
- Add Dijit Requires
- Use `dijit.form.RadioButton`
- Use `dijit.form.TextBox`
- Use `dijit.form.FilteringSelect`
- Use `dijit.form.ValidationTextBox`
- Use `dijit.form.Form`

Using dojo.addOnLoad()

From:

```
function setup() {
    var loc = document.getElementById('location');
    loc.style.display="block";
    document.getElementById('cardReq').onsubmit=validate;
}
.....
<body onload="setup();" >
```

To:

```
dojo.addOnLoad(
    function() {
        var loc = document.getElementById('location');
        loc.style.display="block";
        dojo.byId('cardReq').onsubmit=validate;
    })
.....
<body class="soria" >
```

Move to Web 2.0 with Dojo

- Get Dojo
- Add Dojo
- Add Dijit Theme
- Use `Dojo.addOnLoad()`
- **Add Dijit Requires**
- Use `dijit.form.RadioButton`
- Use `dijit.form.TextBox`
- Use `dijit.form.FilteringSelect`
- Use `dijit.form.ValidationTextBox`
- Use `dijit.form.Form`

Add First Dijit Requires

```
<script type="text/javascript">  
    dojo.require("dijit.form.CheckBox");  
    dojo.require("dijit.form.TextBox");  
    // scan page for widgets and instantiate them  
    dojo.require("dojo.parser");  
</script>
```

Move to Web 2.0 with Dojo

- Get Dojo
- Add Dojo
- Add Dijit Theme
- Use `Dojo.addOnLoad()`
- Add Dijit Requires
- Use `dijit.form.RadioButton`
- Use `dijit.form.TextBox`
- Use `dijit.form.FilteringSelect`
- Use `dijit.form.ValidationTextBox`
- Use `dijit.form.Form`

Use dijit.form.RadioButton

From:

```
<fieldset>
<legend>Residence Location</legend>
<label for="on">On Campus </label><input type="radio"
id="on" name="resides" value="true"
onclick="setResidenceType(this);">
<label for="off">Off Campus</label><input type="radio"
id="off" name="resides" value="false"
onclick="setResidenceType(this);" checked>
</fieldset>
```

To:

```
<fieldset>
<legend>Residence Location</legend>
<label for="on">On Campus </label><input type="radio"
dojoType="dijit.form.RadioButton" id="on" name="resides" value="true"
onclick="setResidenceType(this);">
<label for="off">Off Campus</label><input type="radio"
dojoType="dijit.form.RadioButton" id="off" name="resides"
value="false" checked onclick="setResidenceType(this);">
</fieldset>
```

Move to Web 2.0 with Dojo

- Get Dojo
- Add Dojo
- Add Dijit Theme
- Use `Dojo.addOnLoad()`
- Add Dijit Requires
- Use `dijit.form.RadioButton`
- Use `dijit.form.TextBox`
- Use `dijit.form.FilteringSelect`
- Use `dijit.form.ValidationTextBox`
- Use `dijit.form.Form`

dijit.form.TextBox

- Basic `<input type="text">` form control
- Has rudimentary text formatting
 - Trim
 - Propercase
- Sized via style attribute

Use dijit.formTextBox

From:

```
<label for="fname" class="leftLabel">First Name: </label><input  
id="fname" name="fname" type="text" size="15">&nbsp; &nbsp;
```

To:

```
<label for="fname" class="leftLabel">First Name: </label><input  
dojoType="dijit.form.TextBox" trim="true" propercase="true"  
id="fname" name="fname" type="text" style="width: 10em;">
```


Move to Web 2.0 with Dojo

- Get Dojo
- Add Dojo
- Add Dijit Theme
- Use `Dojo.addOnLoad()`
- Add Dijit Requires
- Use `dijit.form.RadioButton`
- Use `dijit.form.TextBox`
- Use `dijit.form.FilteringSelect`
- Use `dijit.form.ValidationTextBox`
- Use `dijit.form.Form`

