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

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

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

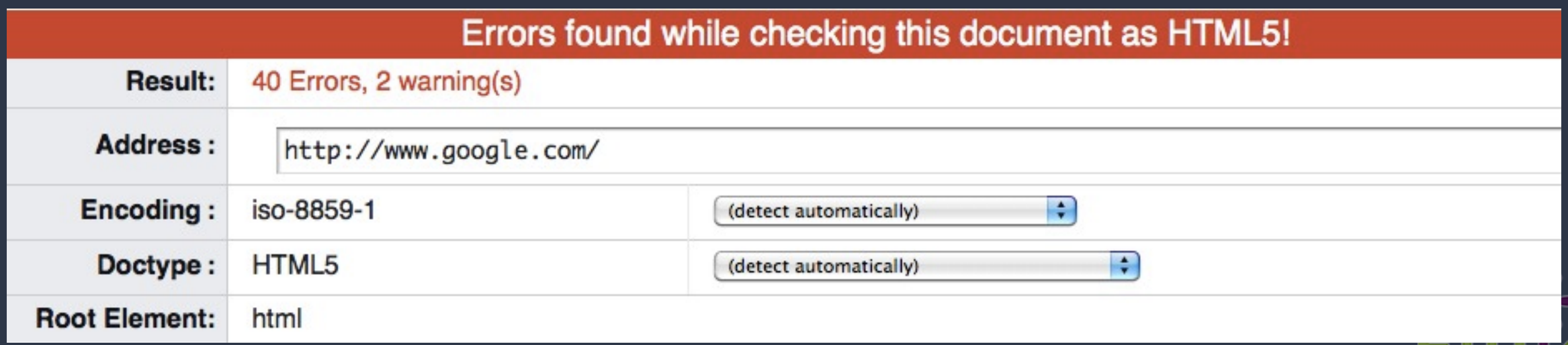
Tools for Evaluating Accessibility

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



Accessibility Checkers

aChecker

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

Wave

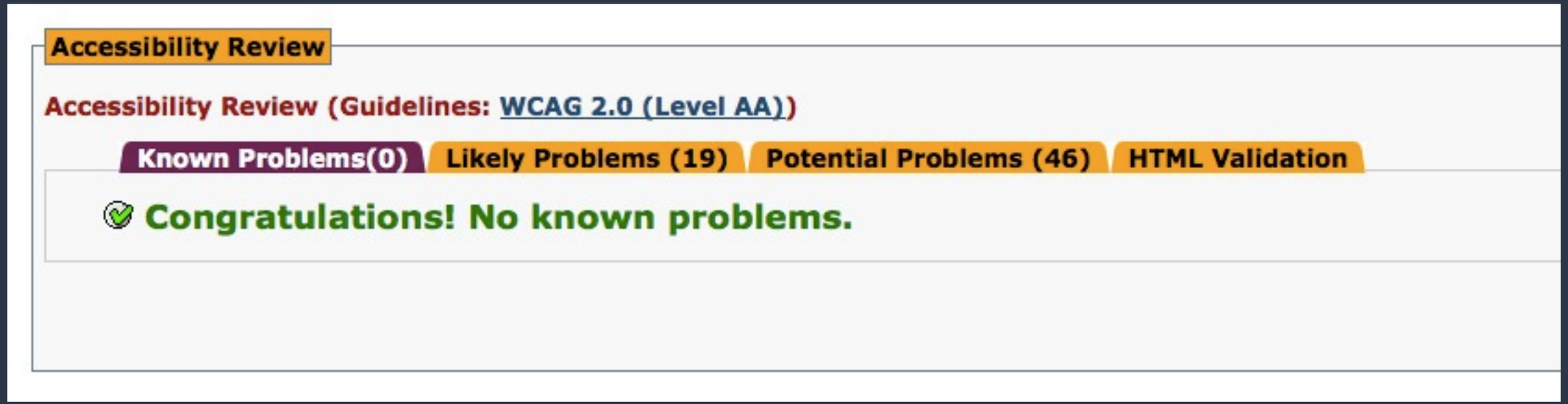
http://jigsaw.w3.org/css-validator/

Fluid

Accessibility Checkers

aChecker

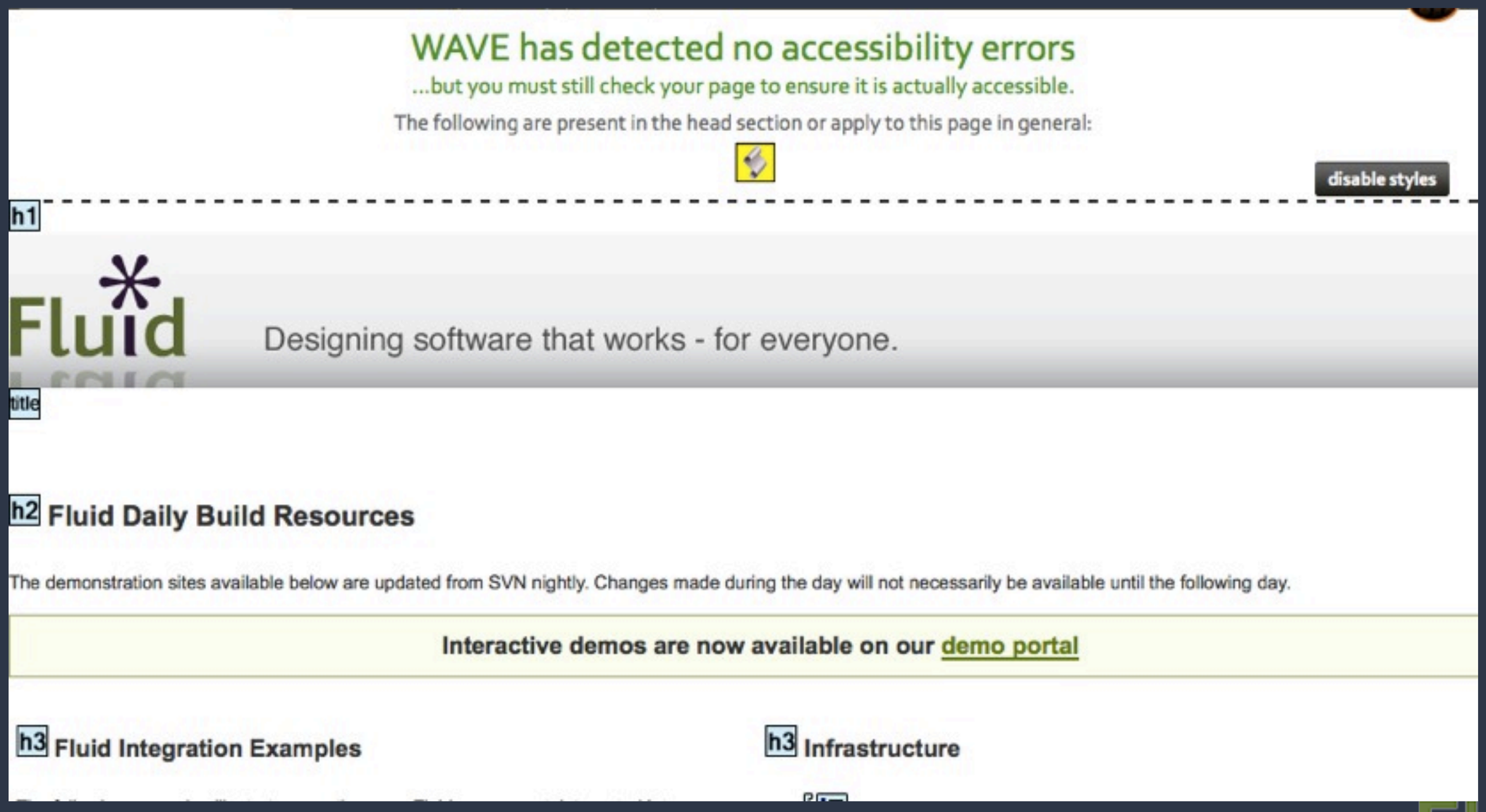
http://achecker.ca/checker/index.php



Accessibility Checkers

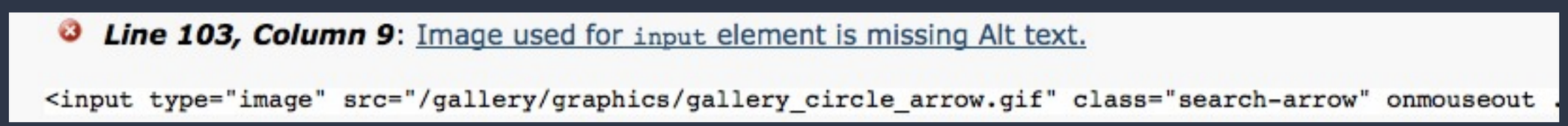
Wave

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



Accessibility Checkers:
What They Do

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



Accessibility Checkers: Limitations

- No Magic
  - Is the alt text meaning full?
- 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

This is example text. **Some of it bolded.**
Some of it italicized.

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

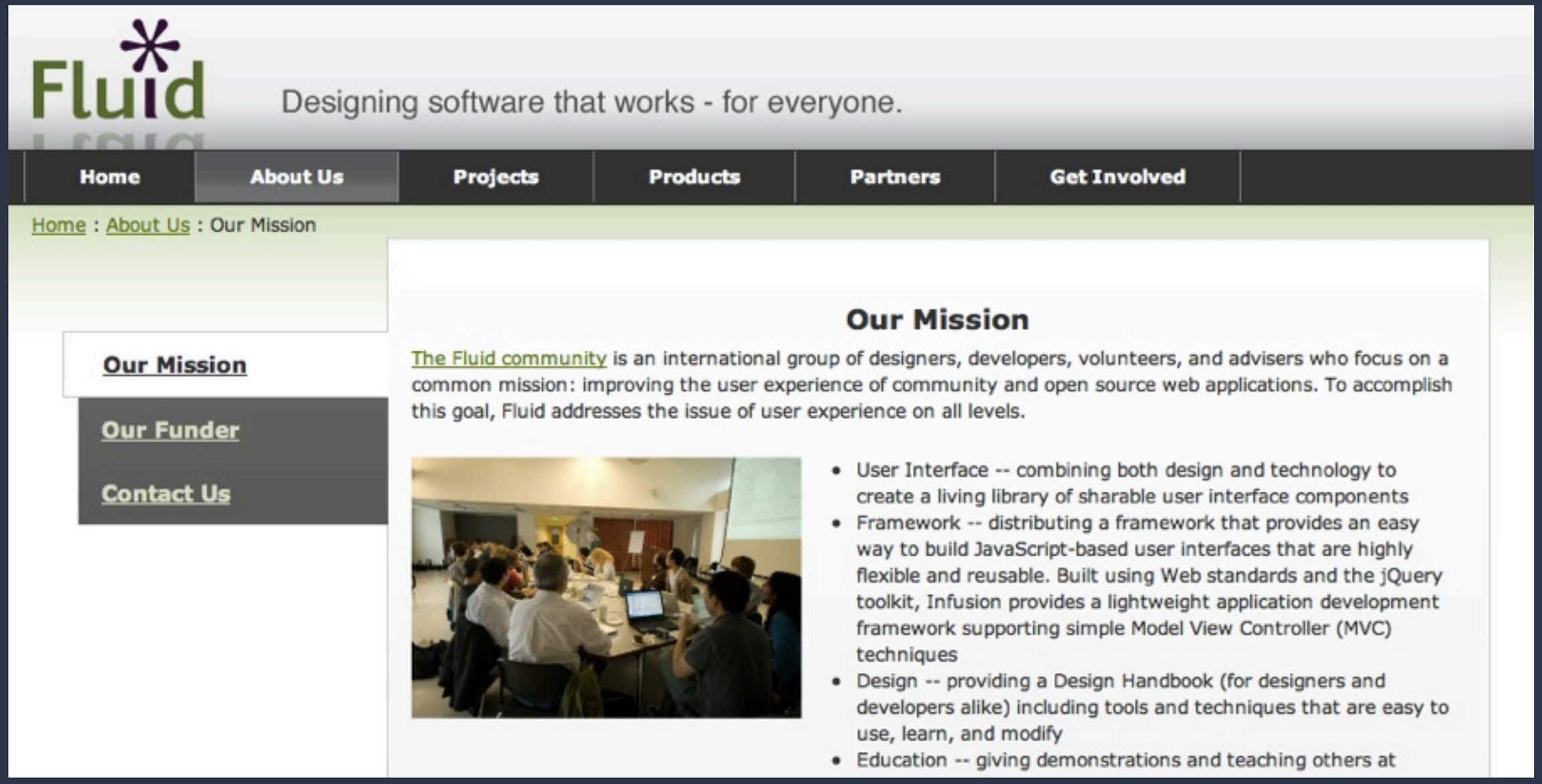
http://www.snook.ca/technical/colour_contrast/colour.html

Accessible Markup

Concepts of HTML Accessibility

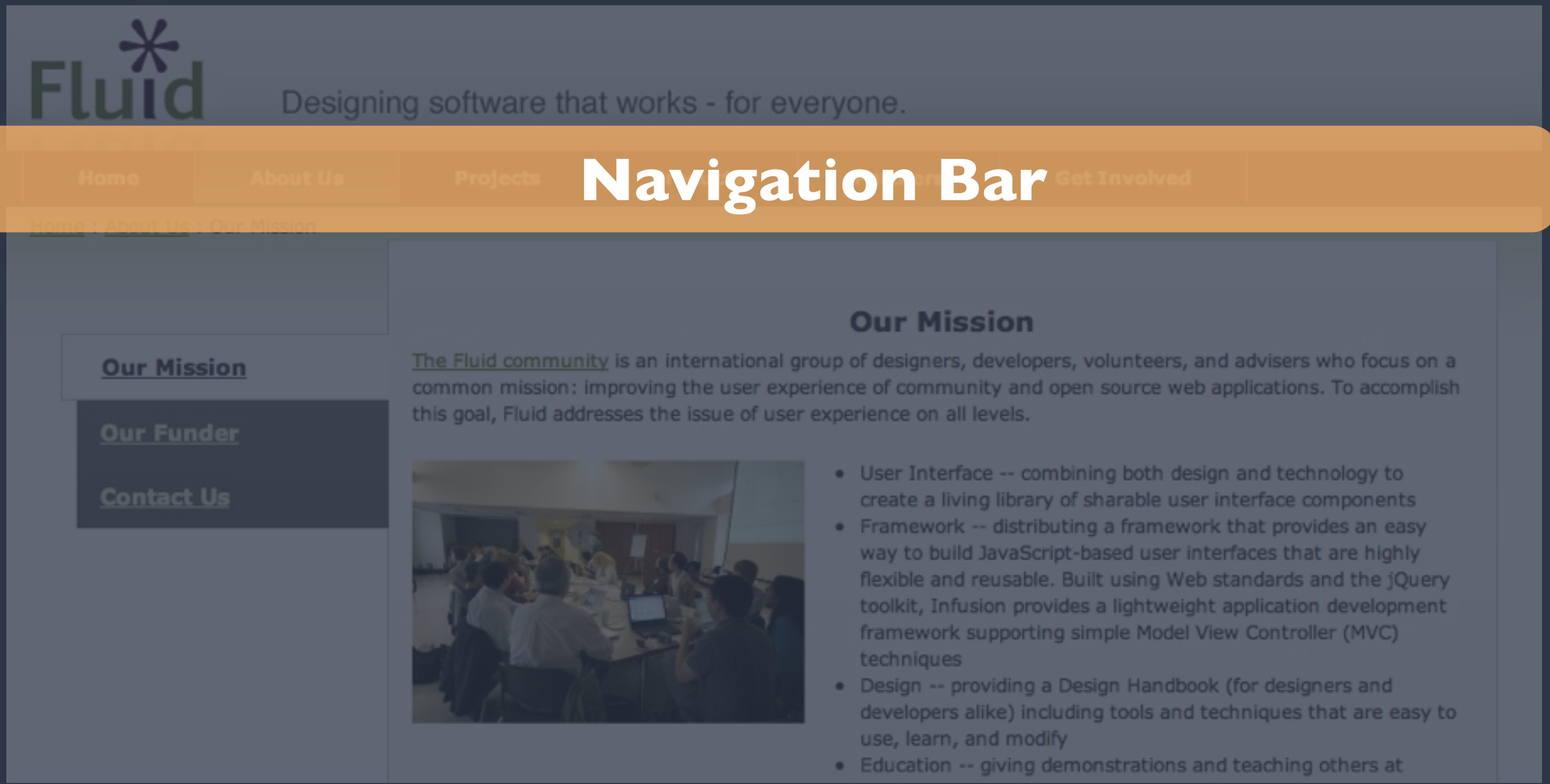
- Label as much as you can
- Use semantic markup to describe your page
- Design in layers
- Textual alternatives
- Clearly delineate navigation from content

