Dojo Widgets (Dijit)

Topics

- What are Widgets, What is Dijit?
- How to use widgets
  - Declaratively or programmatically
- Form widgets
  - Input form validation
- Layout widgets
- Command control widgets
- User assistance and feedback widgets
- Advanced editing and display widgets
- Themes and Design

What is a Widget?
What is Dijit?

What is a Widget?

- Is a UI element such as a button, text box, scroll bar, calendar, tree etc
  - Widgets are a lot easier to deal with than DOM APIs
- Can be a composite element
- Can have custom style
- Event handlers can be registered
  - Event handlers might call backend service
- Can handle browser incompatibilities

What is Dijit?

- Dojo Widget Library
  - A widget system layered on top of dojo.
- Fully theme'able
  - Come with three default themes
  - You can override the theme by container or by element to add nuance and flair.
- Fully internationalized
- Fully accessible

Widgets in Dijit

- Form, Validation, Specialized Input
  - > CheckBox, ComboBox, DateTextBox, InlineEditBox, Slider, Textarea, TextBox, ValidationTextBox, etc
- Layout
  - > AccordionContainer, Border Container, ContentPane, etc.
- Command control
  - > Button, DropDownButton, ComboButton, Menu, Toolbar
- User assistance and feedback
  - > Dialog, TooltipDialog, PregressBar, TitlePane, Tooltip
- Advanced editing and display
  - > Editor, Grid, Tree

How to Use Widgets

How to Use Widgets

- Declaratively or Programmatically
  - Declaratively by using special attributes inside of regular HTML tags (markup)
  - Programmatically through JavaScript.
- You have the same options either way

Using Widgets Declaratively (Markup)

- Be sure to `dojo.require` your widget.
- Be sure to include the Dojo parser!
- Widgets are created as soon as the DOM is created (if possible) before any other `addOnLoad` callbacks.
- Create your markup, and add the following attribute to the main node:

```
<div dojoType=”path.to.widget”></div>
```

...where “*path.to.widget*” is the fully qualified name of the widget constructor

Example: Creating a Progress Bar Declaratively

```
<head>
  <script src=”dojotoolkit/dojo/dojo.js”
    djConfig=”parseOnLoad: true”>
  </script>
  <script>
    dojo.require(“dojo.parser”);
    dojo.require(“dijit.ProgressBar”);
  </script>
</head>
<body>
  <div style="width: 400px;" maximum="200" id="setTestBar"
    progress="20" dojoType="dijit.ProgressBar">
  </div>
</body>
```

Creating Widgets Programatically

- DOM should be available before creating widgets programmatically.
  - Unless you pass it a new parentless node (or no node)
  - **`dojo.addOnLoad`**
- Call the widget constructor directly:

```
var myWidget = new dijit.form.Button({
    // properties
    iconClass:”someCSSclass”
}, “myButtonNode”);
```

Constructor Parameters

- All widget constructors take two arguments:
  - A properties object
  - A node reference
- Node reference is optional, but is considered good practice.
  - A `div` will be automatically created.
- Properties object is a set of named arguments you can use to set various properties on a widget.

Example: Creating a Progress Bar Programmatically

```
<head>
  <script src=”dojotoolkit/dojo/dojo.js”></script>
  <script>
    dojo.require(“dijit.ProgressBar”);

    dojo.addOnLoad(function(){
      var instance = new dijit.ProgressBar({
        maximum: 200,
        progress: 20
      }, “setTestBar”);
    });
  </script>
</head>
<body>
  <div style="width: 400px;" id="setTestBar”></div>
</body>
```

Dijit Form Widgets

Form Widgets – Common Characteristics

- All widgets beginning with “dijit.form”
- The form widgets can be used in a FORM tag, in a dijit.form.Form widget, or outside of a form.

