

# Web Accessibility and Design

**Colin Clark**, Fluid Technical Lead & Architect

**Justin Obara**, Fluid QA Lead

**Jess Mitchell**, Fluid Project Manager



# Topics We'll Cover

- The Fluid community
- What is accessibility?
- Demo: assistive technologies
- Accessibility, Design, and WCAG 2.0
- Testing for accessibility
- Overview of building accessible websites

The logo for Fluid\* features the word "fluid" in a bold, lowercase, sans-serif font. A stylized asterisk symbol is positioned above the letter "i". The entire logo is centered within a white rectangular box, which is itself set against a larger, rounded blue square background.

An Open Source Community for Inclusive Design

Fluid is much more than a software project. Fluid is **culture change, best practices**, a way to approach **design, development**, and the power therein.

*Funded by a grant from*  
The  
Andrew W. Mellon  
Foundation

fluid\*engage

# Fluid...

<http://fluidproject.org>

- Is an open source community of
  - Designers
  - Developers
  - Accessibility experts
- Helps other open communities
- Consists of universities, museums and individuals



# What We Do

- Code (examples, tools, best-practices)
- Design (advice, tools)
- Expertise & Resources
- Teaching & Evangelism
  
- **Help**

# Tangibles



- core product: framework, code, components



- project with museums: visitor experience



- project with museums: collections management





# Fluid Engage

- Open source collaboration with museums
- Visitor engagement: learn and contribute
- Mobile apps and in-gallery kiosks
- Use phones visitors bring into the museum
- All built with open source Web technology



# What about you?

- Tell us who you are...
- Particular questions
- Particular concerns or goals for the day
- Structure for the day



# 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

# Motivations for Accessibility

- Legislative (ADA and Section 508)
  - Target
  - Spy Museum

# Target and ADA

(brick and mortar vs. the Web)

“In early 2006, The National Federation of the Blind (NFB) and others charged that blind people could not access Target.com, citing “public accommodation” provisions under the Americans with Disabilities Act (ADA).

Target fought the lawsuit, claiming that ADA provisions only related to their “brick-and-mortar” stores. The court found that Target.com was a “gateway” to stores and “heavily integrated” with physical stores, making the website subject to ADA requirements.”

<http://gneil.blogspot.com/2008/09/target-6-million-settlement-is-your.html>



# Spy museum and ADA...

- Title III of ADA applies to Museums
- no individual with a disability is discriminated against on the basis of a disability in the full and equal enjoyment of the entities' goods, services, and facilities
- a public accommodation must also provide appropriate auxiliary aids and services in order to ensure effective communication

<http://www.ada.gov/spymuseumfctsh.html>



# Motivations for Accessibility

- Legislative (ADA and Section 508)
- Business and outreach (reach everyone)
- Accessible is better (for everyone)

# It's just better

- “curb cut effect” -- everyone benefits
- accessible technology tends to be
  - more interoperable
  - easier to re-purpose
  - more future-proof
  - more robust
  - easier to use on a variety of devices



# Models for Web Accessibility

- Text-only site
- One site, accessible for all
- Adaptable and Personalizable



# 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



## User Interface Options

[View Demo](#) | [More Info](#)

Transform the presentation of the user interface and content resources so that they are personalized to an individual user's needs.

Let end-users transform the content they see according to their own preferences and needs.





# UI Options

## My Dashboard

### User Interface Options

▼ Easier to see

Font style:

Minimum text size:

Text Spacing:

Line Spacing:

Contrast:

Background Images:  Yes  No

Simplified Layout:  Yes  No

► Easier to find

### Preview window (updates automatically)

Tools

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

Reset Save and apply Cancel

Add Widgets

Edit Appearance



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

fluidengage



# UI Options

My Dashboard

### User Interface Options

▼ Easier to see

Font style:

Minimum text size:  px

Text Spacing:

Line Spacing:

Contrast:

Background Images:  Yes  No

Simplified Layout:  Yes  No

► Easier to find

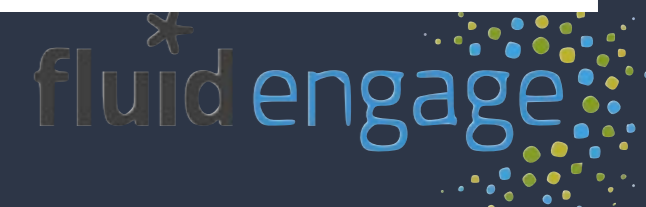
### Preview window (updates automatically)

More Tools		Settings	Close
Add Tools Toggle View			
Home		[remove]	
Profile		[remove]	
Membership		[remove]	
Schedule		[remove]	
Resources		[remove]	
Worksite Setup		[remove]	
Preferences		[remove]	
Account		[remove]	

Reset Save and apply Cancel

Add Widgets Edit Appearance

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



# Assistive Technology Demos



# W3C: Web Content Accessibility Guidelines (WCAG) 2.0

- Principle 1: Perceivable - Information and user interface components must be presentable to users in ways they can perceive.
- Principle 2: Operable - User interface components and navigation must be operable.
- Principle 3: Understandable - Information and the operation of user interface must be understandable.
- Principle 4: Robust - Content must be robust enough that it can be interpreted reliably by a wide variety of user agents, including assistive technologies.

