Stuff we can talk about

- What is Accessibility?
- Accessibility, Design, and Testing
- Evaluation tools
- HTML markup accessibility
- Dynamic Web accessibility (ARIA, keyboard, etc)
- Using the Fluid Skinning System
What is Accessibility?
A New Definition

- Accessibility is the ability of the system to accommodate the needs of the user
- Disability is the mismatch between the user and the interface provided
- We all experience disability
- Accessible software = better software
UI Options

• One size doesn’t fit all

• Allows users to customize your app:
  • layout
  • styling
  • navigation

• Uses FSS by default; can be configured to work with your own classes
Accessibility, Design, and Testing
Better Usability
Designing for everyone

• Look at what an interaction is like for various users and contexts and then (re)envision how it could be
• start with interactions, not technology
• iterative testing
• have a wide-open community
Testing and Checking

• Fluid UX Walkthroughs
• Evaluators: AChecker, WAVE, and more
• General principles:
  • Flexibility
  • Labelling
  • Alternatives

Thursday, September 2, 2010
Step in the shoes of your users...

- Fluid UX Walkthroughs
- Easy ways to assess usability and accessibility
- Combination heuristic evaluation and cognitive walkthrough
- Translated: a checklist with scenarios
- Anyone can do one

http://wiki.fluidproject.org/display/fluid/User+Experience+Walkthroughs
Simple Accessibility Evaluation

1. Try changing your font size, window size and resolution
2. Look critically at your page’s layout, structure & content
3. Use the Tab key to navigate through all controls
4. Check for alternatives to images, sound, and video

Thursday, September 2, 2010
Layout and Structure

• Is the page structured into logical sections?
• Are the sections clearly labeled?
• Are there sufficient non-visual cues for site structure?
• Are there sufficient visual cues?
• Is the most important information prominent?
• Is navigation consistent from page to page?
User Testing and Interaction Design

Remote User Testing

Other techniques for making UIs awesome
Tools for Evaluating Accessibility
Evaluation Tools

1. Static Analyzers
2. HTML and CSS
3. Accessibility
HTML & CSS Validation

- Validators ensure that HTML and CSS are to spec
- Passing doesn’t mean you’re accessible, but it’s a start
- HTML
  - [http://validator.w3.org/](http://validator.w3.org/)
  - [http://jigsaw.w3.org/css-validator/](http://jigsaw.w3.org/css-validator/)

---

### Errors found while checking this document as HTML5!

<table>
<thead>
<tr>
<th>Result</th>
<th>40 Errors, 2 warning(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td><a href="http://www.google.com/">http://www.google.com/</a></td>
</tr>
<tr>
<td>Encoding</td>
<td>iso-8859-1</td>
</tr>
<tr>
<td>Doctype</td>
<td>HTML5</td>
</tr>
<tr>
<td>Root Element</td>
<td>html</td>
</tr>
</tbody>
</table>
Accessibility Checkers

aChecker

http://achecker.ca/checker/index.php

Wave

http://jigsaw.w3.org/css-validator/
Accessibility Checkers

aChecker

http://achecker.ca/checker/index.php
Accessibility Checkers

Wave

http://jigsaw.w3.org/css-validator/

WAVE has detected no accessibility errors...
but you must still check your page to ensure it is actually accessible.
The following are present in the head section or apply to this page in general:

Fluid
Designing software that works - for everyone.

Fluid Daily Build Resources
The demonstration sites available below are updated from SVN nightly. Changes made during the day will not necessarily be available until the following day.

Interactive demos are now available on our demo portal

Fluid Integration Examples

Infrastructure
Accessibility Checkers: What They Do

- Statically analyze markup
- Specify the Guideline for Validation
  - e.g. WCAG 2.0 AA
- Will alert you to inaccessible markup
  - e.g. missing alt text on images

Line 103, Column 9: Image used for input element is missing Alt text.
<input type="image" src="/gallery/graphics/gallery_circle_arrow.gif" class="search-arrow" onmouseout=""/>
Accessibility Checkers: Limitations

- No Magic
  - Is the alt text meaningful?

- Static analysis
  - Will JavaScript make it inaccessible?

- Markup based validation
  - How will CSS affect the page?
Colour Validation

- View in respect to Colour Blindness
- Determine Adjustability of Colours

http://colorfilter.wickline.org
http://vischeck.com
Colour Validation

Limitations

• Automating testing of interfaces is hard
  • e.g. determining contrast levels

Is this contrast level to spec?
Design Early

Easier and cheaper to make good design choices early

Colour Pickers that also measure contrast

http://gmazzocato.altervista.org/colorwheel/wheel.php
http://www.snook.ca/technical/colour_contrast/colour.html
Accessible Markup
Concepts of HTML Accessibility

- Label as much as you can
- Use semantic markup to describe your page
- Design in layers
- Textual alternatives
- Clearly delineate navigation from content
Avoiding Repetition

Our Mission

The Fluid community is an international group of designers, developers, volunteers, and advisers who focus on a common mission: improving the user experience of community and open source web applications. To accomplish this goal, Fluid addresses the issue of user experience on all levels.