Form Widgets – Common Characteristics

- Common Attributes
  - disabled, intermediateChange, tabIndex
- Common Methods
  - focus – set the focus on this widget
  - getValue – get the value of the widget
  - setValue – set the value of the widget
  - reset – resets the widget to its initial value
  - undo – restore the value to the last value passed to onChange
  - setAttribute - Controls all sorts of attributes for widgets like disabled, readonly, etc.
- Common Extension Points
  - onChange – callback when value is changed

Form Widget – convenience methods

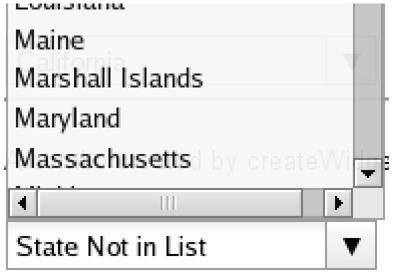
- Convenience methods added to regular HTML form
  - *getValues* - generate JSON structure from form values get widget values
  - *setValues* - fill in form values from a JSON structure generate map from name --> [list of widgets with that name]
  - *isValid* - Return true if every widget's isValid method returns true.
  - *submit* - programmatically submit form

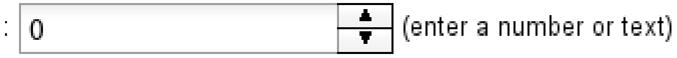
Form Widget – Extension Points

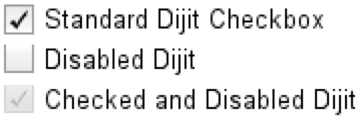
- Extension points
  - execute - User defined function to do stuff when the user hits the submit button

Form Widgets

- CheckBox, RadioButton, ComboBox, DateTextBox, FilterSelect, NumberSpinner, NumberTextBox, Slider, TextArea, TextBox, etc