Avoiding Repetition



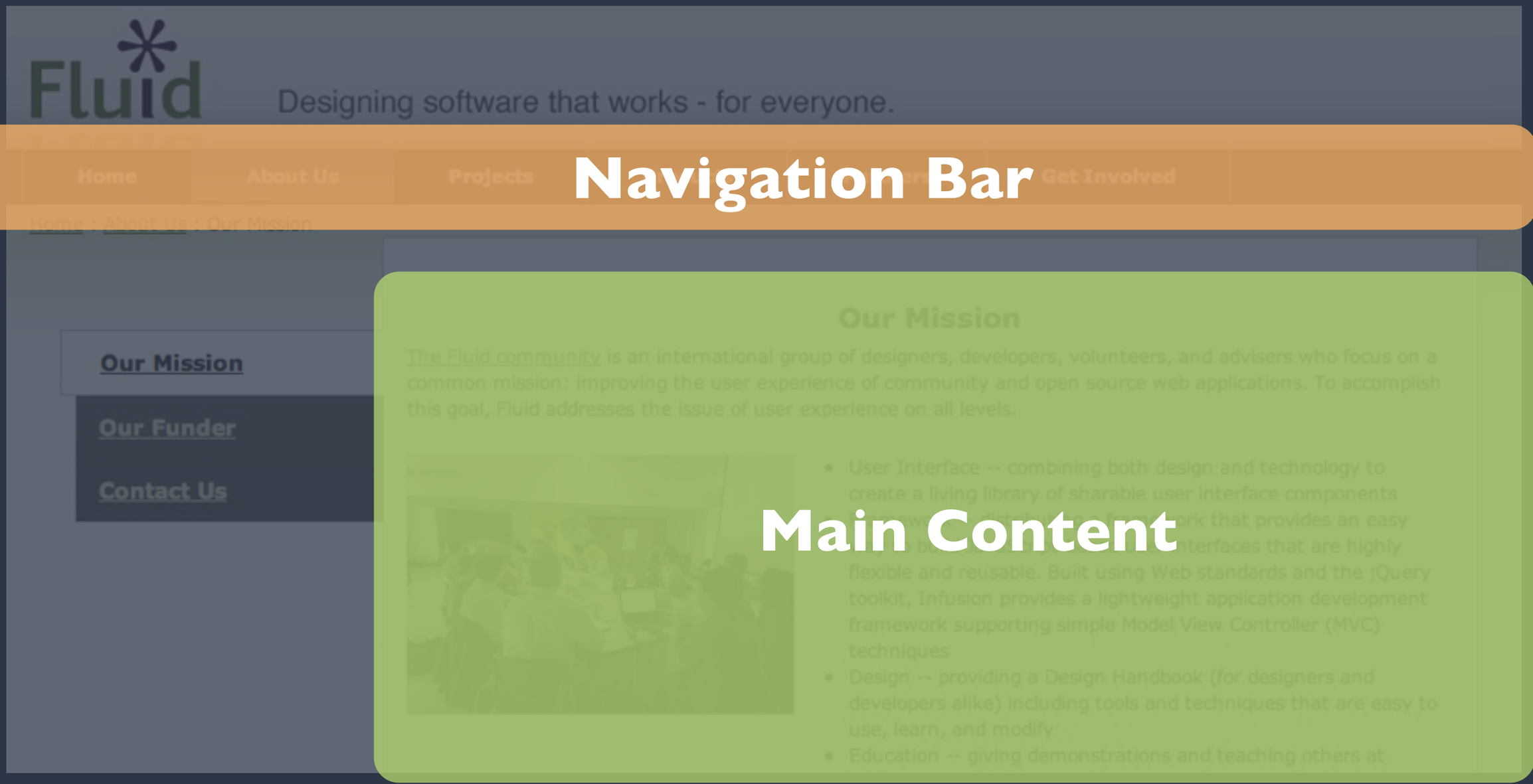
Fluid

Avoiding Repetition

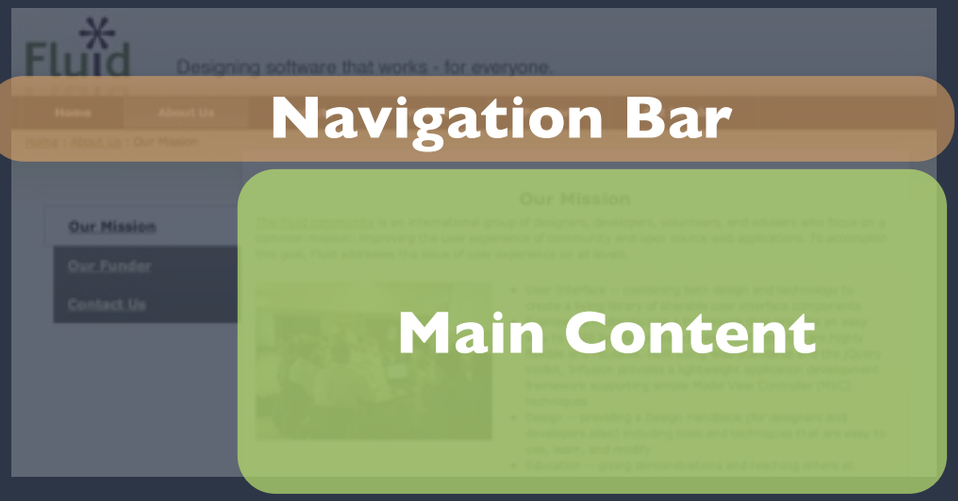


Fluid

Avoiding Repetition



Avoid Repetition



Skip Links

```
<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 -->
```

Fluid

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

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

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

- "Infuison 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

Fluid

Navigable Headings

Level One

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

- "Infuison 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

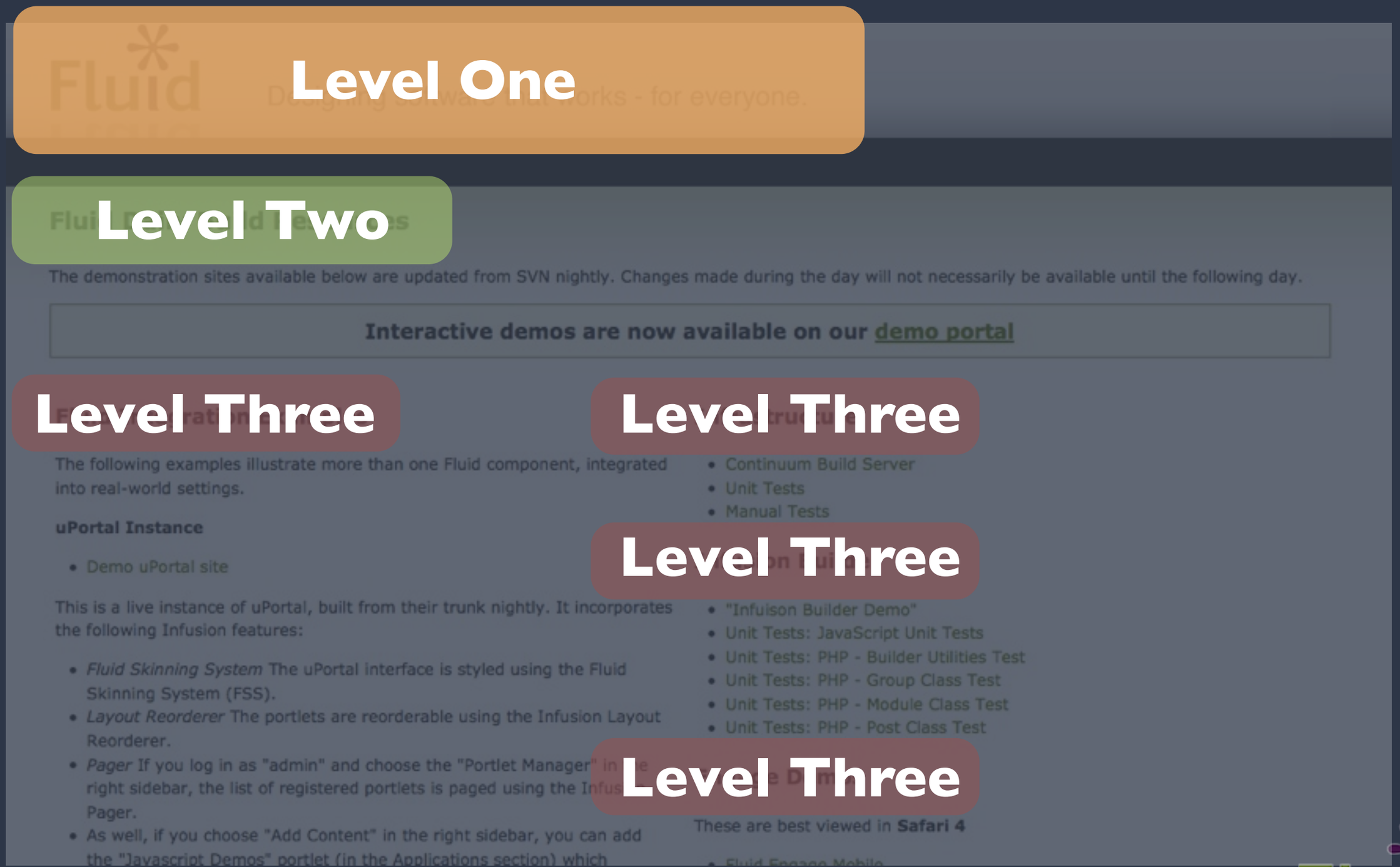
- "Infuison Builder Demo"
- Unit Tests: JavaScript Unit Tests
- Unit Tests: PHP - Builder Utilities Test
- Unit Tests: PHP - Group Class Test
- Unit Tests: PHP - Module Class Test
- Unit Tests: PHP - Post Class Test