- User Interface -- combining both design and technology to create a living library of sharable user interface components
- Framework -- distributing a framework that provides an easy way to build JavaScript-based user interfaces that are highly flexible and reusable. Built using Web standards and the jQuery toolkit, Infusion provides a lightweight application development framework supporting simple Model View Controller (MVC) techniques
- Design -- providing a Design Handbook (for designers and developers alike) including tools and techniques that are easy to use, learn, and modify
- Education -- giving demonstrations and teaching others at
Avoiding Repetition

Our Mission

The Fluid community is an international group of designers, developers, volunteers, and advisers who focus on a common mission: improving the user experience of community and open source web applications. To accomplish this goal, Fluid addresses the issue of user experience on all levels.

- User Interface -- combining both design and technology to create a living library of sharable user interface components
- Framework -- distributing a framework that provides an easy way to build JavaScript-based user interfaces that are highly flexible and reusable. Built using Web standards and the jQuery toolkit, Infusion provides a lightweight application development framework supporting simple Model View Controller (MVC) techniques
- Design -- providing a Design Handbook (for designers and developers alike) including tools and techniques that are easy to use, learn, and modify
- Education -- giving demonstrations and teaching others at
Avoiding Repetition

Our Mission

The Fluid community is an international group of designers, developers, volunteers, and advisers who focus on a common mission: improving the user experience of community and open source web applications. To accomplish this goal, Fluid addresses the issue of user experience on all levels.

- User Interface -- combining both design and technology to create a living library of sharable user interface components.
  - Fluid provides a framework that supports modern interfaces that are highly flexible and reusable. Built using Web standards and the jQuery toolkit, Infusion provides a lightweight application development framework supporting simple Model View Controller (MVC) techniques.
- Design -- providing a Design Handbook (for designers and developers alike) including tools and techniques that are easy to use, learn and modify.
- Education -- giving demonstrations and teaching others at...
Avoid Repetition

Skip Links

```html
<div id="jumplinks">
    <a href="#content" title="Jump to content"></a>
    <a href="#nav" title="Jump to navigation menu"></a>
</div>

<a id="nav" title="navigation menu"></a>
<!-- Navigation bar goes here -->
<a title="content area" name="content"></a>
<!-- Main page content goes here -->
```
Designing Navigation

- Keep in mind that keyboard navigation is:
  - not just for screen reader users
  - is linear and 1-dimensional
  - can be slow and tedious

- Skip links should be available and **visible to all**

- Place them as high in the page as possible
Navigable Headings

Fluid Daily Build Resources

The demonstration sites available below are updated from SVN nightly. Changes made during the day will not necessarily be available until the following day.

Interactive demos are now available on our demo portal

Fluid Integration Examples

The following examples illustrate more than one Fluid component, integrated into real-world settings.

uPortal Instance

- Demo uPortal site

This is a live instance of uPortal, built from their trunk nightly. It incorporates the following Infusion features:

- Fluid Skinning System The uPortal interface is styled using the Fluid Skinning System (FSS).
- Layout Reorderer The portlets are reorderable using the Infusion Layout Reorderer.
- Pager If you log in as "admin" and choose the "Portlet Manager" in the right sidebar, the list of registered portlets is paged using the Infusion Pager.
- As well, if you choose "Add Content" in the right sidebar, you can add the "Javascript Demos" portlet (in the Applications section) which

Infrastructure

- Continuum Build Server
- Unit Tests
- Manual Tests

Infusion Builder

- "Infusion Builder Demo"
- Unit Tests: JavaScript Unit Tests
- Unit Tests: PHP - Builder Utilities Test
- Unit Tests: PHP - Group Class Test
- Unit Tests: PHP - Module Class Test
- Unit Tests: PHP - Post Class Test

Engage Demos

These are best viewed in Safari 4

- Fluid Engage Mobile
Navigable Headings

Fluid Daily Build Resources

The demonstration sites available below are updated from SVN nightly. Changes made during the day will not necessarily be available until the following day.

Interactive demos are now available on our demo portal

Fluid Integration Examples

The following examples illustrate more than one Fluid component, integrated into real-world settings.

uPortal Instance

- Demo uPortal site

This is a live instance of uPortal, built from their trunk nightly. It incorporates the following Infusion features:

- Fluid Skinning System The uPortal interface is styled using the Fluid Skinning System (FSS).
- Layout Reorderer The portlets are reorderable using the Infusion Layout Reorderer.
- Pager If you log in as "admin" and choose the "Portlet Manager" in the right sidebar, the list of registered portlets is paged using the Infusion Pager.
- As well, if you choose "Add Content" in the right sidebar, you can add the "Javascript Demos" portlet (in the Applications section) which

Infrastructure

- Continuum Build Server
- Unit Tests
- Manual Tests

Infusion Builder

- "Infusion Builder Demo"
- Unit Tests: JavaScript Unit Tests
- Unit Tests: PHP - Builder Utilities Test
- Unit Tests: PHP - Group Class Test
- Unit Tests: PHP - Module Class Test
- Unit Tests: PHP - Post Class Test

Engage Demos

These are best viewed in Safari 4

- Fluid Engage Mobile
Navigable Headings

**Level One**

**Level Two**

The demonstration sites available below are updated from SVN nightly. Changes made during the day will not necessarily be available until the following day.