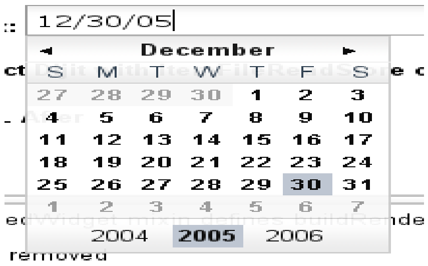


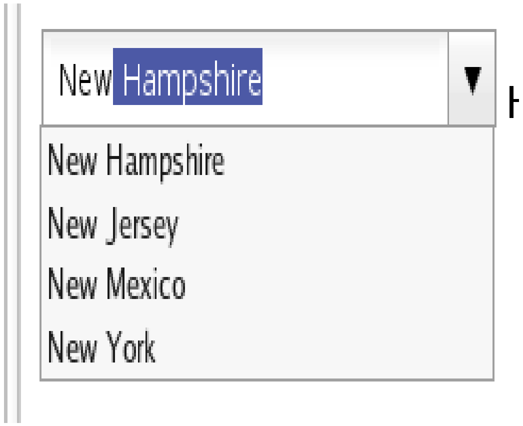


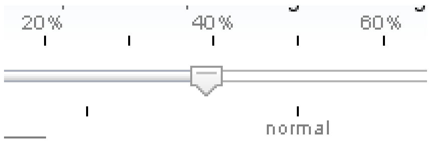

☑ Standard Dijit Checkbox

□ Disabled Dijit

☑ Checked and Disabled Dijit

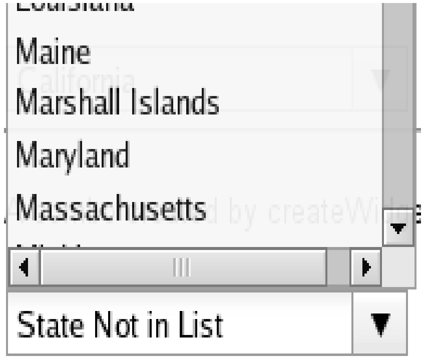






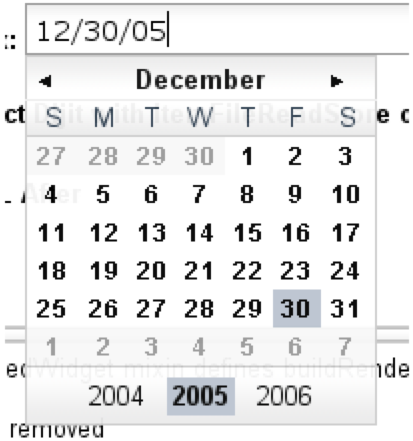
dijit.form.ComboBox Example

```
<head>
   ...
   <script type="text/javascript">
      dojo.require("dijit.form.ComboBox");
      function setVal1(value) {
         alert("Selected "+value);
      }
   </script>
</head>

<body class="tundra">
   <select name="state1"
         dojoType="dijit.form.ComboBox"
         autocomplete="true"
         value="California"
         onChange="setVal1">
         <option selected="selected">California</option>
         <option>Connecticut</option>
         <option >Illinois</option>
         <option >New York</option>
   </select>
</body>
```



dijit.form.DateTextBox Example

```
<head>
   ...
   <script type="text/javascript">
      dojo.require("dijit.form.DateTextBox");
   </script>
</head>

<body class="tundra">
   <h1>DateTextBox</h1>
   <input type="text" name="date1" value="2009-09-30"
         dojoType="dijit.form.DateTextBox"
         required="true" />
</body>
```



12/30/05

◂ December ▸

S	M	T	W	T	F	S
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

2004 **2005** 2006

Validation

Example: Validation

```
dojo.require("dijit.form.ValidationTextBox");

<input type="text" name="phone"
       id="phone"
       value="someTestString"
       dojoType="dijit.form.ValidationTextBox"
       regExp="[\w]+"
       required="true"
       invalidMessage="Invalid Non-Space Text.">
```

Demo:

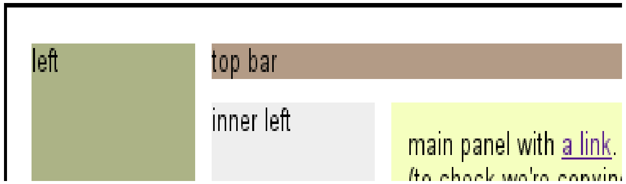
Input Form Validation

http://dojocampus.org/explorer/#Dijit_Form%20Controls_Text%20Boxes_Validation

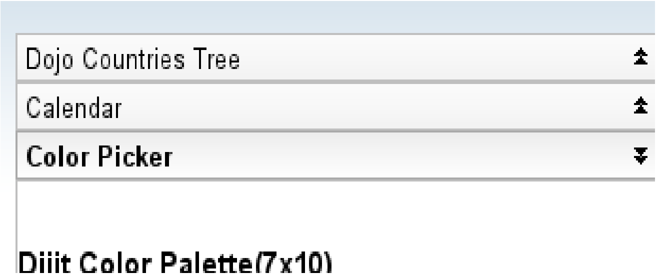
Dijit Layout Widgets

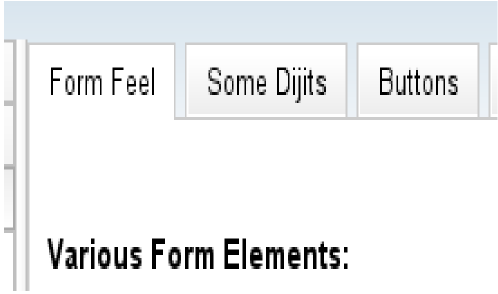
Dijit Layout Widgets

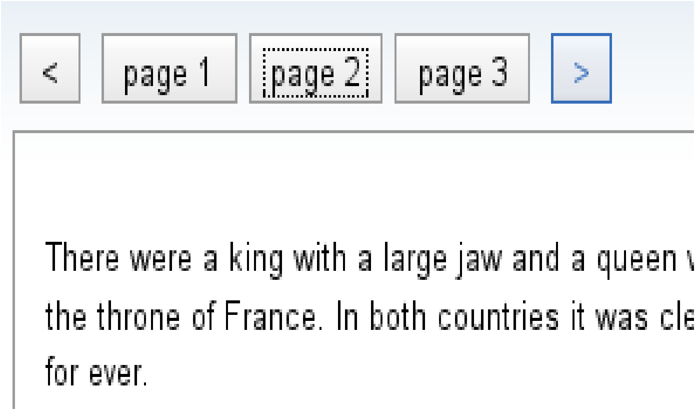
- Accordion Container, Content Pane, Layout Container, Split Container, Stack Container, Tab Container



main panel with a link.
(to check we're convin







There were a king with a large jaw and a queen w
the throne of France. In both countries it was cle
for ever.