Engage Demos

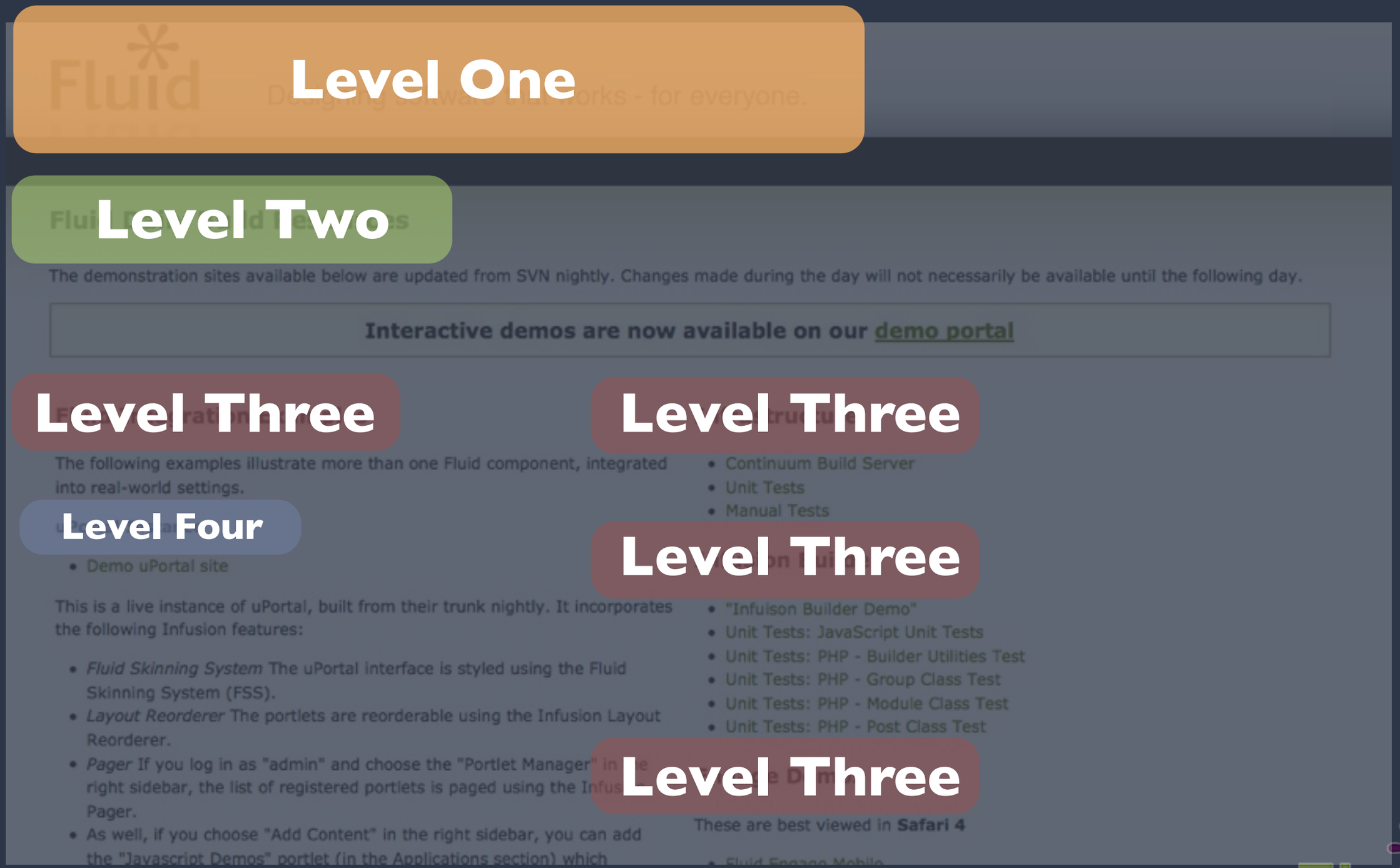
These are best viewed in **Safari 4**

- Fluid Engage Mobile

Navigable Headings



Navigable Headings



Navigable Headings

<H1>

<H2>

<H3>

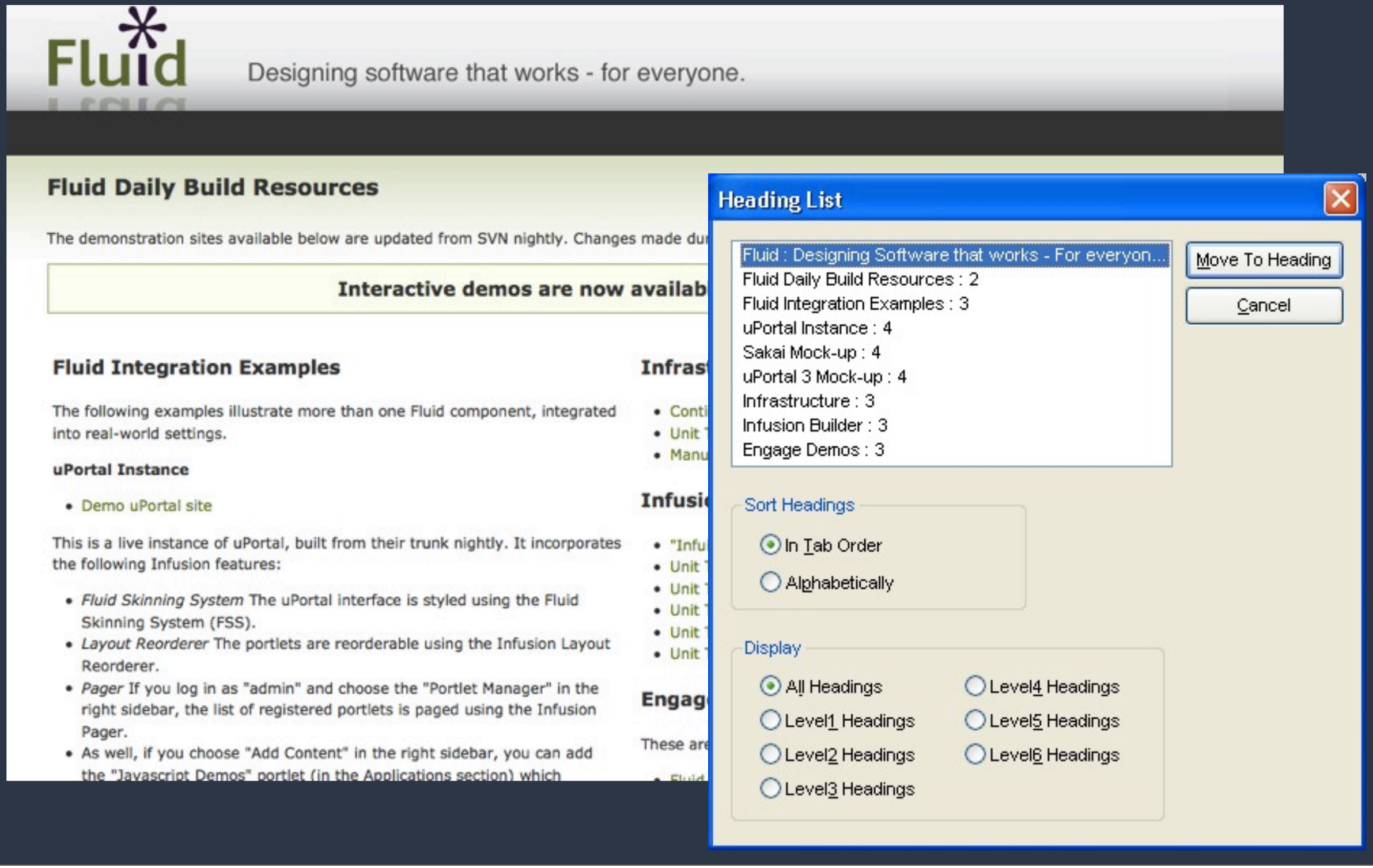
<H3>

<H4>

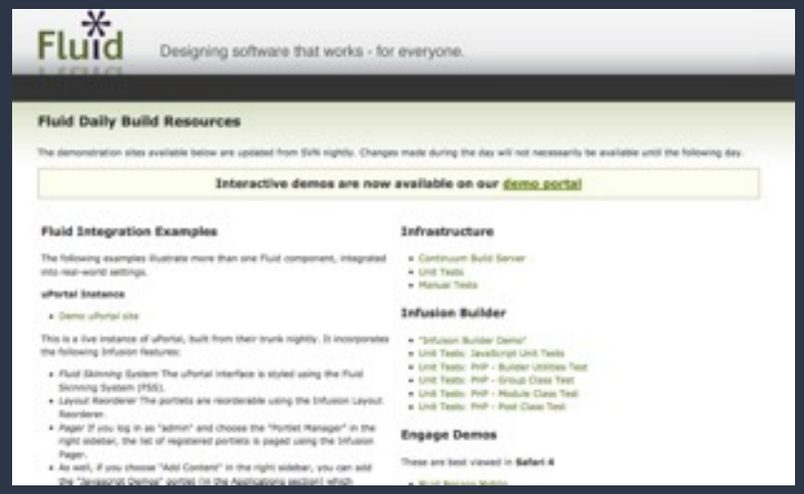
<H3>

<H3>

Navigating Headings

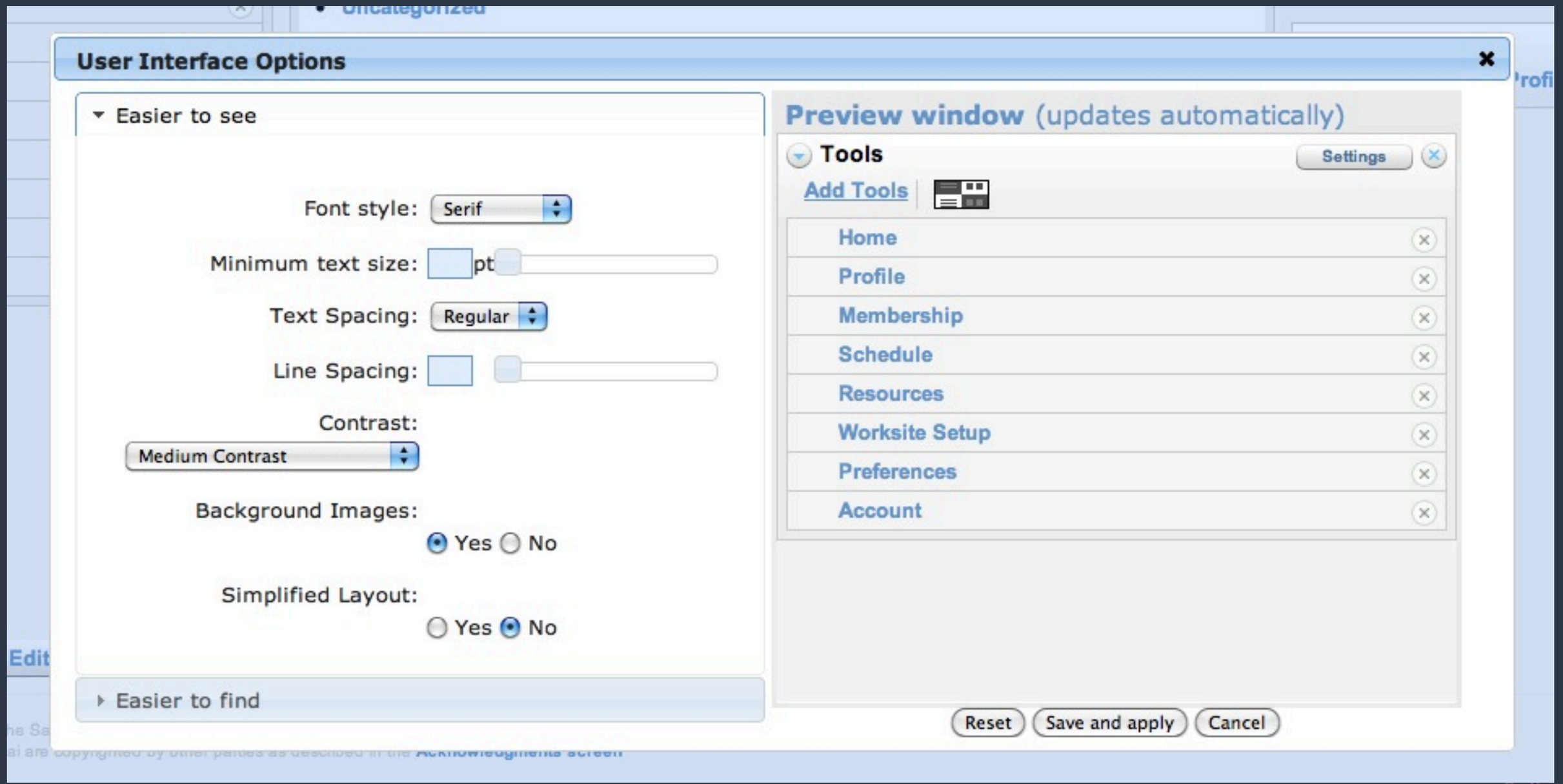


Navigable Headings

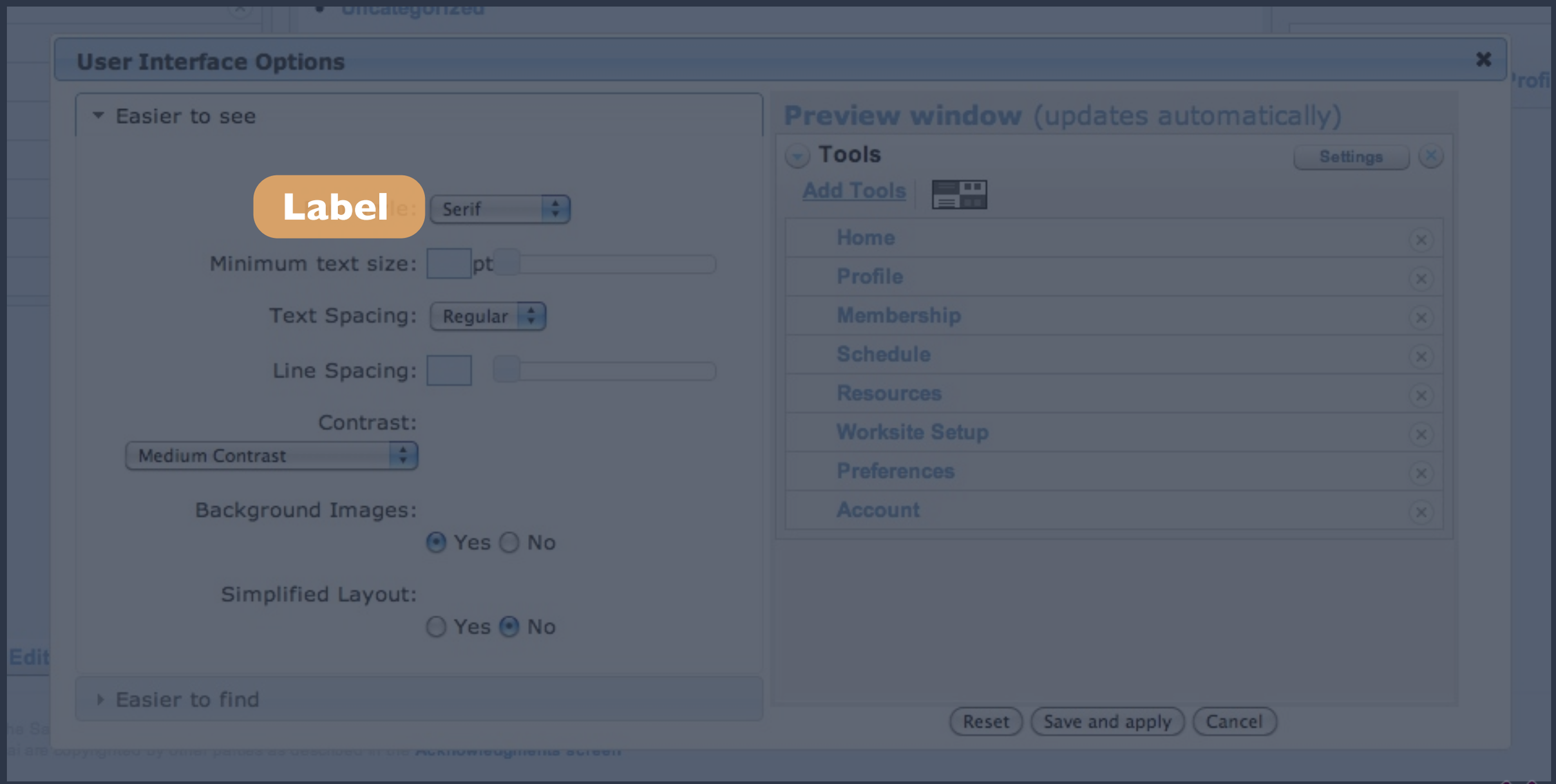



```
<body>
  <h1>Fluid: Designing Software that works - For everyone.</h1>
  <h2>Fluid Daily Build Resources</h2>
    <div>
      <div class="fl-col">
        <h3>Fluid Integration Examples</h3>
          <h4>uPortal Instance</h4>
          <-- Content goes here -->
          <h4>Sakai Mock-up</h4>
        </div>
        <div class="fl-col">
          <h3>Infrastructure</h3>
          ...
```