**Interactive demos are now available on our [demo portal](#)**

**Fluid Integration Examples**

The following examples illustrate more than one Fluid component, integrated into real-world settings.

**uPortal Instance**

- Demo uPortal site

This is a live instance of uPortal, built from their trunk nightly. It incorporates the following Infusion features:

- **Fluid Skinning System** The uPortal interface is styled using the Fluid Skinning System (FSS).
- **Layout Reorderer** The portlets are reorderable using the Infusion Layout Reorderer.
- **Pager** If you log in as "admin" and choose the "Portlet Manager" in the right sidebar, the list of registered portlets is paged using the Infusion Pager.
- As well, if you choose "Add Content" in the right sidebar, you can add the "Javascript Demos" portlet (in the Applications section) which...

**Infrastructure**

- Continuum Build Server
- Unit Tests
- Manual Tests

**Infusion Builder**

- "Infusion Builder Demo"
- Unit Tests: JavaScript Unit Tests
- Unit Tests: PHP - Builder Utilities Test
- Unit Tests: PHP - Group Class Test
- Unit Tests: PHP - Module Class Test
- Unit Tests: PHP - Post Class Test

**Engage Demos**

These are best viewed in [Safari 4](#)

- Fluid Engage Mobile
Navigable Headings

Level One

Level Two

The demonstration sites available below are updated from SVN nightly. Changes made during the day will not necessarily be available until the following day.

Interactive demos are now available on our demo portal

Level Three

The following examples illustrate more than one Fluid component, integrated into real-world settings.

uPortal Instance

- Demo uPortal site

This is a live instance of uPortal, built from their trunk nightly. It incorporates the following Infusion features:

- Fluid Skinning System The uPortal interface is styled using the Fluid Skinning System (FSS).
- Layout Reorderer The portlets are reorderable using the Infusion Layout Reorderer.
- Pager If you log in as "admin" and choose the "Portlet Manager" in the right sidebar, the list of registered portlets is paged using the Infusion Pager.
- As well, if you choose "Add Content" in the right sidebar, you can add the "Javascript Demos" portlet (in the Applications section) which

Level Three

- Continuum Build Server
- Unit Tests
- Manual Tests

Level Three

- Infusion Builder Demo
- Unit Tests: JavaScript Unit Tests
- Unit Tests: PHP - Builder Utilities Test
- Unit Tests: PHP - Group Class Test
- Unit Tests: PHP - Module Class Test
- Unit Tests: PHP - Post Class Test

These are best viewed in Safari 4

Level Three

- Fluid Engage Mobile
Navigable Headings

Level One

Level Two

The demonstration sites available below are updated from SVN nightly. Changes made during the day will not necessarily be available until the following day.

Interactive demos are now available on our demo portal

Level Three

The following examples illustrate more than one Fluid component, integrated into real-world settings.

Level Four

- Demo uPortal site

This is a live instance of uPortal, built from their trunk nightly. It incorporates the following Infusion features:

- Fluid Skinning System The uPortal interface is styled using the Fluid Skinning System (FSS).
- Layout Reorderer The portlets are reorderable using the Infusion Layout Reorderer.
- Pager If you log in as "admin" and choose the "Portlet Manager" in the right sidebar, the list of registered portlets is paged using the Infusion Pager.
- As well, if you choose "Add Content" in the right sidebar, you can add the "Javascript Demos" portlet (in the Applications section) which

Level Three

- Continuum Build Server
- Unit Tests
- Manual Tests

- "Infusion Builder Demo"
- Unit Tests: JavaScript Unit Tests
- Unit Tests: PHP - Builder Utilities Test
- Unit Tests: PHP - Group Class Test
- Unit Tests: PHP - Module Class Test
- Unit Tests: PHP - Post Class Test

These are best viewed in Safari 4

- Fluid Express Mobile
Navigable Headings

Fluid
Designing what works - for everyone.

Interactive demos are now available on our demo portal

Fluid Skinning System
The uPortal interface is styled using the Fluid Skinning System (FSS).

Layout Reorderer
The portlets are reorderable using the Infusion Layout Reorderer.

Pager
If you log in as "admin" and choose the "Portlet Manager" in the right sidebar, the list of registered portlets is paged using the Infusion Pager.

As well, if you choose "Add Content" in the right sidebar, you can add the "Javascript Demos" portlet (in the Applications section) which...