Use dijit.form.FilteringSelect

- Like an HTML Select
- Type ahead filtering to modify the options displayed
- Only valid selections allowed (dijit.form.ComboBox allows any value)
- Can be used with a dojo data store (*data stores will be briefly discussed in part 2*)
- Size using style attribute on widget

Switch to dijit.form.FilteringSelect

From:

```
<select id="state" name="state">  
  <option value="AL">Alabama</option>  
  <option value="AK">Alaska</option>  
  <option value="AZ">Arizona</option>  
  <option value="AR">Arkansas</option>  
  <option value="CA" selected>California</option>
```

To:

```
<select id="state" name="state" dojoType="dijit.form.FilteringSelect">  
  <option value="AL">Alabama</option>  
  <option value="AK">Alaska</option>  
  <option value="AZ">Arizona</option>  
  <option value="AR">Arkansas</option>  
  <option value="CA" selected>California</option>
```

Move to Web 2.0 with Dojo

- Get Dojo
- Add Dojo
- Add Dijit Theme
- Use `Dojo.addOnLoad()`
- Add Dijit Requires
- Use `dijit.form.RadioButton`
- Use `dijit.form.TextBox`
- Use `dijit.form.FilteringSelect`
- Use `dijit.form.ValidationTextBox`
- Use `dijit.form.Form`

dijit.form.ValidationTextBox

- Checks to make sure user input is valid according to specified criterion
 - Required attribute
 - Sets ARIA required attribute
 - Regular expression
- On error will
 - Display error icon
 - Change background color
 - Set ARIA invalid property
 - Display error message via tooltip (if provided)
- Displays prompt message (if provided)

StudentID field as dijit.form.ValidationTextBox

From:

```
<input name="studentId" id="studentId" type="text"
size="9"> (example: A123-4567)
```

Plus code in validate() function

To:

```
<input name="studentId" id="studentId" type="text"
style="width:10em;"
dojoType="dijit.form.ValidationTextBox"
regExp="^[A-Z]\d{3}\-\d{4}$"
required="true"
invalidMessage="Invalid Student ID - A123-1234"
promptMessage="A123-1234" >
```

Move to Web 2.0 with Dojo

- Get Dojo
- Add Dojo
- Add Dijit Theme
- Use `Dojo.addOnLoad()`
- Add Dijit Requires
- Use `dijit.form.RadioButton`
- Use `dijit.form.TextBox`
- Use `dijit.form.FilteringSelect`
- Use `dijit.form.ValidationTextBox`
- Use `dijit.form.Form`

Use dijit.form.Form

- Perform validation of contained widgets
- Convert form to object or json for Ajax submittal

Using dijit.form.Form

- Update dijit.form.TextBox fields to dijit.form.ValidationTextBox and add required attribute
 - First Name, Last Name, Street, City – required = “true”
 - Campus Address – required = false since it is not shown on page load
 - Update setResidenceType() to update required attribute as hide (required=false) and show (required=true) fields
- Remove original validate() function since it is no longer needed
- Add reset button

```
<button type="reset">Reset</button>
```
- Add form event handlers for onsubmit and onreset functions
- Advanced – use Ajax and server functions to evaluate that the City is valid for the State selected (*not implemented in demo*)

Modify form control to dijit.form.Form

```
<form id="cardReq" encType="multipart/form-data" action="fakeSubmit.php"
method="post" dojoType="dijit.form.Form" >
  <script type="dojo/method" event="onSubmit">
    var bSubmit = true;
    if(this.validate()){
      bSubmit = confirm('Form is valid, press OK to submit');
    } else {
      alert('Form contains invalid data. Please correct!');
      bSubmit = false;
    }
    return bSubmit;
  </script>
  <script type="dojo/method" event="onReset">
    var bReset = confirm('reset Form?');
    if (bReset){ // reset back to off campus address
      dojo.byId('off').checked = true;
      setResidenceType(dojo.byId('off'));
    }
    return bReset;
  </script>
```