# WCAG is Good Design





# Perceivable

- Text alternatives for:
  - Images
  - Time-based media
  - CAPTCHAs
- Adaptable presentation
- Use colour and contrast effectively
- Organize content in a meaningful sequence

*Searchable, readable, faster*



# Operable

- Content needs to work with the keyboard
- Provide enough time to read and use
- Help users to navigate, find content, and locate themselves in your site

*Easier to use and interact with*



# Understandable

- Use plain language
- Define jargon and abbreviations
- Consistent and predictable user interfaces
- Help users avoid mistakes

*Speaks to users on their terms; less frustrating UX*



# Robust

- Use valid markup and standards
- Describe the names, roles, and values of all user interface controls

*Sites last longer, and are easier to repurpose*



## WCAG 2.0 Quick Reference List

- **1.1 Text Alternatives:** Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language.
- **1.2 Time-based Media:** Provide alternatives for time-based media.
- **1.3 Adaptable:** Create content that can be presented in different ways (for example simpler layout ) without losing information or structure.
- **1.4 Distinguishable:** Make it easier for users to see and hear content including separating foreground from background.
- **2.1 Keyboard Accessible:** Make all functionality available from a keyboard.
- **2.2 Enough Time:** Provide users enough time to read and use content.
- **2.3 Seizures:** Do not design content in a way that is known to cause seizures.
- **2.4 Navigable:** Provide ways to help users navigate, find content and determine where they are.
- **3.1 Readable:** Make text content readable and understandable.
- **3.2 Predictable:** Make Web pages appear and operate in predictable ways.
- **3.3 Input Assistance:** Help users avoid and correct mistakes.
- **4.1 Compatible:** Maximize compatibility with current and future user agents, including assistive technologies.

<http://www.w3.org/WAI/WCAG20/quickref/>



# 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



# Designing for Everyone

- Accessibility is no longer optional
- Visual design is no longer optional
- Accessibility cannot be absolutely measured
- Accessibility is a continuum
- Accessibility and Design have the same  
GOAL

[http://webaim.org/blog/access\\_vs\\_design/](http://webaim.org/blog/access_vs_design/)



# Designing for Everyone

- Problem Statement / Goals
- Contextual Inquiry / Observations
- Personas / Scenarios
- Design Patterns / Patterns of Activity
- Test Test Test (there are ways to test all of this)



# How to start:

- I. Use “alt” text

# How to start:

## I. Use “alt” text

- caution: it might make you re-think what you use as images!

# How to start:

## 1. Use “alt” text

- caution: it might make you re-think what you use as images!

## 2. Label and describe your content

# How to start:

## 1. Use “alt” text

- caution: it might make you re-think what you use as images!

## 2. Label and describe your content

- makes it more robust and searchable

# How to start:

## 1. Use “alt” text

- caution: it might make you re-think what you use as images!

## 2. Label and describe your content

- more useful, compatible, and searchable

## 3. Add captions and transcripts to your media

# How to start:

## 1. Use “alt” text

- caution: it might make you re-think what you use as images!

## 2. Label and describe your content

- more useful, compatible, and searchable

## 3. Add captions and transcripts to your media

- Searchable, usable in diverse environments



# Taking a Look at Accessibility



# 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

# Screen Enlargement

When you make things bigger or resize...

1. Is all the text visible?
2. Does it overlap or break up in any way?
3. Are headers and labels still correctly associated?
4. Do columns shift or realign unexpectedly?

# 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?

# Keyboard Navigation

- Many different types of users use the keyboard
- You probably do, too!
- Keyboard access is 1-dimensional: forward & back
- *Everything that works with the mouse, works with the keyboard*

# Keyboard Navigation Conventions

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

# Keyboard Navigation Conventions

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

## I. The `Tab` key moves focus to the next control

# Keyboard Navigation Conventions

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

1. The `Tab` key moves focus to the next control
2. `Shift-Tab` moves focus backwards



# Keyboard Navigation Conventions

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

1. The **Tab** key moves focus to the next control
2. **Shift-Tab** moves focus backwards
3. **The Enter and Spacebar keys activate a control**

# Keyboard Navigation Checklist

1. Do all links, buttons, and form controls receive focus?
2. Can all controls be activated with Enter or Spacebar?
3. Are there any areas you get stuck or need the mouse?
4. Do calendar pickers and other rich widgets work?