These are best viewed in Safari 4

Fluid Engine Mobile

- Continuum Build Server
- Unit Tests
- Manual Tests
- "Infusion Builder Demo"
- Unit Tests: JavaScript Unit Tests
- Unit Tests: PHP - Builder Utilities Test
- Unit Tests: PHP - Group Class Test
- Unit Tests: PHP - Module Class Test
- Unit Tests: PHP - Post Class Test
Navigating Headings

Fluid Daily Build Resources
The demonstration sites available below are updated from SVN nightly. Changes made during

Interactive demos are now available

Fluid Integration Examples
The following examples illustrate more than one Fluid component, integrated into real-world settings.

uPortal Instance
- Demo uPortal site

Infusion
- "Infusion"
- Unit 1
- Unit 2
- Unit 3
- Unit 4

Engage Demos
These are:
- Fluid

Heading List
Fluid Daily Build Resources: 2
Fluid Integration Examples: 3
uPortal Instance: 4
Sakai Mock-up: 4
uPortal 3 Mock-up: 4
Infrastructure: 3
Infusion Builder: 3
Engage Demos: 3

Sort Headings
- In Tab Order
- Alphabetically

Display
- All Headings
- Level1 Headings
- Level2 Headings
- Level3 Headings
<body>

<h1>Fluid: Designing Software that works - For everyone.</h1>

<h2>Fluid Daily Build Resources</h2>

<div>
  <div class="fl-col">
    <h3>Fluid Integration Examples</h3>
    <h4>uPortal Instance</h4>
    <!-- Content goes here -->
    <h4>Sakai Mock-up</h4>
  </div>
  <div class="fl-col">
    <h3>Infrastructure</h3>
    ...
  </div>
</div>

Thursday, September 2, 2010
Labelling Forms

User Interface Options

- **Easier to see**
  - Font style: Serif
  - Minimum text size: [ ] 8 pt
  - Text Spacing: Regular
  - Line Spacing: [ ]
  - Contrast: Medium Contrast
  - Background Images: [ ] Yes
  - Simplified Layout: [ ] Yes

- **Easier to find**

Preview window (updates automatically)

Tools

- Home
- Profile
- Membership
- Schedule
- Resources
- Worksite Setup
- Preferences
- Account

Reset  Save and apply  Cancel
Labelling Forms
Labelling Forms

Thursday, September 2, 2010
Labelling Forms

<label for="text-font">Font style:</label>
<select id="text-font" name="text-font-selection">
  <option value="serif">Serif</option>
  <option value="sansSerif">Sans-Serif</option>
  <option value="arial">Arial</option>
  <option value="verdana">Verdana</option>
  <option value="courier">Courier</option>
  <option value="times">Times</option>
</select>

Thursday, September 2, 2010
Grouping Forms
Grouping Forms
<fieldset>
    <legend>Background Images</legend>
    <span>
        <input type="radio" value="true" checked="checked"
            name="background-images-selection" id="background-yes">
        <label for="background-yes">Yes</label>
    </span>
    <span>
        <input type="radio" value="false" name="background-images-selection"
            id="background-no">
        <label for="background-no">No</label>
    </span>
</fieldset>
Tables

- Tables got a really bad rap in the ‘90s
- Deservedly so
- Use them for data, not layouts
- They can be big: summarize them
- Seriously, you don’t need them for layouts
# Grouping Forms

## Header

<table>
<thead>
<tr>
<th>Name</th>
<th>User Id</th>
<th>Points</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ahn, Jason</td>
<td>15234314</td>
<td>87</td>
<td>Lorem ipsum dolor sit amet.</td>
</tr>
<tr>
<td>Akerman, Krista</td>
<td>19206726</td>
<td>79</td>
<td>Consectetuer adipiscing elit.</td>
</tr>
<tr>
<td>Almeida, Elaine</td>
<td>19286942</td>
<td>92</td>
<td>Suspendisse eu.</td>
</tr>
<tr>
<td>Azer, Tamer</td>
<td>19446940</td>
<td>68</td>
<td></td>
</tr>
<tr>
<td>Bandali, Salim</td>
<td>19283721</td>
<td>69</td>
<td>Ut egestas, urna at rhoncus gravida, nisl nisi condimentum pede.</td>
</tr>
</tbody>
</table>

## Body
### Table Structure

<table summary="An editable table of student grades and instructor comments.">
  <thead>
    <tr>
      <th>Name</th>
      <th>User ID</th>
      <th>Points</th>
      <th>Comments</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Ahn, Jason</td>
      <td>15234314</td>
      <td><input type="text" name="points" value="87"></td>
      <td>Lorem ipsum dolor sit amet.</td>
    </tr>
  </tbody>
</table>
what is “alt” text?

- It is read by screen readers in place of images allowing the content and function of the image to be accessible to those with visual or certain cognitive disabilities.

- It is displayed in place of the image in user agents (browsers) that don't support the display of images or when the user has chosen not to view images.

- It provides a semantic meaning and description to images which can be read by search engines or be used to later determine the content of the image from page context alone.

http://www.webaim.org/techniques/alttext/
The “alt” attribute

• Be accurate and equivalent in presenting the same content and function as presented by the image.

• Be succinct. Typically no more than a few words are necessary.

• NOT be redundant or provide the exact same information as text within the context.