dijit.layout.SplitContainer Example

```
<script type="text/javascript">
  dojo.require("dijit.layout.SplitContainer");
  dojo.require("dijit.layout.ContentPane");
</script>

<div dojoType="dijit.layout.SplitContainer"
  orientation="horizontal"
  sizerWidth="7"
  activeSizing="false"
  style="border: 1px solid #bfbfbf; float: left; width: 400px; height: 300px;">
  <div dojoType="dijit.layout.ContentPane" sizeMin="20" sizeShare="20">
    this box has two horizontal panes
  </div>
  <div dojoType="dijit.layout.ContentPane" sizeMin="50" sizeShare="50">
    without active resizing, a smaller sizer, different starting sizes and minimum sizes
  </div>
</div>
```

Demo:

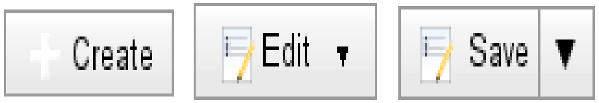
Layout Widgets

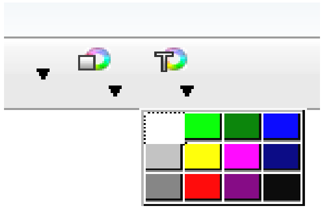
http://dojocampus.org/explorer/#Dijit_Layout

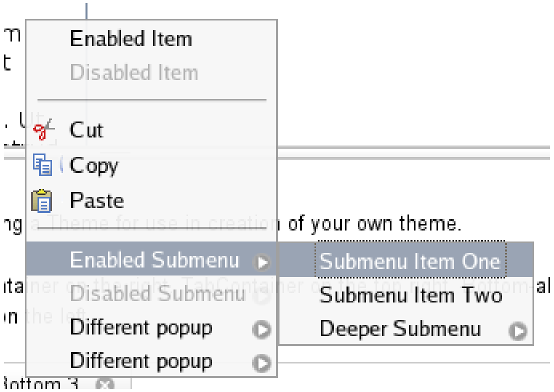
Dijit Command Control Widgets

Dijit Command Control Widgets

- Button, ComboButton, DropDownButton
- Menu, Toolbar







dijit.Menu Example

```
<script language="JavaScript" type="text/javascript">
  dojo.require("dijit.Menu");
  dojo.require("dojo.parser");  // scan page for widgets and instantiate them
  function doNothing() {
    alert('not actually doing anything, just a test!');
  }
</script>

<div dojoType="dijit.Menu" id="popup1campus" contextMenuForWindow="false" style="display: none;"
targetNodeIds="btn1">
  <div dojoType="dijit.MenuItem" onClick="alert('Hello world');">Enabled Item</div>
  <div dojoType="dijit.MenuItem" disabled="true">Disabled Item</div>
  <div dojoType="dijit.MenuSeparator"></div>
  <div dojoType="dijit.MenuItem" iconClass="dijitEditorIcon dijitEditorIconCut"
    onClick="doNothing();">Cut</div>
  <div dojoType="dijit.MenuItem" iconClass="dijitEditorIcon dijitEditorIconCopy"
    onClick="doNothing();">Copy</div>
  <div dojoType="dijit.MenuItem" iconClass="dijitEditorIcon dijitEditorIconPaste"
    onClick="doNothing();">Paste</div>
</div>

<div id="btn1">Right click Me To Show Menu</div>
```

Demo:

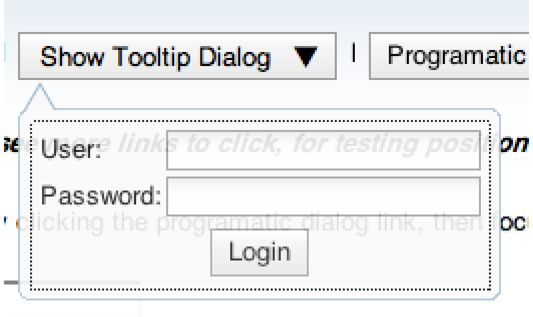
Menu Widgets

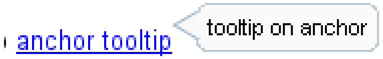
http://dojocampus.org/explorer/#Dijit_Menu

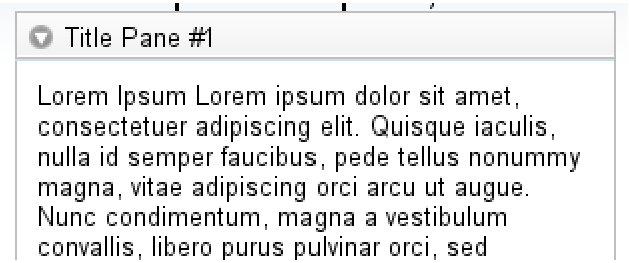
Dijit User Assistance & Feedback Widgets

Dijit User Assistance & Feedback Widgets

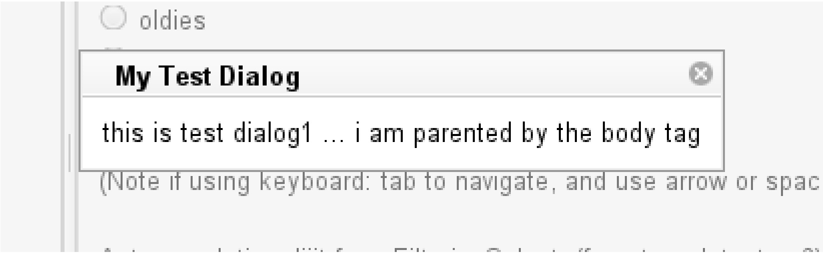
- Dialog, TooltipDialog, ProgressBar, TitlePane, Tooltip





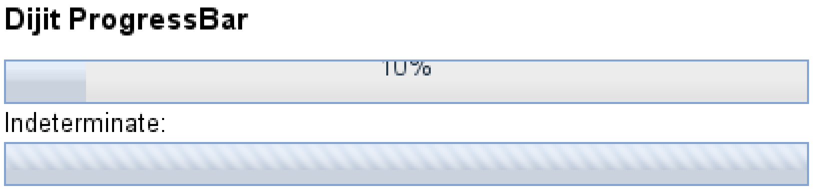


Lorem Ipsum Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Quisque iaculis, nulla id semper faucibus, pede tellus nonummy magna, vitae adipiscing orci arcu ut augue. Nunc condimentum, magna a vestibulum convallis, libero purus pulvinar orci, sed



this is test dialog1 ... i am parented by the body tag

(Note if using keyboard: tab to navigate, and use arrow or spac



dijit.Dialog Example

```
<body>
    ...
     <button id="buttonOne" dojoType="dijit.form.Button" type="button">
        Click me to display a dialog!
        <script type="dojo/method" event="onClick" args="evt">
            // Show the Dialog:
            dijit.byId("dialogOne").show();
        </script>
    </button>

    <div id="dialogOne" dojoType="dijit.Dialog" title="My Dialog Title">
        <div dojoType="dijit.layout.TabContainer" style="width: 200px; height: 300px;">
            <div dojoType="dijit.layout.ContentPane" title="foo">
                Content of Tab "foo"
            </div>
            <div dojoType="dijit.layout.ContentPane" title="bar">
                Hi, I'm Tab "bar"
            </div>
        </div>
    </div>
</body>
```

Demo:

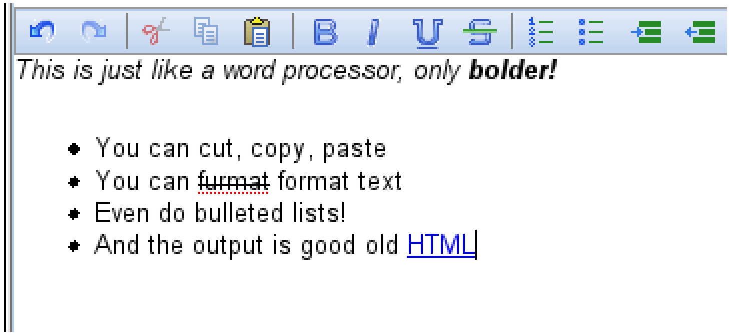
Dialog Widgets

http://dojocampus.org/explorer/#Dijit_Dialog

Dijit Advanced Editing & Display

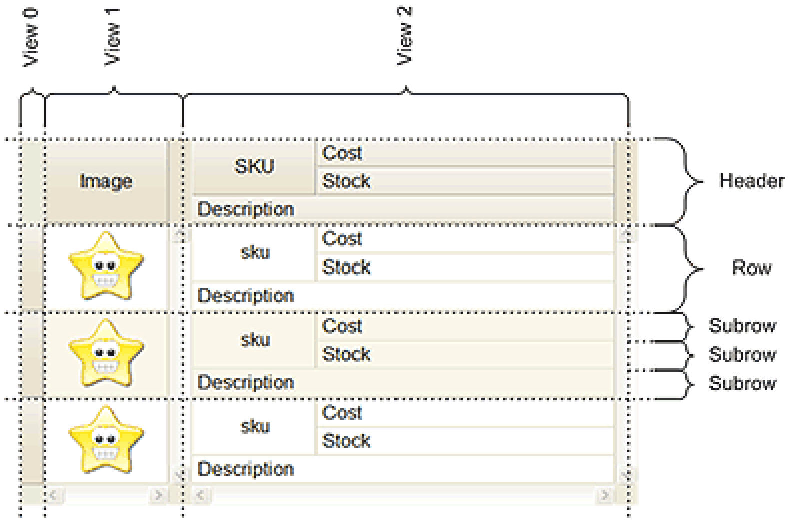
Dijit Advanced Editing & Display Widgets

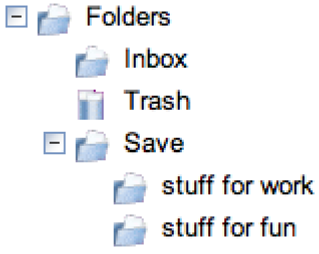
- Editor, Grid, Tree, ColorPalette, InlineEditBox



This is just like a word processor, only ***bolder!***

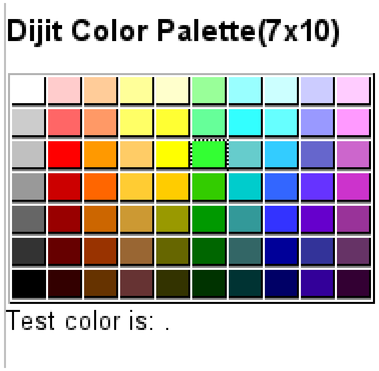
- You can cut, copy, paste
- You can furmat format text
- Even do bulleted lists!
- And the output is good old HTML





- Folders
  - Inbox
  - Trash
  - Save
    - stuff for work
    - stuff for fun

Edit - I trigger the onCha



Dijit Color Palette(7x10)

Test color is: .

dijit.Tree Example (Code)

```
<script language="JavaScript" type="text/javascript">
  dojo.require("dojo.data.ItemFileReadStore");
  dojo.require("dijit.Tree");
  dojo.require("dojo.parser");  // scan page for widgets and instantiate them
</script>

<div dojoType="dojo.data.ItemFileReadStore" jsId="continentStore"
  url="featureexplorer/Dojo/countries.json"></div>

<h3>Tree with hardcoded root node (not corresponding to any item in the store)</h3>
<div dojoType="dijit.Tree" id="tree1" store="continentStore"
   query="{type:'continent'}" labelAttr="name" label="Continents">
  <script type="dojo/method" event="onClick" args="item">
    if(item){
      alert("Execute of node " + continentStore.getLabel(item)
        +", population=" + continentStore.getValue(item, "population"));
    }else{
      alert("Execute on root node");
    }
  </script>
</div>
Click <a href="featureexplorer/Dojo/countries.json" target="_new">here</a> to see the data used to
populate the tree.
```