Labelling Forms

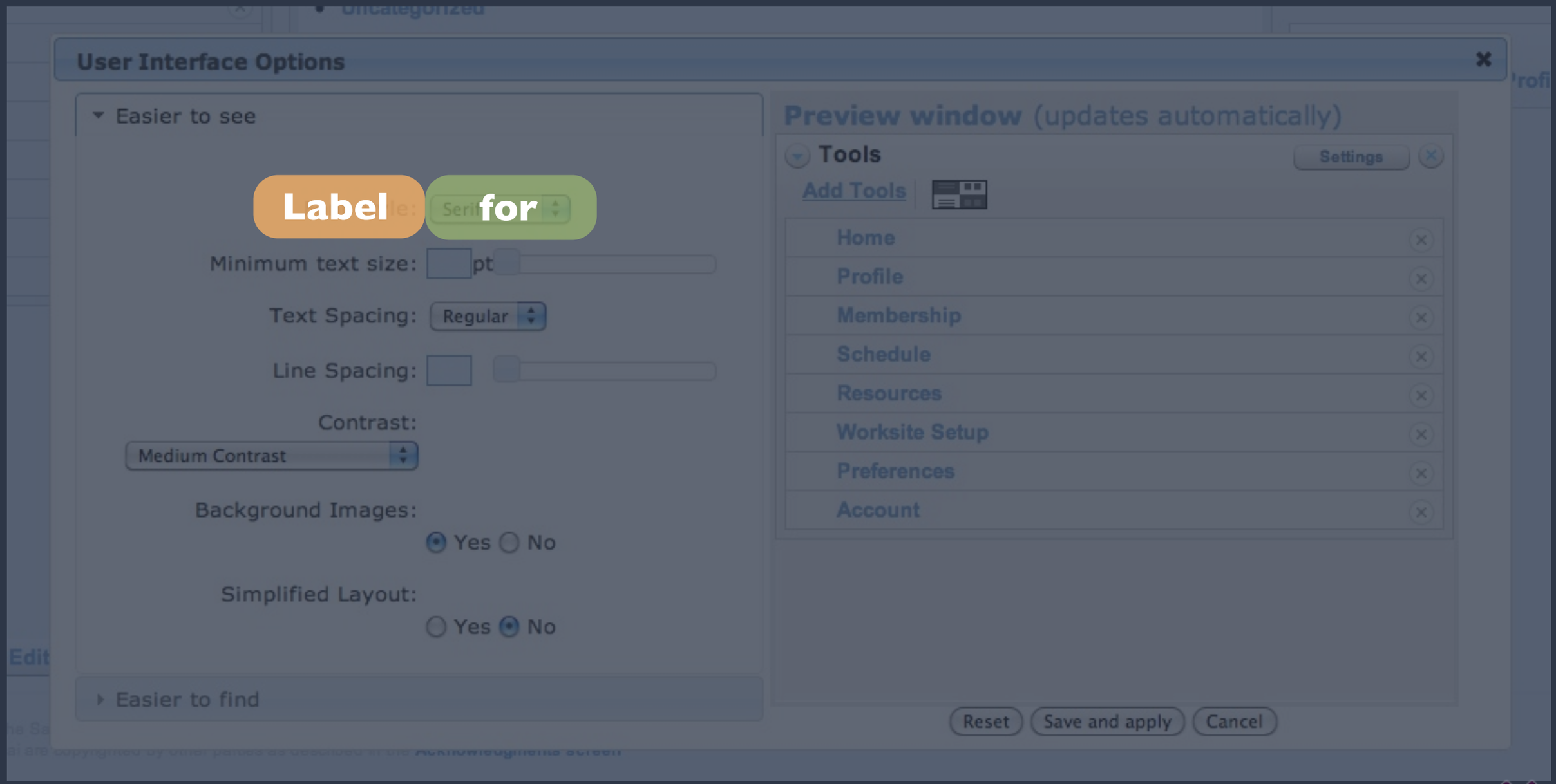


Labelling Forms



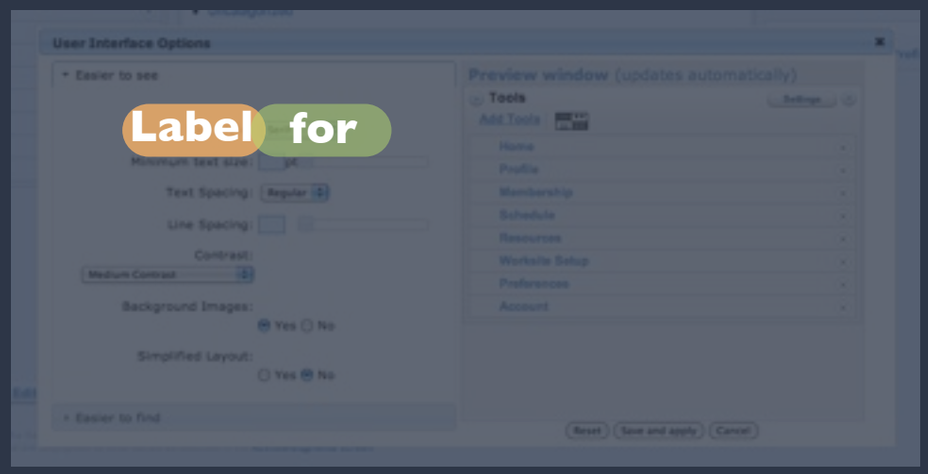
Fluid

Labelling Forms



Fluid

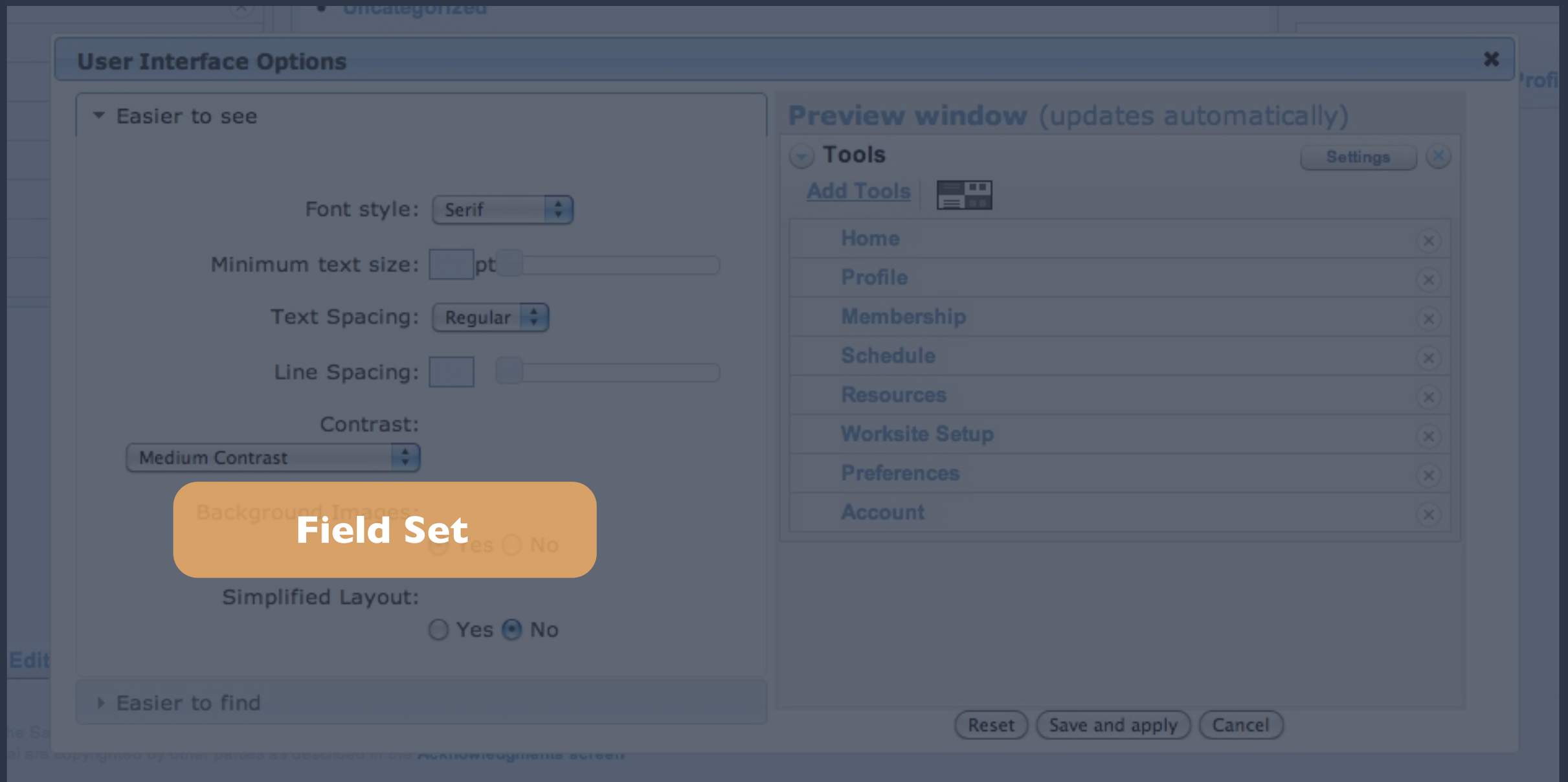
Labelling Forms




```
<li>
    <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>
<li>
```

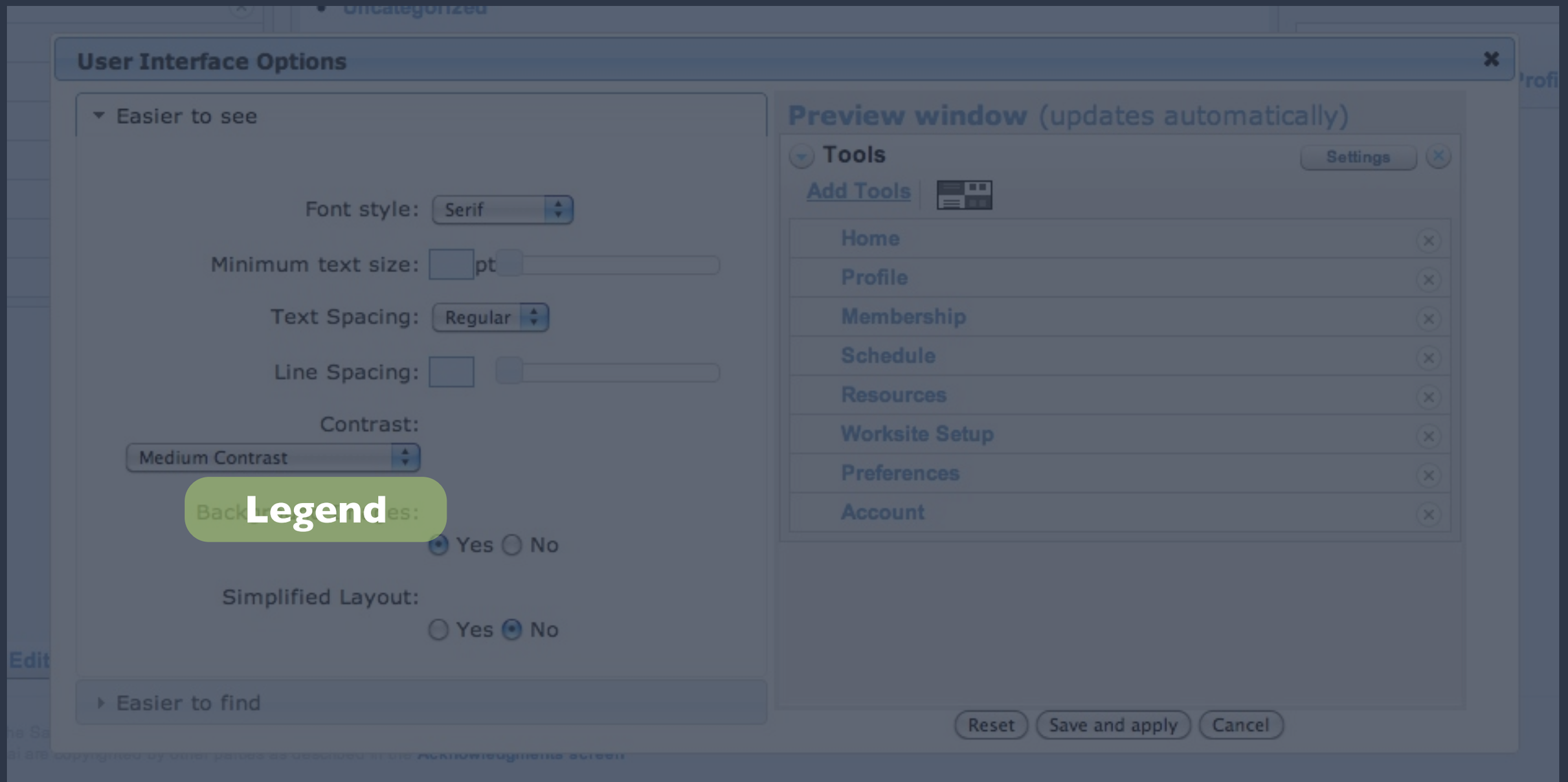
Fluid

Grouping Forms



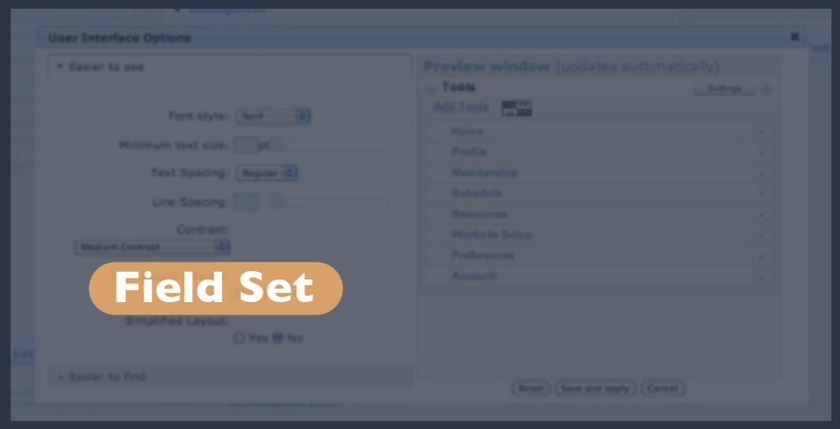
Fluid

Grouping Forms



Fluid

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>
```

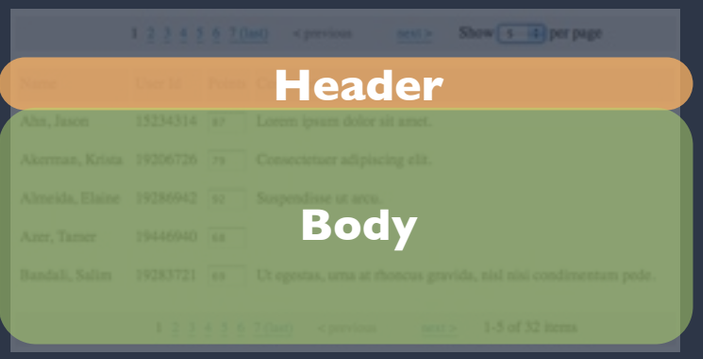
Fluid

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

Table Structure




```
<table summary="An editable table of student grades and
                instructor comments.">
  <thead>
    <tr>
      <th>Name</th>
      <th abbr="I.D.”>User ID</th>
      <th>Points</th>
      <td>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

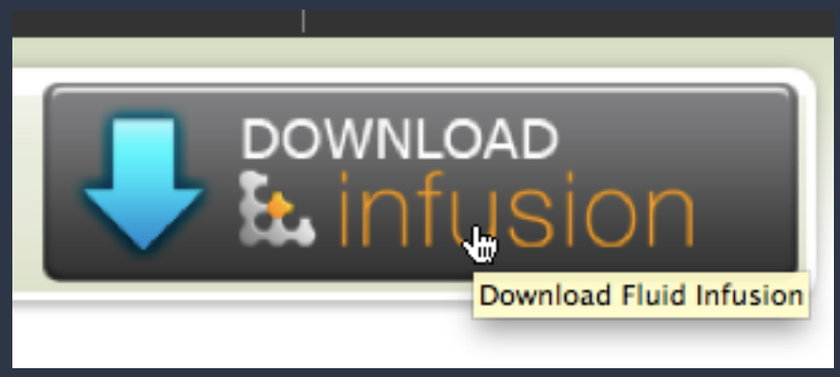
- Be accurate and equivalent in presenting the same content and function as presented by the image.
- Be succinct. Typically no more than a few words are necessary.
- NOT be redundant or provide the exact same information as text within the context.
- NOT use the phrases "image of ..." or "graphic of ..." 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

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

Background Images + Real Text

```
<a href="http://fluidproject.org/products/infusion/download-infusion"
   title="Download Fluid Infusion">
    Download Infusion
</a>
```

Fluid

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

Screen readers
Screen magnifiers
On-screen keyboards

OS AT APIs

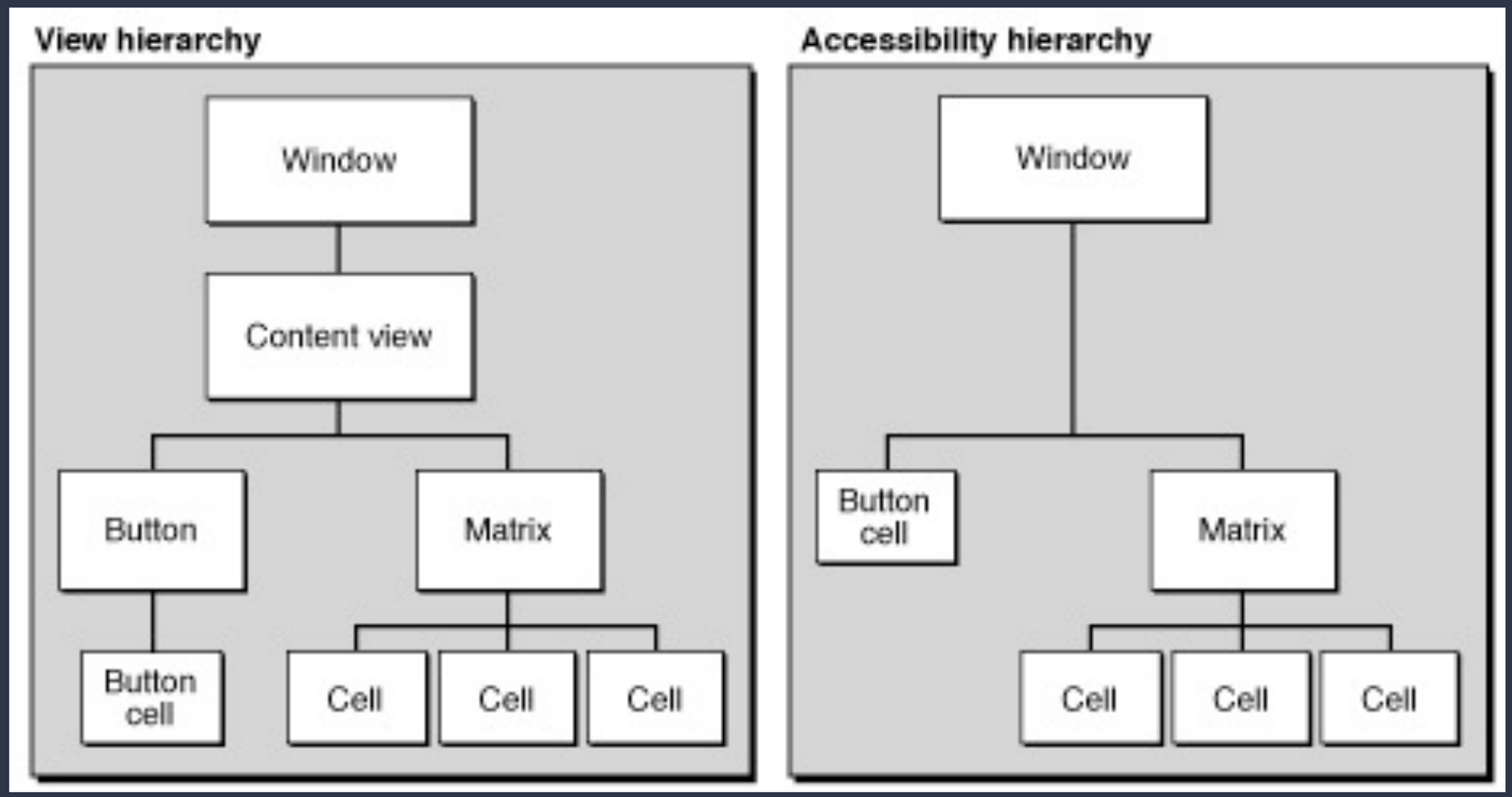
- A channel for UI introspection
- What's on screen?
- How are things labelled, organized, etc.?
- What states are things in?