• NOT use the phrases "image of ..." or "graphic of ..." to describe the image. It usually apparent to the user that it is an image.

http://www.webaim.org/techniques/alttext/
“alt” text in code

<img src="boat.gif" alt="Big Boat" />
Alternatives to Images

Alt Text

```html
<img src="http://eich.wsg.mcgill.ca/largeimages/M2000.38.97-P1.jpg" alt="Curling stone">
```

Background Images + Real Text

```html
<a href="http://fluidproject.org/products/infusion/download-infusion" title="Download Fluid Infusion">
  Download Infusion
</a>
```
Web 2.0 Accessibility
DHTML: A New Can of Worms

- Shift from documents to applications
- Familiar a11y techniques aren’t enough
- Most DHTML is completely inaccessible
- New techniques are still being figured out
The Problem

• Custom widgets often look, but don’t act, like their counterparts on the desktop
• HTML provides only simple semantics
• Not enough information for ATs
• Dynamic updates require new design strategies to be accessible
The Solution

- Describe user interfaces with ARIA
- Add consistent keyboard controls
- Provide flexible styling and presentation
How assistive technologies work
Assistive Technologies

• Present and control the user interface in different ways

• Not just screen readers!

• Use built-in operating system APIs to understand the user interface
OS AT APIs

- A channel for UI introspection
- What’s on screen?
- How are things labelled, organized, etc.?
- What states are things in?
- UI Roles, states, properties
The Role-Based Model

- Most platform APIs work the same way
- Give each UI widget a name
  - e.g. slider, tabs, dialog, button, text field
- Names imply behaviour
- For AT users, names define interactions
View Hierarchies
Supporting Assistive Technology
Opaque Markup

// These are tabs. How would you know?
<ol>
  <li id="ch1Tab">
    <a href="#ch1Panel">CHAPTER 1</a>
  </li>
  <li id="ch2Tab">
    <a href="#ch2Panel">CHAPTER 2</a>
  </li>
  <li id="quizTab">
    <a href="#quizPanel">QUIZ</a>
  </li>
</ol>
<div>
  <div id="ch1Panel">Chapter 1 Stuff</div>
  <div id="ch2Panel">Chapter 2 Stuff</div>
  <div id="quizTab">Quiz Stuff</div>
</div>
Chapter 1
Prolegomena

What is logic?

Logic is the study of the consistency of beliefs. For beliefs to be consistent it must be possible for them to obtain at the same time. For example, it is illogical to believe that the sky is completely blue and that the sky is completely red because the sky being entirely blue is inconsistent with its being entirely red, i.e. it is not possible for the sky to be entirely red at the same time as its being entirely blue.

Logic is also a study of "logical consequence", i.e. what follows by necessity from something else. By studying inconsistency of beliefs, philosophers are able to study the validity of arguments, as will be shown later. Methods of finding whether certain arguments are valid is described later.

The symbolisation of these sentences, known as formalisation, simplifies and quickens this process. It also enables the philosopher to clarify ideas using an unambiguous language in which to represent thoughts. The sophistication of the language used enables greater insights into the significance of these thoughts (and a cursory analysis of more logical languages is described in Other Logics).
ARIA

- Accessible Rich Internet Applications
- W3C specification in the works
- Fills the semantic gaps in HTML
- Roles, states, and properties
- Live regions
Roles, States, Properties

- **Roles** describe widgets not present in HTML 4
  - slider, menubar, tab, dialog
- **Properties** describe characteristics:
  - draggable, hasPopup, required
- **States** describe what’s happening:
  - busy, disabled, selected, hidden
Using ARIA

// Now *these* are Tabs!
<ol role="tablist" tabindex="0">
  <li id="ch1Tab" role="tab"><a href="#ch1Panel" tabindex="-1">CHAPTER 1</a></li>
  <li id="ch2Tab" role="tab"><a href="#ch2Panel" tabindex="-1">CHAPTER 2</a></li>
  <li id="quizTab" role="tab"><a href="#quizPanel">QUIZ</a></li>
</ol>

<div>
  <div id="ch1Panel" role="tabpanel" aria-labelledby="ch1Tab">Chapter 1 Stuff</div>
  <div id="ch2Panel" role="tabpanel" aria-labelledby="ch2Tab">Chapter 2 Stuff</div>
  <div id="quizTab" role="tabpanel" aria-labelledby="quizTab">Quiz Stuff</div>
</div>
Breaking it Down

<ol>
  <li>CHAPTER 1</li>
  <li>CHAPTER 2</li>
  <li>QUIZ</li>
</ol>

<div>
  <div>Chapter 1 Stuff</div>
  <div>Chapter 2 Stuff</div>
  <div>Quiz Stuff</div>
</div>
Progressive Enhancement

Graceful Degradation
Make it work without JavaScript

<ol>
  <li><a href="#ch1Panel">CHAPTER 1</a></li>
  <li><a href="#ch2Panel" tabindex="-1">CHAPTER 2</a></li>
  <li><a href="#quizPanel">QUIZ</a></li>
</ol>

<div>
  <div id="ch1Panel">Chapter 1 Stuff</div>
  <div id="ch2Panel">Chapter 2 Stuff</div>
  <div id="quizTab">Quiz Stuff</div>
</div>
// Identify the container as a list of tabs.
tabContainer.attr("role", "tablist");

// Give each tab the "tab" role.
tabs.attr("role", "tab");

// Give each panel the appropriate role,
panels.attr("role", "tabpanel");
panels.each(function (idx, panel) {
    var tabForPanel = that.tabs.eq(idx);
    // Relate the panel to the tab that labels it.
    $(panel).attr("aria-labelledby", tabForPanel[0].id);
});
The Problem with Roles

Roles are driven by 1980’s era desktop widgets
The Problem with Roles

The Web is driving hybrid UIs
The Problem with Roles

... even on the desktop
Analyzing User Interfaces
Inline Edit Roles

The quick brown fox jumped over the lazy dogs and then...