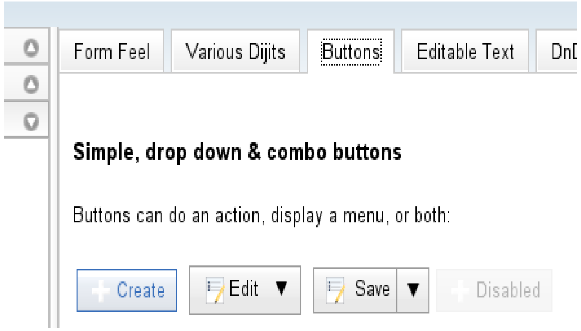
dijit.Tree Example (JSON)

```
{ identifier: 'name',
  label: 'name',
  items: [
     { name:'Africa', type:'continent',
         children:[{_reference:'Egypt'}, {_reference:'Kenya'}, {_reference:'Sudan'}] },
     { name:'Egypt', type:'country' },
     { name:'Kenya', type:'country',
         children:[{_reference:'Nairobi'}, {_reference:'Mombasa'}] },
     { name:'Nairobi', type:'city' },
     { name:'Mombasa', type:'city' },
     { name:'Sudan', type:'country',
         children:{_reference:'Khartoum'} },
     { name:'Khartoum', type:'city' },
     { name:'Asia', type:'continent',
         children:[{_reference:'China'}, {_reference:'India'}, {_reference:'Russia'},
                          {_reference:'Mongolia'}] },
     { name:'China', type:'country' },
     { name:'India', type:'country' },
     { name:'Russia', type:'country' },
     { name:'Mongolia', type:'country' },
     { name:'Australia', type:'continent', population:'21 million',
         children:{_reference:'Commonwealth of Australia'}},
     { name:'Commonwealth of Australia', type:'country', population:'21 million'},
     …
]}
```

Themes and Design

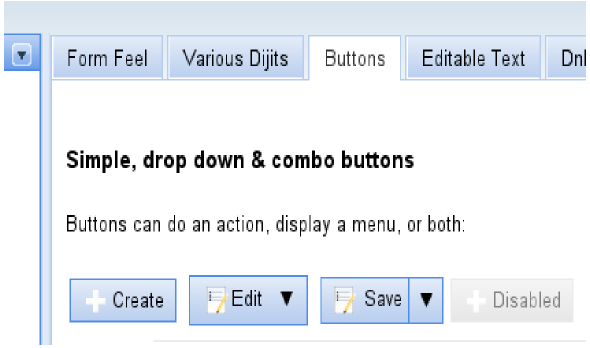
Themes

- Dijit Themes lend a consistent look and feel to widgets
- Built-in themes in Dijit
  - > Tundra (Most commonly used)
  - > Soria
  - > Noir



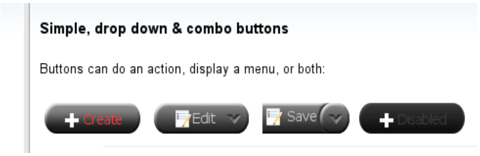
Simple, drop down & combo buttons

Buttons can do an action, display a menu, or both:



Simple, drop down & combo buttons

Buttons can do an action, display a menu, or both:



Simple, drop down & combo buttons

Buttons can do an action, display a menu, or both:

How to Specify a Theme

```
 <head>
   ....
   <style type="text/css">
      @import "/dojo-release-1.5.0/dijit/themes/tundra/tundra.css";
      @import "/dojo-release-1.5.0/dojo/resources/dojo.css";
   </style>
   ...
</head>
<body class="tundra">
   <button dojoType="dijit.form.Button" onclick="create_button">
      Click me to create a new button programatically!!
   </button>
   <div id="button-placeholder"> </div>
</body>
```

Thank you!

Check JavaPassion.com Codecamps!

http://www.javapassion.com/codecamps

"Learn with Passion!"