- 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



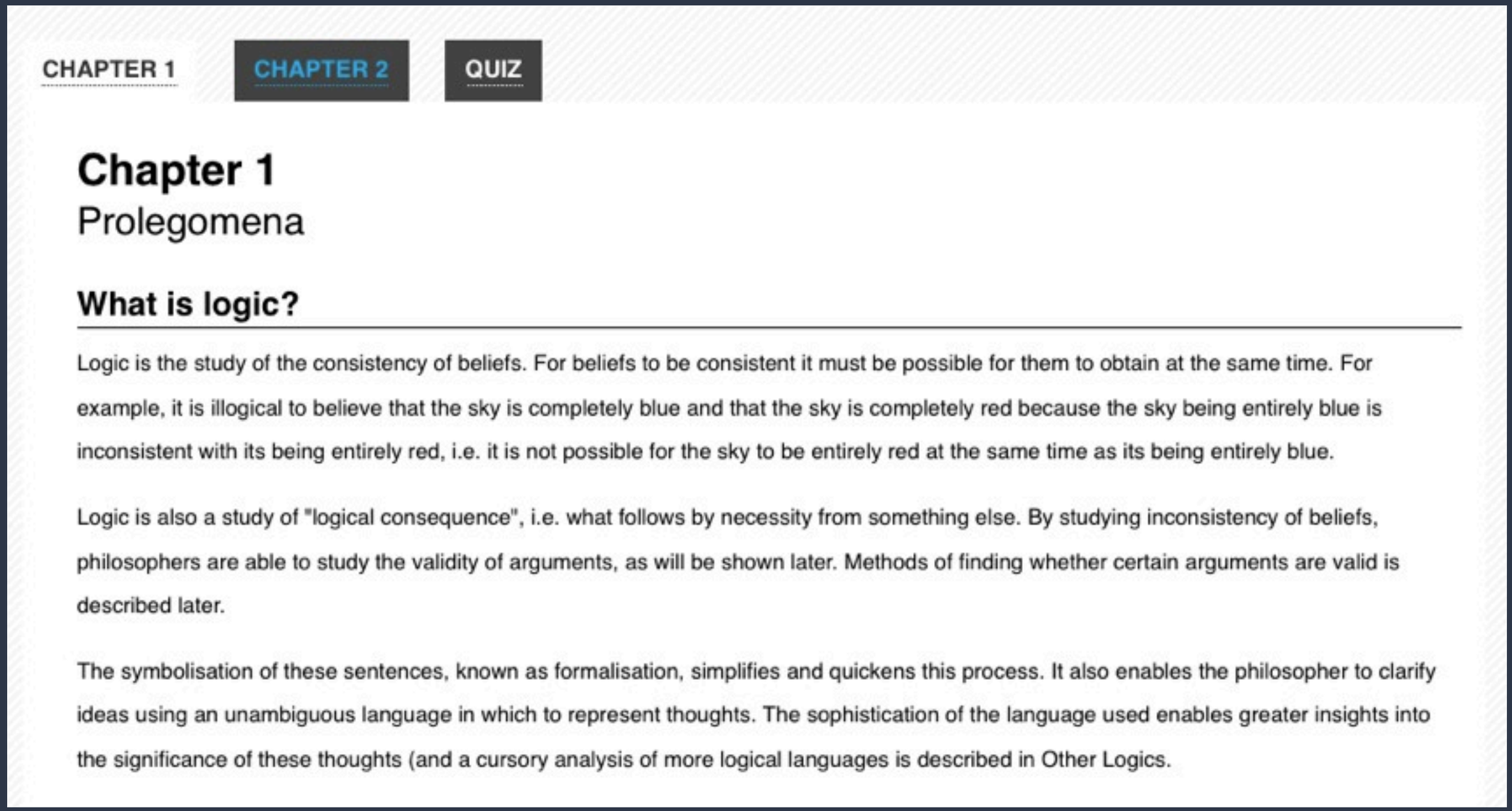
Fluid

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>
```

Opaque Markup: Tabs



Fluid

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

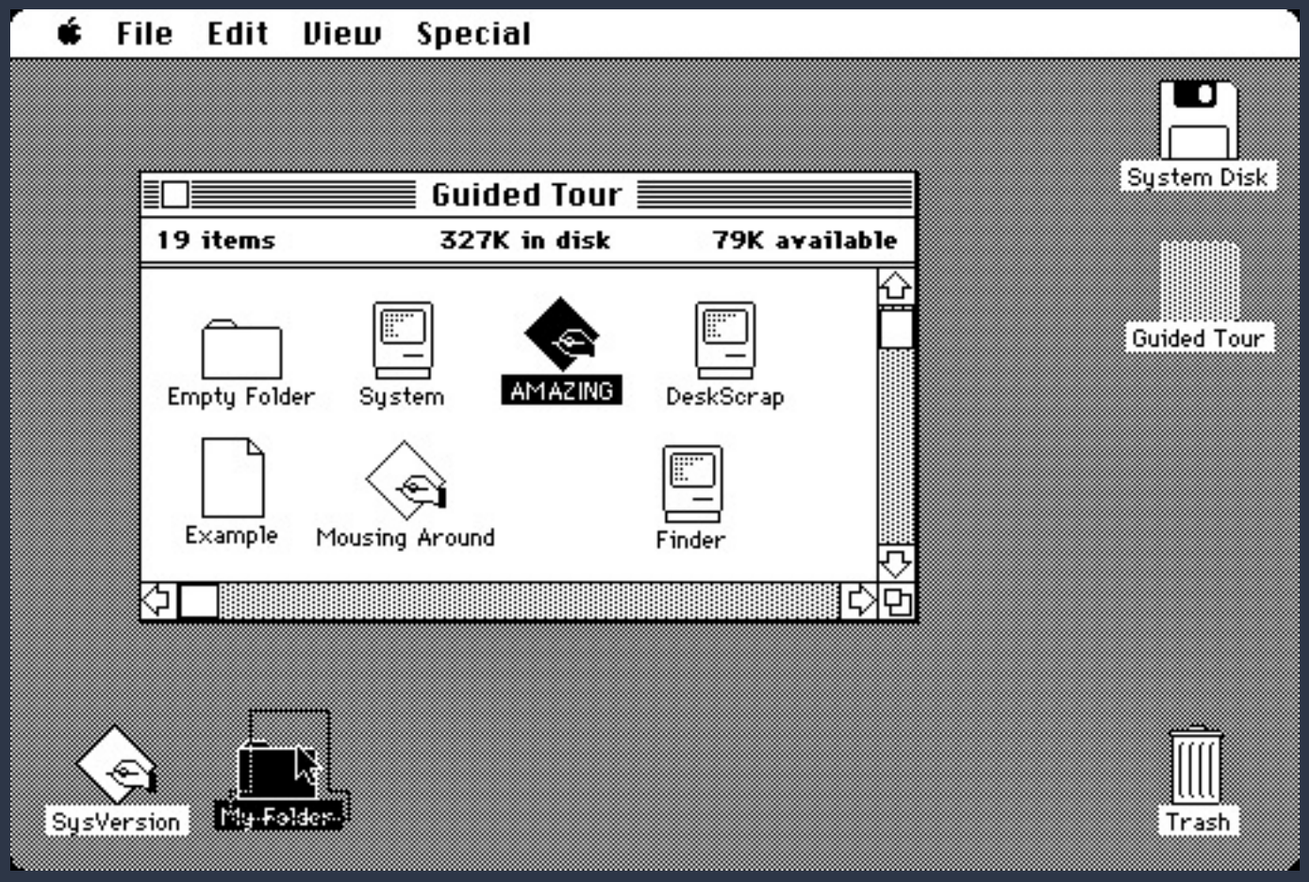
Make it work without JavaScript

```
<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>
```

Adding ARIA in Code