The quick brown fox jumped the lazy dogs and then...

The quick brown fox jumped over the lazy dogs and then...

The over quick brown fox jumped the lazy dogs and then...

The quick brown fox jumped over the lazy dogs and then...

The quick brown fox jumped the lazy dogs and then...

The quick brown fox jumped over the lazy dogs and then...

The over quick brown fox jumped the lazy dogs and then...
Inline Edit Roles

The quick brown fox jumped over the lazy dogs and then...
The quick brown fox over jumped the lazy dogs and then...
The quick brown fox jumped over the lazy dogs and then...
The quick brown fox jumped over the lazy dogs and then...
The quick brown fox jumped over the lazy dogs and then...
The over quick brown fox jumped the lazy dogs and then...
Inline Edit Roles

The quick brown fox jumped over the lazy dogs and then...
The quick brown fox jumped over the lazy dogs and then...
The quick brown fox jumped over the lazy dogs and then...
The quick brown fox jumped over the lazy dogs and then...
The quick brown fox jumped over the lazy dogs and then...
The quick brown fox jumped over the lazy dogs and then...
The over quick brown fox jumped the lazy dogs and then...
The quick brown fox over jumped the lazy dogs and then...
The quick brown fox over jumped the lazy dogs and then...
The quick brown fox over jumped the lazy dogs and then...
The quick brown fox over jumped the lazy dogs and then...
The quick brown fox over jumped the lazy dogs and then...
The quick brown fox over jumped the lazy dogs and then...

Button?

Text Field?
Inline Edit Behaviours

- Read-only
- Activatable
- Editable
- Undoable
Keyboard Accessibility
Keyboard Navigation

- Everything that works with the mouse should work with the keyboard
- ... but not always in the same way
- Support familiar conventions

http://dev.aol.com/dhtml_style_guide
Keyboard Conventions

• **Tab** key focuses the control or widget

• **Arrow keys** select an item

• **Enter** or **Spacebar** activate an item

• Tab is handled by the browser. For the rest, you need to write code. A lot of code.
Keyboard a11y: Tabs

Chapter 1
Prolegomena
Original chapter from Wikibooks

What is logic?

Logic is the study of the consistency of beliefs. For beliefs to be consistent it must be possible for them to obtain at the same time. For example, it is illogical to believe that the sky is completely blue and that the sky is completely red because the sky being entirely blue is inconsistent with its being entirely red, i.e. it is not possible for the sky to be entirely red at the same time as its being entirely blue.

Logic is also a study of "logical consequence", i.e. what follows by necessity from something else. By studying inconsistency of beliefs, philosophers are able to study the validity of arguments, as will be shown later. Methods of finding whether certain arguments are valid is described later.

The symbolisation of these sentences, known as formalisation, simplifies and quickens this process. It also enables the philosopher to clarify ideas using an unambiguous language in which to represent thoughts. The sophistication of the language used enables greater insights into the significance of these thoughts (and a cursory analysis of more logical languages is described in Other Logics).
Tabindex examples

<!-- Tab container should be focusable -->
<ol id="animalTabs" tabindex="0">
  <!-- Individual Tabs shouldn’t be focusable -->
  <!-- We’ll focus them with JavaScript instead -->
  <li id="tab1">
    <a href="#cats" tabindex="-1">Cats</a>
  </li>
  <li id="tab2">
    <a href="#cats" tabindex="-1">Dogs</a>
  </li>
  <li id="tab3">
    <a href="#cats" tabindex="-1">Alligators</a>
  </li>
</ol>
Making Things Tabbable

- Tabindex varies subtly across browsers
- jQuery.attr() normalizes it as of 1.3
- For all the gory details:

// Make the tablist accessible with the Tab key.
.tabContainer.attr("tabindex", "0");
// And take the anchors out of the Tab order.
$("a", tabs).attr("tabindex", "-1");
Adding the Arrow Keys

// Make each tab accessible with the left and right arrow keys.
tabContainer.fluid("selectable", {
    selectableSelector: "li",
    direction: fluid.a11y.orientation.HORIZONTAL,
    onSelect: function (tab) {
        $(tab).addClass(that.options.styles.highlighted);
    },
    onUnselect: function (tab) {
        $(tab).removeClass(that.options.styles.highlighted);
    }
});
Making Them Activatable

// Make each tab activatable with Spacebar and Enter.
tabs.fluid("activatable", function (evt) {
    // Your handler code here. Maybe the same as .click()?
});
Documentation

• Tutorial:
  http://wiki.fluidproject.org/display/fluid/Keyboard+Accessibility+Tutorial

• API Reference:
  http://wiki.fluidproject.org/display/fluid/Keyboard+Accessibility+Plugin+API
Accessibility Resources

http://codetalks.org

http://wiki.fluidproject.org/display/fluid/DHTML+Developer+Checklist

http://wiki.fluidproject.org/display/fluid/UX+Accessibility+Walkthrough+Protocols

http://developer.mozilla.org/en/docs/Accessible_DHTML

http://developer.mozilla.org/en/docs/Key-navigable_custom_DHTML_widgets

CSS Frameworks

• “If you’re going to use a framework, it should be yours; one that you’ve created. You can look at existing frameworks for ideas and hack at it. But the professionals in this room are not well served by picking up a framework and using it as-is.”