Update setResidenceType() Function

```
function setResidenceType(theRadio) {
    var on = document.getElementById('onCampus');
    var off = document.getElementById('offCampus');
    var show, hide;
    if (theRadio.id == "on" && theRadio.checked) { // show onCampus
        show = on;
        hide = off;
        dijit.byId('street').attr("required", false);
        dijit.byId("city").attr("required", false);
        dijit.byId("zip").attr("required", false);
        dijit.byId("state").attr("required", false);
        dijit.byId("campusAddr").attr("required", true);
    } else if (theRadio.id == "off" && theRadio.checked) {
        show = off;
        hide = on;
        dijit.byId('street').attr("required", true);
        dijit.byId("city").attr("required", true);
        dijit.byId("zip").attr("required", true);
        dijit.byId("state").attr("required", true);
        dijit.byId("campusAddr").attr("required", false);
    }
    show.style.display="block";
    hide.style.display="none";
}
```

Review - Move to Web 2.0 with Dojo

- Get Dojo
- Add Dojo
- Add Dijit Theme
- Use `Dojo.addOnLoad()`
- Add Dijit Requires
- Use `dijit.form.RadioButton`
- Use `dijit.form.TextBox`
- Use `dijit.form.FilteringSelect`
- Use `dijit.form.ValidationTextBox`
- Use `dijit.form.Form`

Using our Form with JAWS 10 and Firefox 3

Simple Dojo Enhancements

- Simple, Immediate Validation of input
 - Visual icon
 - Background Color Change
 - Error Message
 - ARIA Support
- Prompt Message Accessible Tooltip to assist with input format
- Type-ahead Filtering Select for faster selection
- Validation of regular expression w/o additional Scripting

Dojo Features Used

- `dojo.js`
- `dojo.require()`
- `dojo.byId()`
- `dojo.addOnLoad()`
- Dijit Widgets
 - `dijit.form.ValidationTextBox`
 - `dijit.form.RadioButton`
 - `dijit.form.Button`
 - `dijit.form.FilteringSelect`
 - `dijit.form.Form`
 - `<script type="dojo/method" >`

Additional Dojo Features to Consider

- Field validation via Ajax and server side methods
 - Is city valid for selected state?
 - Is zip valid for selected city
 - Is student ID in good standing?
- Use of additional dijit widgets
 - dijit.form.DateTextBox for birthdate
 - dijit.form.NumberSpinner for year of graduation

Summary

- Dojo can enhance your HTML pages
- Dojo is easy!
- Use as much or as little of Dojo as you need
- Progressively include more Dojo features
- Move to Web 2.0 today with Dojo!



Resources

- Becky's Accessibility Presentations and Papers
 - <http://www.weba11y.com/Presentations/presentations.html>
- ARIA Roadmap, Best Practices, Primer, Specifications
 - <http://www.w3.org/wai/pf>
- Mozilla Developer Center - Firefox ARIA Information
 - http://developer.mozilla.org/en/docs/Accessible_DHTML
- Mailing list for ARIA issues
 - <http://lists.w3.org/Archives/Public/wai-xtech/>
- Dojo
 - Home - <http://www.dojotoolkit.org/>
 - Dojo Campus - <http://dojocampus.org>
 - Book - <http://www.dojotoolkit.org/book/dojo-book-1-0>
 - API Reference - <http://api.dojotoolkit.org>
 - 1.3 Book (in progress) - <http://docs.dojocampus.org>
- DHTML Style Guide Draft
 - http://dev.aol.com/dhtml_style_guide

Dojo Books in Print

- Dojo The Definitive Guide by Matthew Russel, O'Reilly Press
- Dojo Using the Dojo JavaScript Library to Build Ajax Applications by James E. Harmon, Addison-Wesley Developer's Library
- Mastering Dojo by Rawld Gill, Craig Riecke, Alex Russel, Pragmatic Press