```
// 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);
});
```

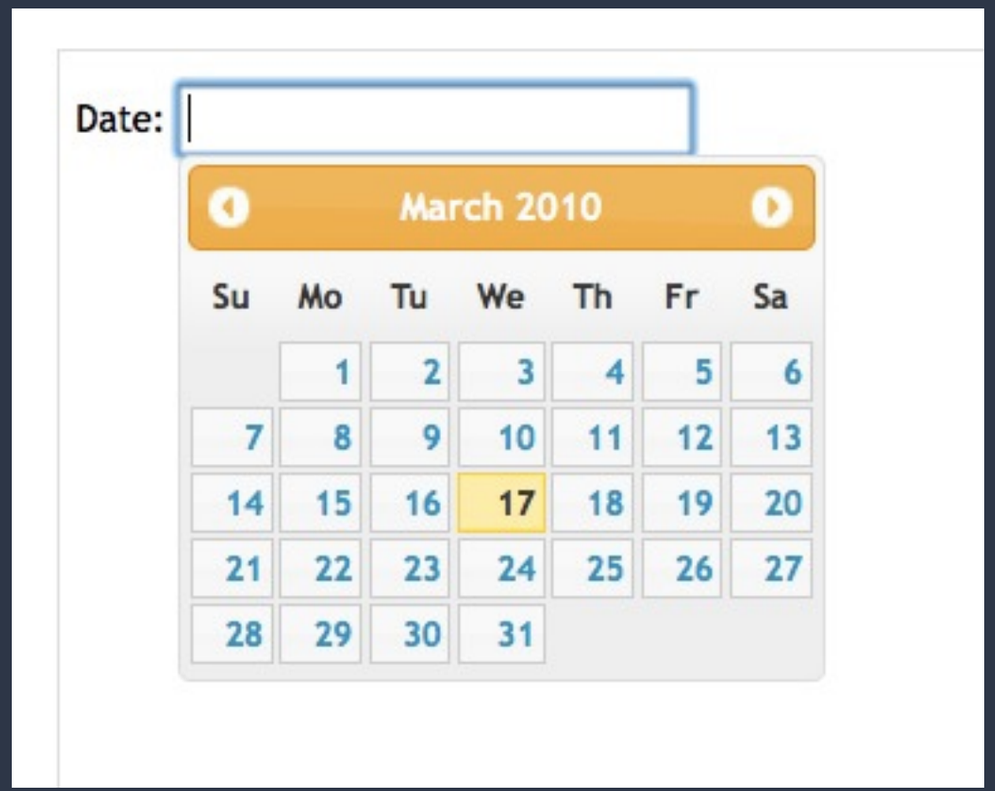
The Problem with Roles



Roles are driven by 1980's era desktop widgets

Fluid

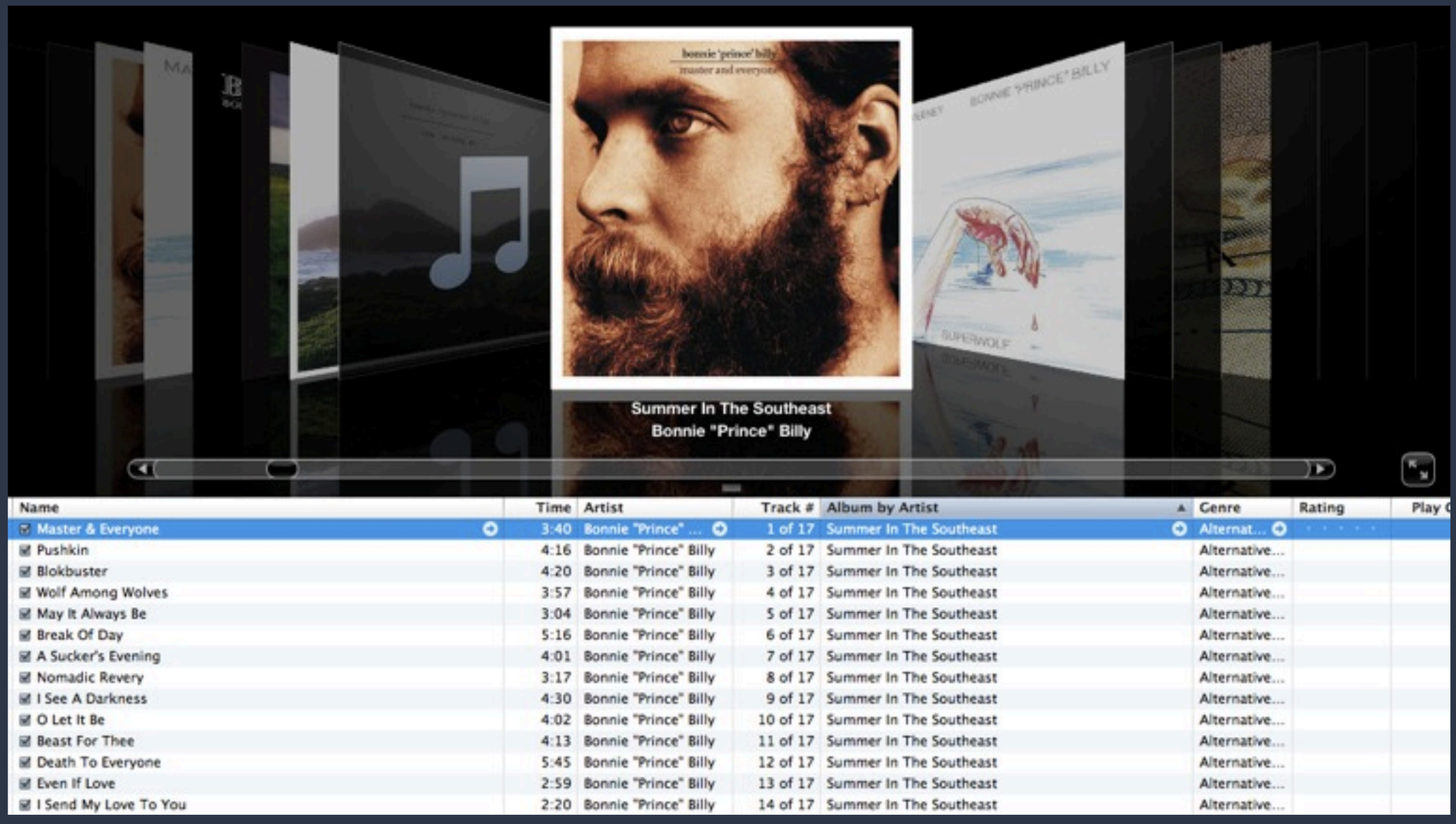
The Problem with Roles



The Web is driving hybrid UIs

Fluid

The Problem with Roles

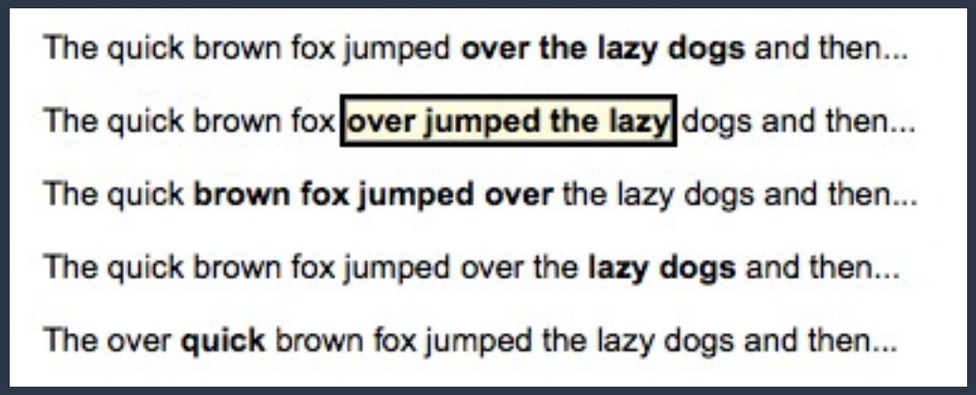


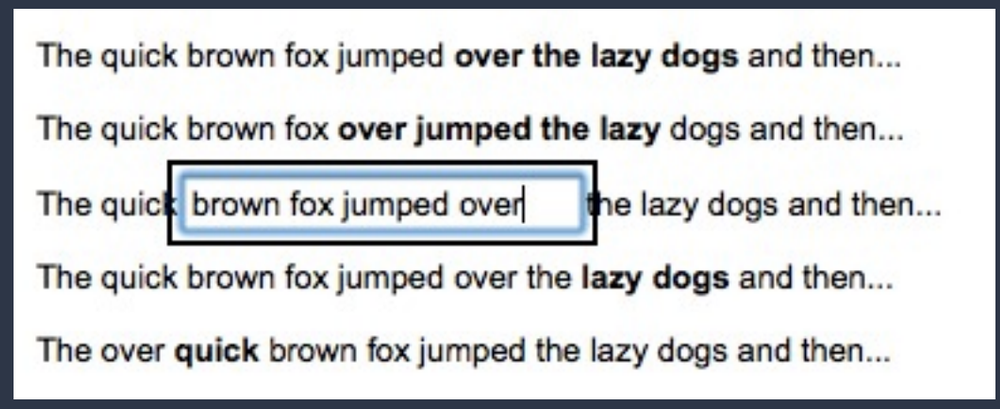
Name	Time	Artist	Track #	Album by Artist	Genre	Rating	Play C
Master & Everyone	3:40	Bonnie "Prince" ...	1 of 17	Summer In The Southeast	Alternat...		
Pushkin	4:16	Bonnie "Prince" Billy	2 of 17	Summer In The Southeast	Alternative...		
Blokbuster	4:20	Bonnie "Prince" Billy	3 of 17	Summer In The Southeast	Alternative...		
Wolf Among Wolves	3:57	Bonnie "Prince" Billy	4 of 17	Summer In The Southeast	Alternative...		
May It Always Be	3:04	Bonnie "Prince" Billy	5 of 17	Summer In The Southeast	Alternative...		
Break Of Day	5:16	Bonnie "Prince" Billy	6 of 17	Summer In The Southeast	Alternative...		
A Sucker's Evening	4:01	Bonnie "Prince" Billy	7 of 17	Summer In The Southeast	Alternative...		
Nomadic Revery	3:17	Bonnie "Prince" Billy	8 of 17	Summer In The Southeast	Alternative...		
I See A Darkness	4:30	Bonnie "Prince" Billy	9 of 17	Summer In The Southeast	Alternative...		
O Let It Be	4:02	Bonnie "Prince" Billy	10 of 17	Summer In The Southeast	Alternative...		
Beast For Thee	4:13	Bonnie "Prince" Billy	11 of 17	Summer In The Southeast	Alternative...		
Death To Everyone	5:45	Bonnie "Prince" Billy	12 of 17	Summer In The Southeast	Alternative...		
Even If Love	2:59	Bonnie "Prince" Billy	13 of 17	Summer In The Southeast	Alternative...		
I Send My Love To You	2:20	Bonnie "Prince" Billy	14 of 17	Summer In The Southeast	Alternative...		

... even on the desktop

Fluid

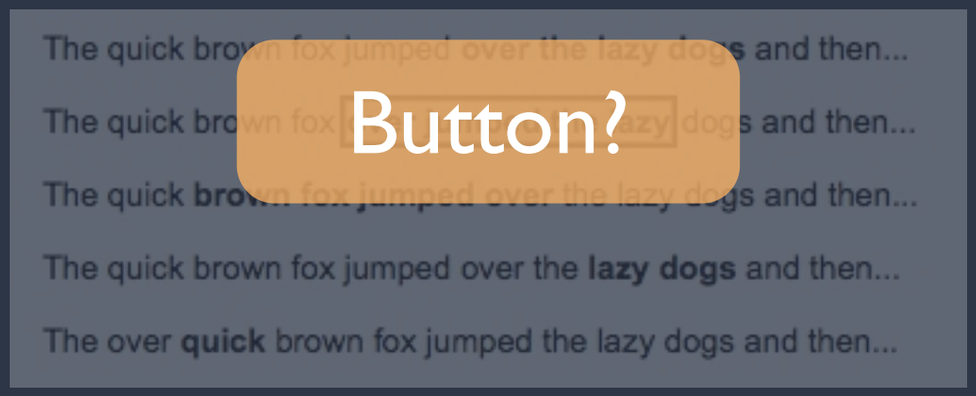
Analyzing User Interfaces

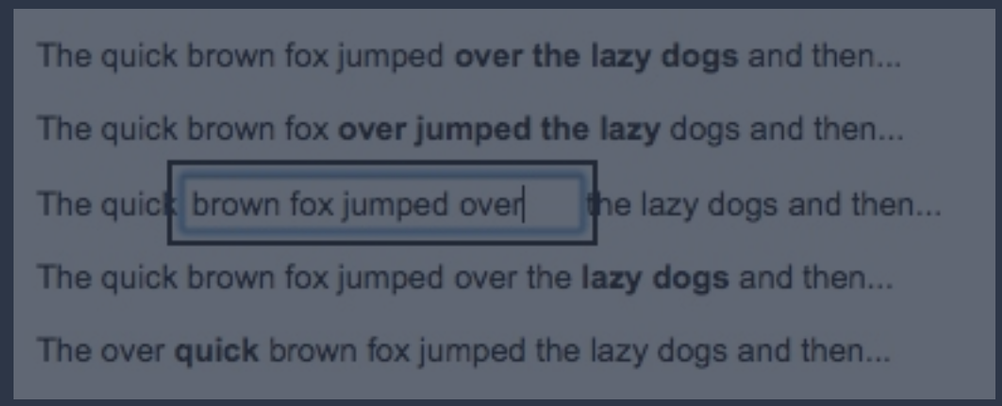




Fluid

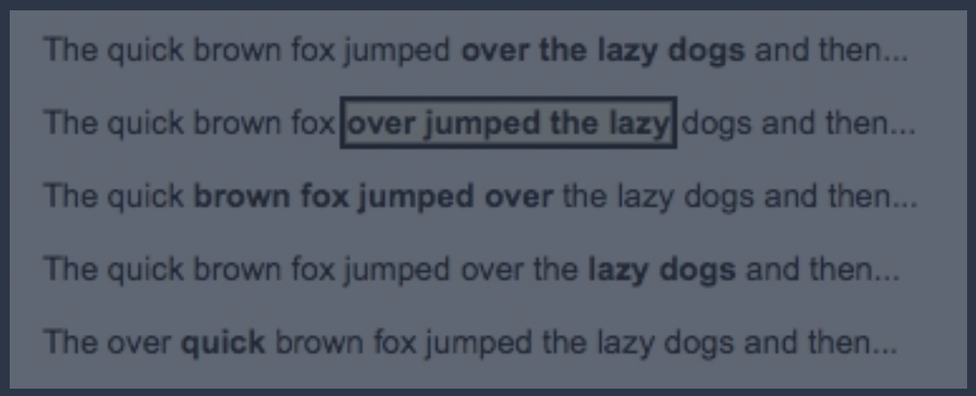
Inline Edit Roles





Fluid

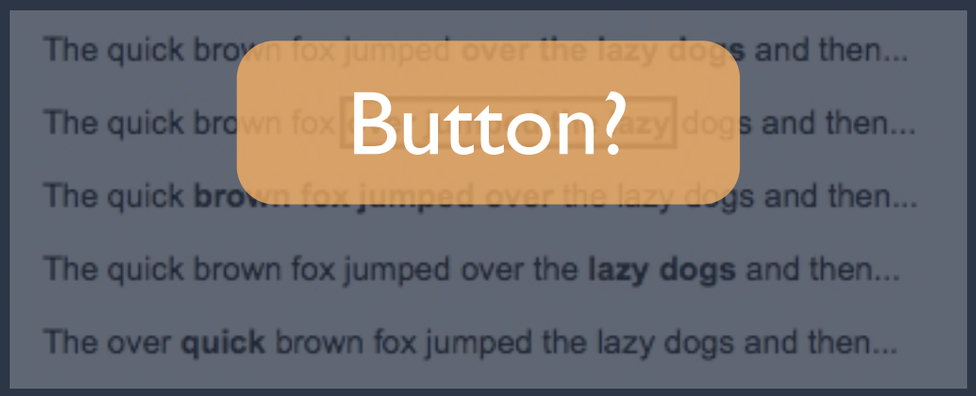
Inline Edit Roles



Text Field?

Fluid

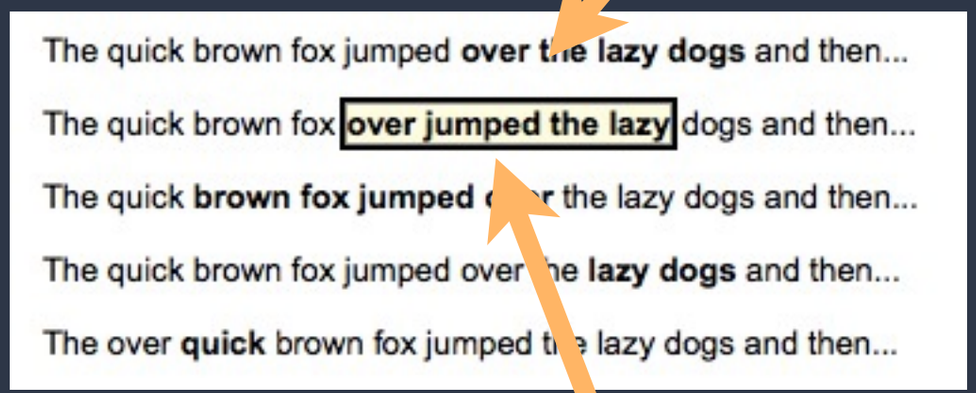
Inline Edit Roles



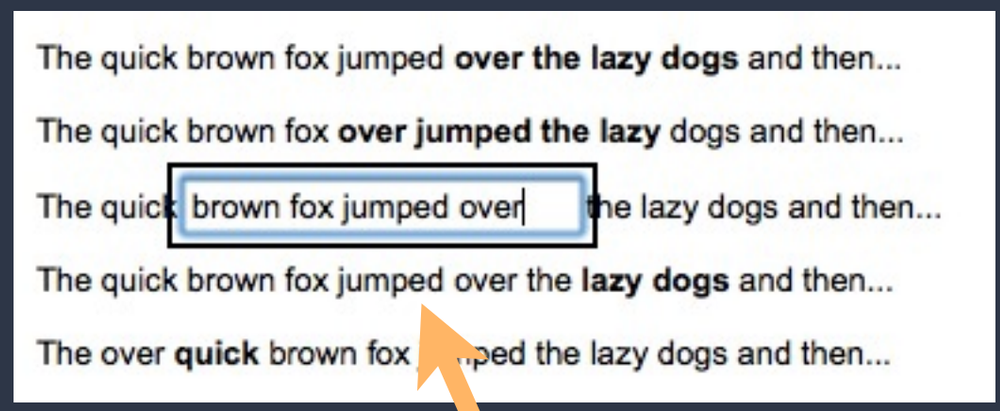
Fluid

Inline Edit Behaviours

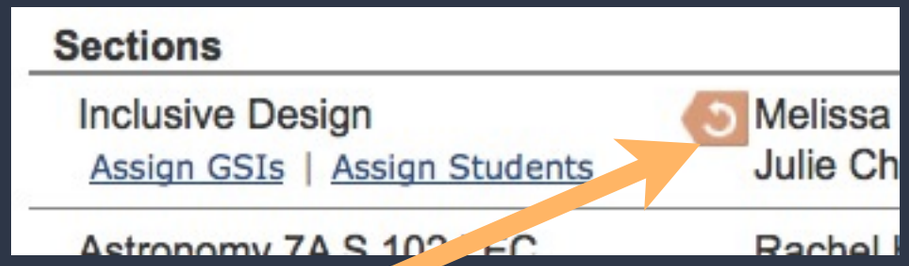
Read-only



Activatable



Editable



Undoable

Fluid

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

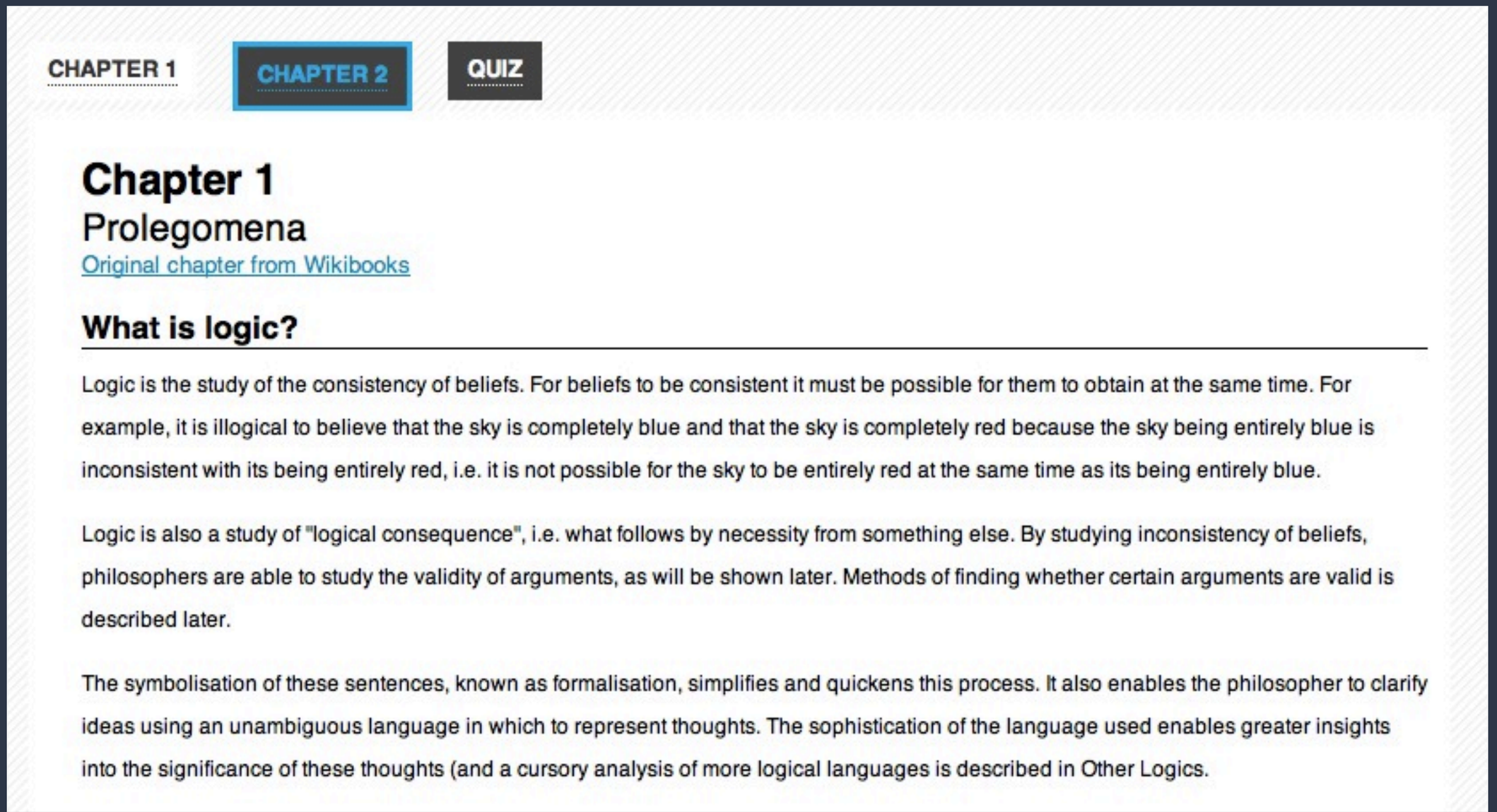
Keyboard Conventions

- 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.

Fluid

Keyboard a11y: Tabs



Fluid

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:

  http://fluidproject.org/blog/2008/01/09/
  getting-setting-and-removing-tabindex-values-with-javascript/