• - Eric Meyer
Fluid Skinning System

- FSS is built to be hacked on
- Provides a core set of building blocks
- Reset, text, layouts, themes
- Namespaced: no conflicts with your stuff
- Themes for better legibility & readability

http://wiki.fluidproject.org/x/96M7
Dynamic Skinning

Demo UI Options?
UI Options

My Dashboard

User Interface Options

- Easier to see
  - Font style: Serif
  - Minimum text size: pt
  - Text Spacing: Regular
  - Line Spacing:
  - Contrast: Medium Contrast
  - Background Images: Yes
  - Simplified Layout: Yes

- Easier to find

Reset | Save and apply | Cancel

Preview window (updates automatically)

Tools
- Home
- Profile
- Membership
- Schedule
- Resources
- Worksite Setup
- Preferences
- Account

Add Widgets | Edit Appearance

© 2004-2008 The Sakai Foundation
Portions of Sakai are copyrighted by other parties as described in the Acknowledgments screen

fluid engage

Thursday, September 2, 2010
UI Options
Code!
FSS Themes

Slate

Mist

High Contrast
mFSS: iPhone Theme

<head>
    <link type="text/css" rel="stylesheet"
        href="fss-mobile-layout.css" />
    <link type="text/css" rel="stylesheet"
        href="fss-mobile-theme-iphone.css" />
</head>

<body class="fl-theme-iphone">
mFSS: Android Theme

<head>
  <link type="text/css" rel="stylesheet"
  href="fss-mobile-layout.css" />
  <link type="text/css" rel="stylesheet"
  href="fss-mobile-theme-android.css" />
</head>

<body class="fl-theme-android">

</body>
mFSS: Desktop

<head>
  <link type="text/css" rel="stylesheet"
       href="fss-layout.css" />
  <link type="text/css" rel="stylesheet"
       href="fss-theme-rust.css" /></head>

<body class="fl-theme-rust">

</body>
FSS: Desktop

<head>
  <link type="text/css" rel="stylesheet" href="fss-layout.css" />
  <link type="text/css" rel="stylesheet" href="fss-theme-slate.css" />
</head>

<body class="fl-theme-slate">

</body>
mFSS Themes

iPhone

Android

Small Thumbnails
- Go Back
- Go Back
- Go Back

Small Thumbnails
- Go Back
- Go Back
- Go Back

Lots of info
- Go Back
Donec convallis viverra quam, nec aliquet libero interdum placerat. Nunc arcu turpis, tincidunt at condimentum in, dapibus vel ipsum.

Lots of info
- Go Back
Donec convallis viverra quam, nec aliquet libero interdum placerat. Nunc arcu turpis, tincidunt at condimentum in, dapibus vel ipsum.

- Go Back
Donec convallis viverra quam, nec aliquet libero interdum placerat. Nunc arcu turpis, tincidunt at condimentum in, dapibus vel ipsum.
mFSS: iPhone Theme

<head>
  <link type="text/css" rel="stylesheet"
       href="fss-mobile-layout.css"/>
  <link type="text/css" rel="stylesheet"
       href="fss-mobile-theme-iphone.css"/>
</head>

<body class="fl-theme-iphone">

</body>
mFSS: Android Theme

<head>
    <link type="text/css" rel="stylesheet"
        href="fss-mobile-layout.css" />
    <link type="text/css" rel="stylesheet"
        href="fss-mobile-theme-android.css" />
</head>

<body class="f1-theme-android">

</body>
mFSS: Back Button

<a href="../mobile.html" class="fl-button fl-backButton">
  <span class="fl-button-inner">Back Button</span>
</a>
FSS: Lists

<ul class="fl-list-menu">
  <li><a href="#">Link Text</a></li>
  <li><a href="#">Link Text</a></li>
  <li><a href="#">Link Text</a></li>
</ul>
mFSS: Image Grids

<ul class="fl-list-menu fl-list-thumbnails fl-grid">
  <li><a href="20.png">
    <img class="fl-icon" src="20.png" alt="twisty icon" />
  </a></li>
  <li><a href="2.png">
    <img class="fl-icon" src="2.jpg" alt="round icon" />
  </a></li>
</ul>
FSS: Widgets

<div class="fl-widget">
    <div class="fl-widget-titlebar">
        <a class="fl-force-left fl-icon fl-icon-more" href="#">More</a>
        <a class="fl-force-right fl-icon fl-icon-close" href="#">Close</a>
        <a class="fl-button-right" href="#"><strong class="fl-button-inner">Settings</strong></a>
    </div>
    <h2>Regular Widget</h2>
    <div class="fl-widget-options">
        <ul>
            <li><a href="option1.html">Option 1</a></li>
            <li><a href="option2.html">Option 2</a></li>
        </ul>
    </div>
    <div class="fl-widget-content">
        <p>this is where the content goes</p>
    </div>
</div>
Other Stuff We Can Chat About
The Web
Architecture of the Web