```
// 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

http://developer.mozilla.org/en/docs/AJAX:WAI_ARIA_Live_Regions

Building with FSS

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

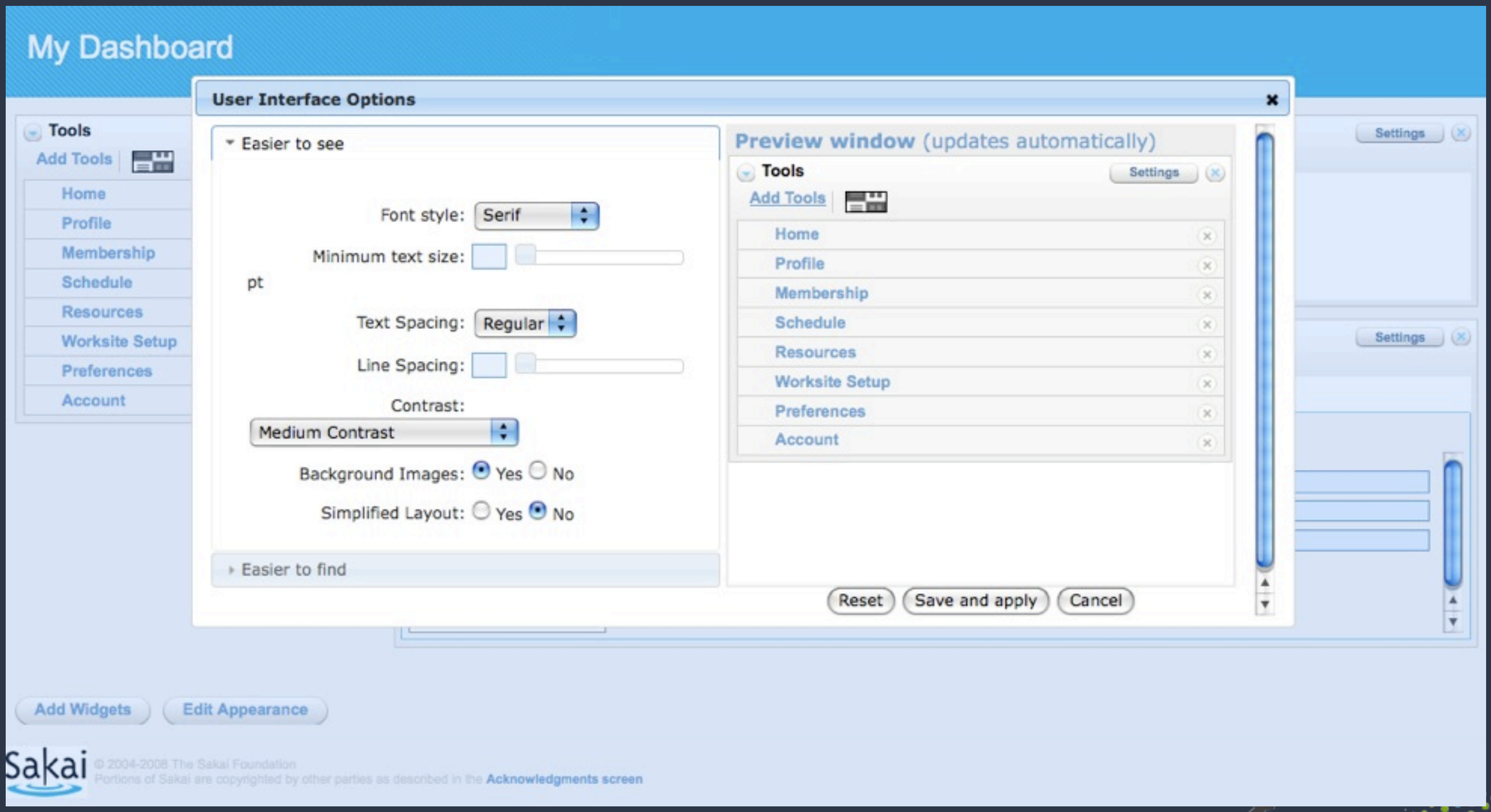
http://wiki.fluidproject.org/x/96M7

Fluid

Dynamic Skinning

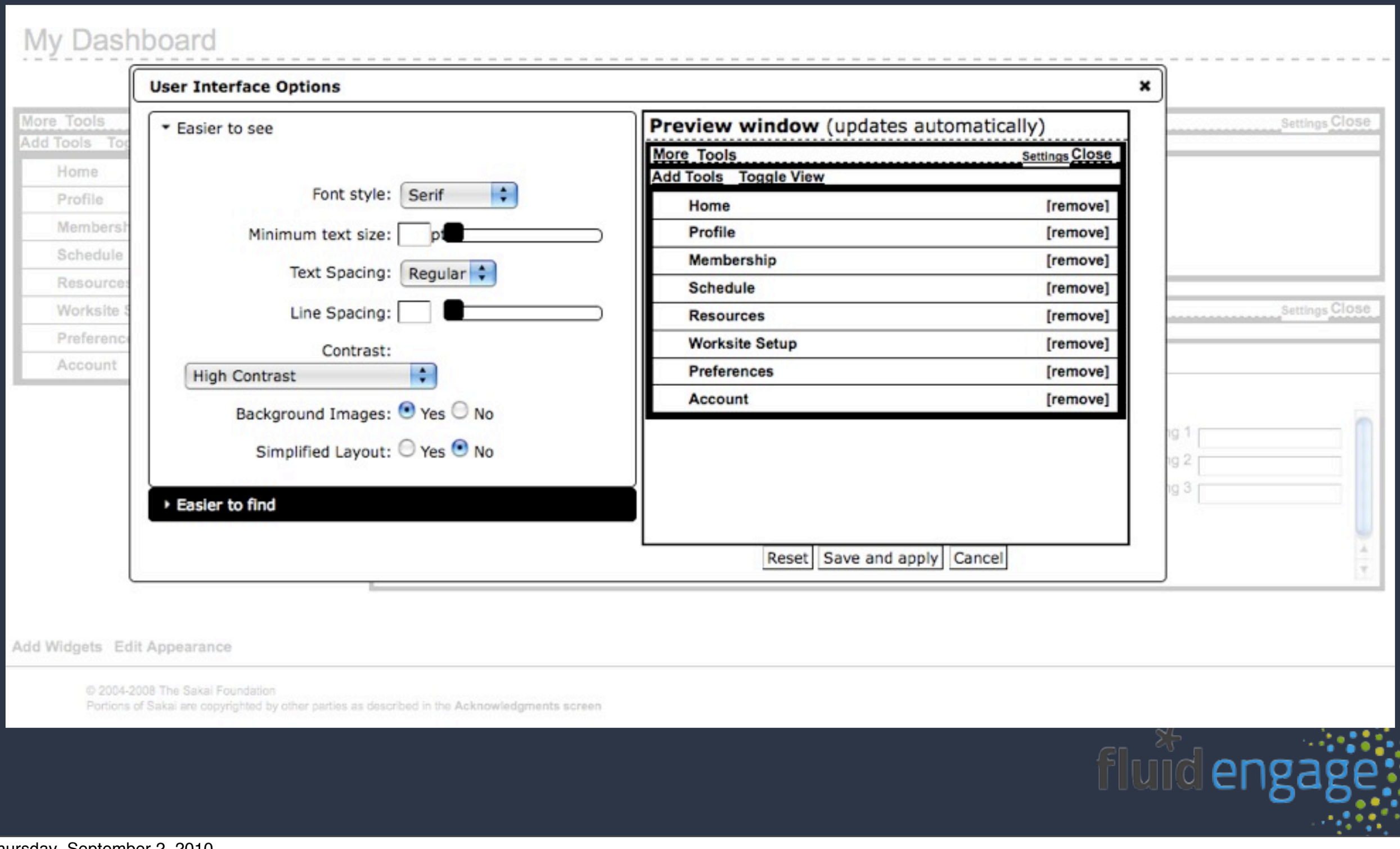
Demo UI Options?

UI Options



fluid engage

UI Options



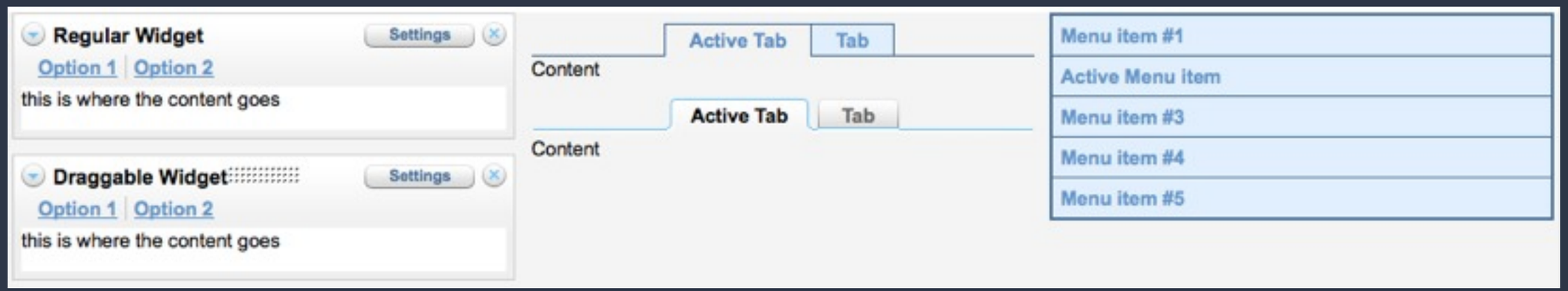
fluid engage

Code!

FSS Themes

Slate

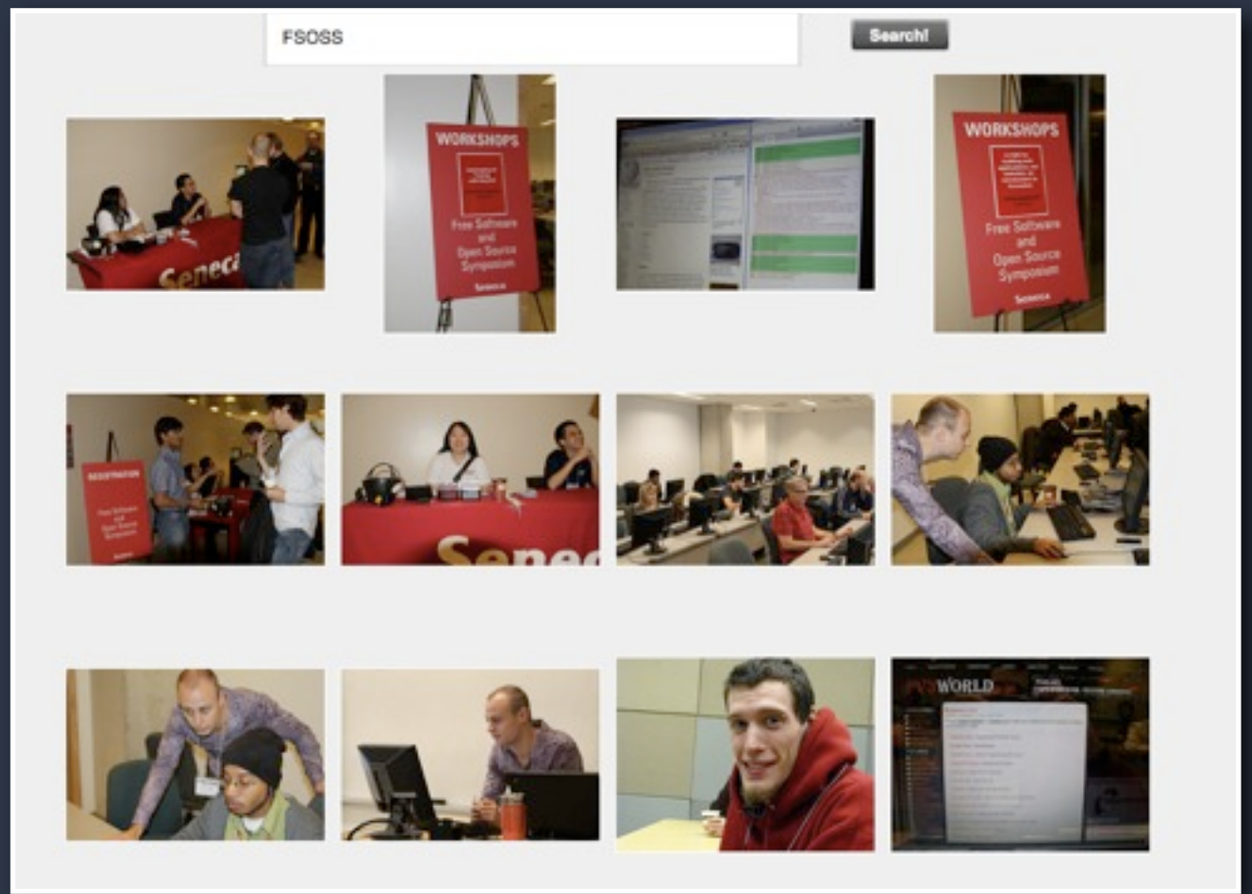
Mist



High Contrast

Mobile FSS Applied

.fl-theme-coal



=

.fl-theme-iphone

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=”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>
```

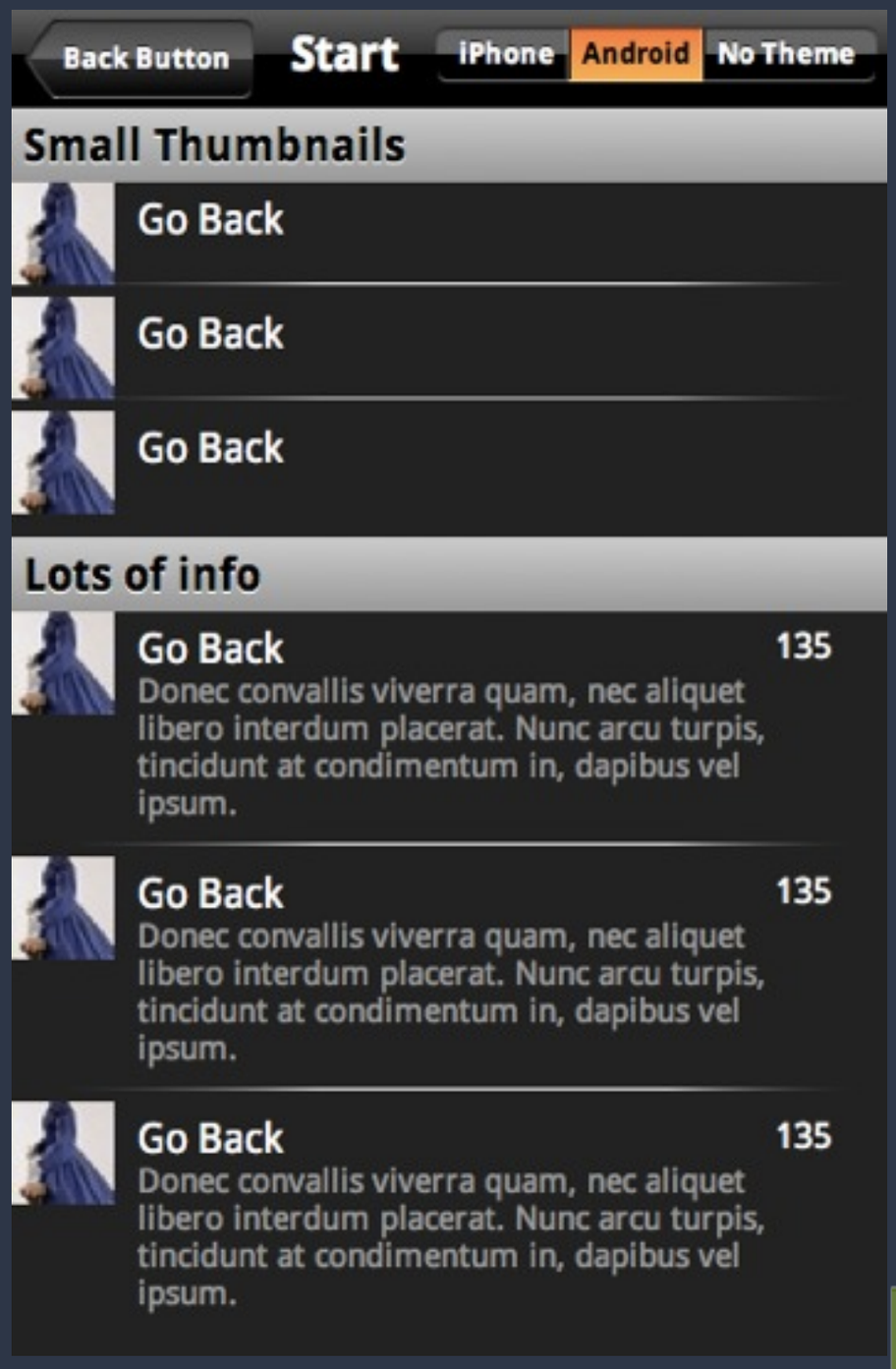
FSS: Desktop

```
<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



Fluid

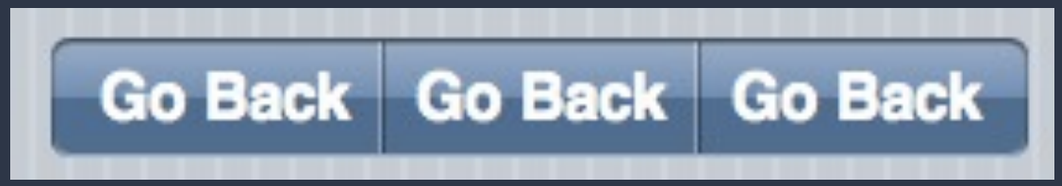
mFSS: iPhone Theme

```
<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

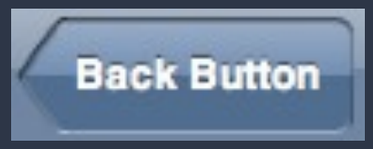
```
<head>
        <link type="text/css" rel="stylesheet"
              href="fss-mobile-layout.css" />        <link
type="text/css" rel="stylesheet"
                href="fss-mobile-theme-android.css" /></head>

<body class=”fl-theme-android”>

</body>
```

FSS: Tabs




```
<ul class="fl-tabs">  <li class="fl-tabs-active">
    <a href="#">Go Back</a>
  </li>  <li>
      <a href="#">Go Back</a>
  </li>  <li>
    <a href="#">Go Back</a>
  </li>
</ul>
```

mFSS: Back Button



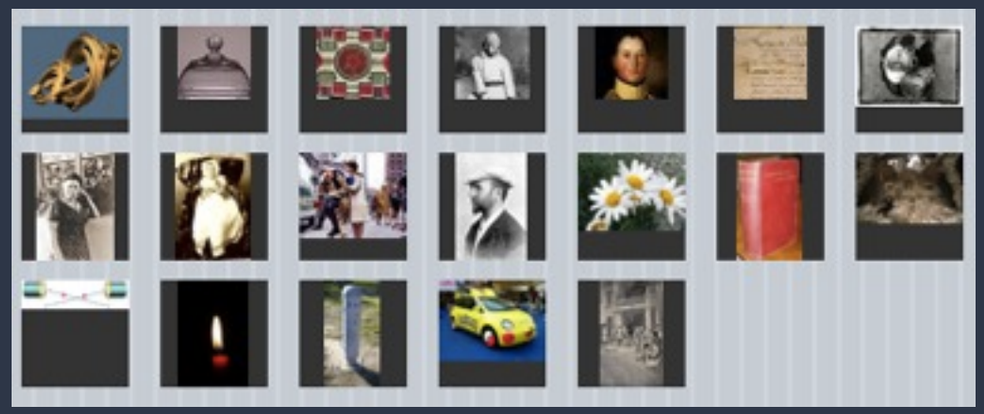

```
<a href="../mobile.html"
    class="fl-button fl-backButton">
      <span class="fl-button-inner">Back Button</span>
</a>
```

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>
```

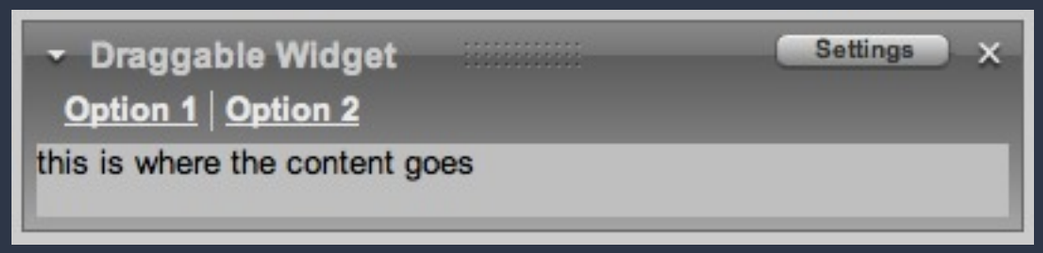
Fluid

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>       <h2>Regular Widget</h2>    </div>    <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

Architecture and Accessibility

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>
```

```
.fl-list-menu li {
    padding:0;
}
.fl-list-menu li a {
    display:block;
    padding: 12px 0px 12px
12px;
    text-decoration: none;
    font-weight: bold;
    outline: none;
}
```

Fluid

Declarative Programming

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

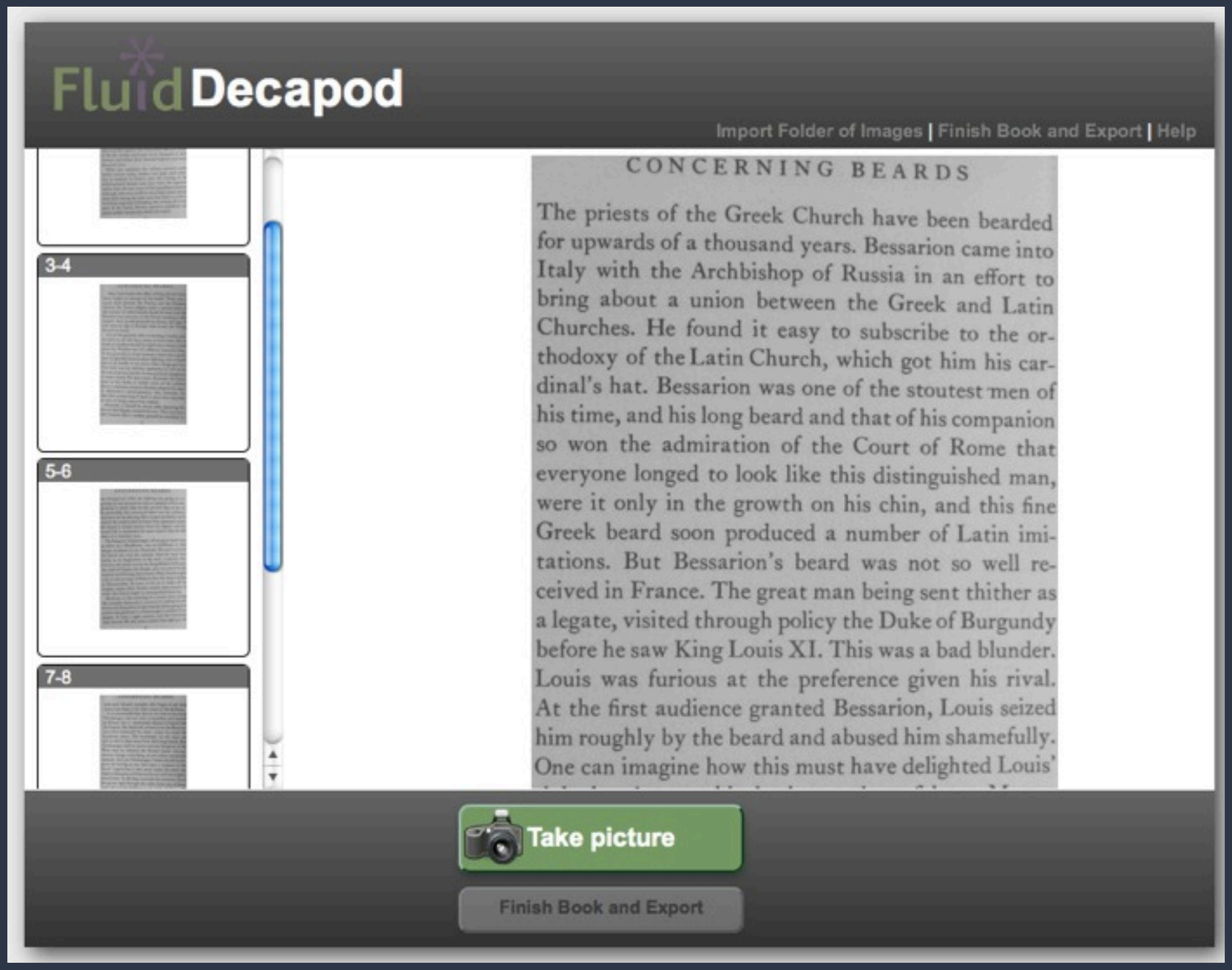
Declarative Programming

what

not

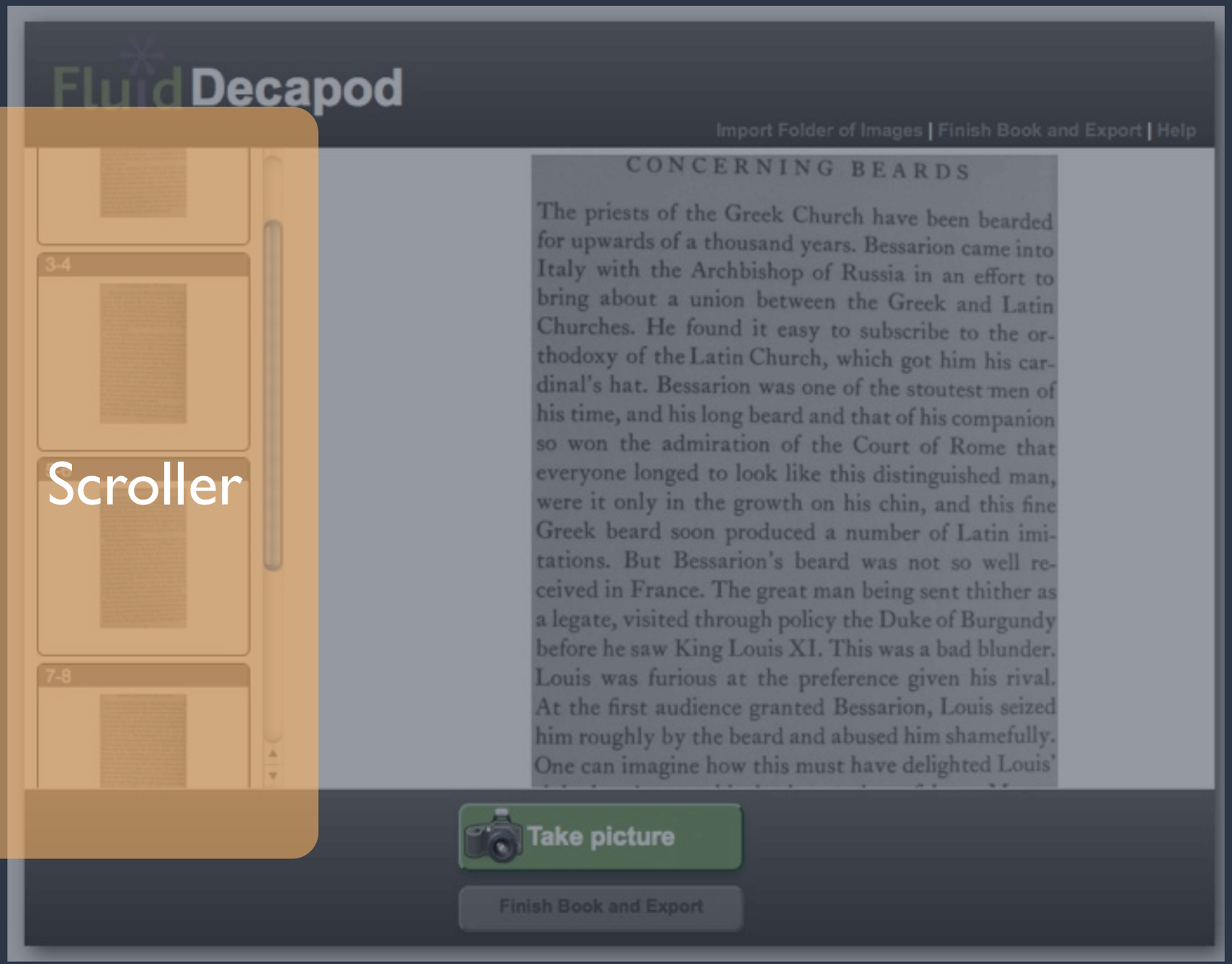
how

Compare & Contrast



Fluid

Compare & Contrast



Fluid

Imperative

```
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 - 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();
```

Fluid

Declarative

```
<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
```

Interoperable

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

- This means they are adaptable and extensible