- Separation of structure from presentation
- Declarative
- Stateless
- Interoperable
Structure vs. Presentation

<ul class="fl-list-menu">
  <li>
    <a href="#">Link Text</a>
  </li>
  <li>
    <a href="#">Link Text</a>
  </li>
  <li>
    <a href="#">Link Text</a>
  </li>
</ul>

 Thursday, September 2, 2010
Declarative Programming

Declarative programming is a programming paradigm that expresses the logic of a computation without describing its control flow.
Declarative Programming

what

not

how
CONCERNING BEARDS

The priests of the Greek Church have been bearded for upwards of a thousand years. Bessarion came into Italy with the Archbishop of Russia in an effort to bring about a union between the Greek and Latin Churches. He found it easy to subscribe to the orthodoxy of the Latin Church, which got him his cardinal’s hat. Bessarion was one of the stoutest men of his time, and his long beard and that of his companion so won the admiration of the Court of Rome that everyone longed to look like this distinguished man, were it only in the growth on his chin, and this fine Greek beard soon produced a number of Latin imitations. But Bessarion’s beard was not so well received in France. The great man being sent thither as a legate, visited through policy the Duke of Burgundy before he saw King Louis XI. This was a bad blunder. Louis was furious at the preference given his rival. At the first audience granted Bessarion, Louis seized him roughly by the beard and abused him shamefully. One can imagine how this must have delighted Louis'
The priests of the Greek Church have been bearded for upwards of a thousand years. Bessarion came into Italy with the Archbishop of Russia in an effort to bring about a union between the Greek and Latin Churches. He found it easy to subscribe to the orthodoxy of the Latin Church, which got him his cardinal’s hat. Bessarion was one of the stoutest men of his time, and his long beard and that of his companion so won the admiration of the Court of Rome that everyone longed to look like this distinguished man, were it only in the growth on his chin, and this fine Greek beard soon produced a number of Latin imitations. But Bessarion’s beard was not so well received in France. The great man being sent thither as a legate, visited through policy the Duke of Burgundy before he saw King Louis XI. This was a bad blunder. Louis was furious at the preference given his rival. At the first audience granted Bessarion, Louis seized him roughly by the beard and abused him shamefully. One can imagine how this must have delighted Louis’ subjects. The result, however, was far from pleasant for Bessarion.
if (!highlighted) {
    ctx.globalAlpha = 0.3;
} else {
    // Draw the scroll track rectangle.
    var clientLength = this._getClientLength();
    ctx.fillStyle = theme.scrollTrackFillStyle;
    ctx.fillRect(NIB_PADDING + 0.5, 0.5,
                 clientLength - 2*NIB_PADDING, thickness - 1);
    ctx.strokeStyle = theme.scrollTrackStrokeStyle;
    ctx.strokeRect(NIB_PADDING + 0.5, 0.5,
                   clientLength - 2*NIB_PADDING, thickness - 1);
}

var buildHandlePath = function() {
    ctx.beginPath();
    ctx.arc(handleDistance + halfThickness + 0.5,                // x
             halfThickness,                                     // y
             halfThickness,                                     // x
             halfThickness - 0.5, Math.PI / 2, 3 * Math.PI / 2, false);
    ctx.arc(handleDistance + handleLength - halfThickness - 0.5, // x
             halfThickness,                                     // y
             halfThickness - 0.5, 3 * Math.PI / 2, Math.PI / 2, false);
    ctx.lineTo(handleDistance + halfThickness + 0.5, thickness - 0.5);
    ctx.closePath();
};
buildHandlePath();

// Paint the interior of the handle path.
var gradient = ctx.createLinearGradient(handleDistance, 0, handleDistance, thickness);
gradient.addColorStop(0, theme.scrollBarFillGradientTopStart.replace(/%a/, alpha));
gradient.addColorStop(0.4, theme.scrollBarFillGradientTopStop.replace(/%a/, alpha));
gradient.addColorStop(0.41, theme.scrollBarFillStyle.replace(/%a/, alpha));
gradient.addColorStop(0.8, theme.scrollBarFillGradientBottomStart.replace(/%a/, alpha));
gradient.addColorStop(1, theme.scrollBarFillGradientBottomStop.replace(/%a/, alpha));
ctx.fillStyle = gradient;
ctx.fill();
Declarative

```html
<form class="fl-thumbnailContainer">
  ...
</form>

.fl-thumbnailContainer {
  position: absolute;
  top: 0pt;
  left: 0pt;
  bottom: 0pt;
  overflow: auto;
  width: 185px;
  z-index: 100;
}
```
Statelessness

“The Web is broken. It can’t remember me between requests”
Statelessness

- The Web is stateless for a reason: *it scales*
- State is visible, not encapsulated

http://build.fluidproject.org:8095/engage/artifacts/view.html?
  accessNumber=M2000.38.97&db=mccord&lang=en

Thursday, September 2, 2010
Interoperable

- Web formats are:
  - Plain text
  - Declarative
  - Openly published and standardized

- This means they are adaptable and extensible