# What About Shortcuts?

- Accessibility guidelines use to suggest “access keys”
- Keyboard shortcuts bound to the `alt` key
- A huge source of controversy
- Bottom line: *don't add new keyboard shortcuts*

# Alternatives to Media

1. Do all images have meaningful, non-redundant alt text?
2. Do videos have captions? (They're easy to add!)
3. Does audio have a transcript?

Hover your mouse over images in Internet Explorer to see alt text...



# 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/>

## Errors found while checking this document as HTML5!

<b>Result:</b>	40 Errors, 2 warning(s)		
<b>Address :</b>	<input type="text" value="http://www.google.com/"/>		
<b>Encoding :</b>	iso-8859-1	<input type="text" value="(detect automatically)"/>	
<b>Doctype :</b>	HTML5	<input type="text" value="(detect automatically)"/>	
<b>Root Element:</b>	html		

# Accessibility Checkers

aChecker

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



The screenshot shows the aChecker website interface. It features a white background with a grey border. At the top left, there is a yellow tab labeled "Input". Below this, there are two main sections. The first section is titled "Check Accessibility by URL" in red text, followed by a text input field and a "Check It" button. The second section is titled "Check Accessibility by File Upload" in red text, followed by a text input field, a "Browse..." button, and a "Check It" button. At the bottom left, there is a link labeled "Options" with a right-pointing arrow.

Wave

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



The screenshot shows the Wave website interface. It features a white background with a grey border. The top section is titled "Enter a web site address" in bold black text. Below this, there is a text input field and a "WAVE this page!" button. The bottom section is titled "Upload a file" in bold black text. Below this, there is a paragraph of text: "If you have files that are not publicly available on the internet, you can upload the files for WAVE evaluation. Simply browse to the file using the form below." This is followed by a text input field, a "Browse..." button, and a "WAVE this file!" button.

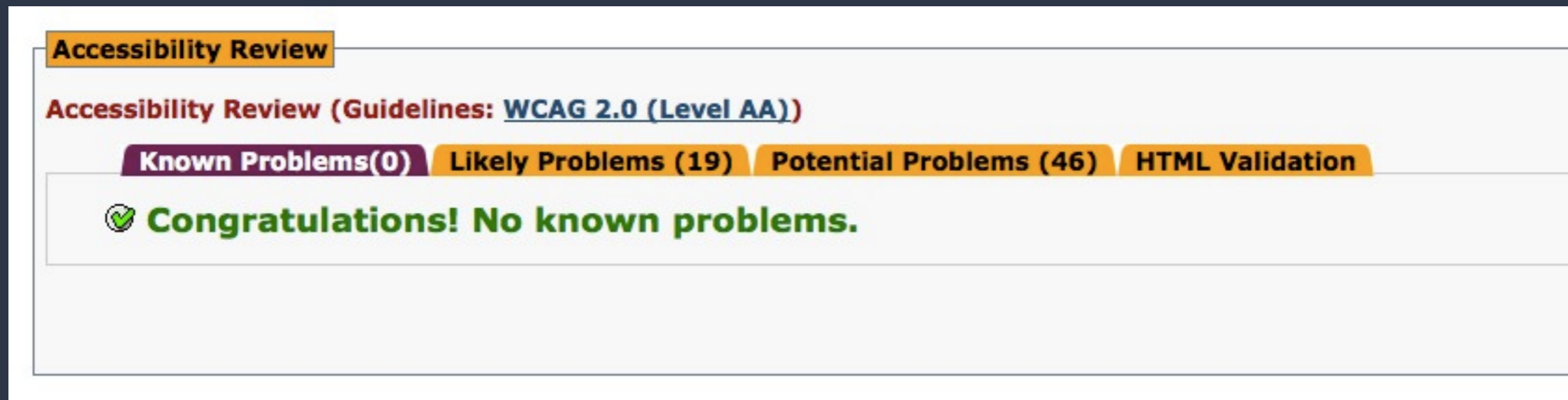




# Accessibility Checkers

aChecker

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




The screenshot shows the 'Accessibility Review' section of the aChecker tool. It features a header with the title 'Accessibility Review' and a sub-header 'Accessibility Review (Guidelines: WCAG 2.0 (Level AA))'. Below this, there are four tabs: 'Known Problems(0)', 'Likely Problems (19)', 'Potential Problems (46)', and 'HTML Validation'. The 'Known Problems(0)' tab is selected, displaying a green checkmark icon and the text 'Congratulations! No known problems.'

# 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:

 disable styles

---

**h1**

 Designing software that works - for everyone.

**title**

**h2** 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](#)

**h3** Fluid Integration Examples **h3** Infrastructure



# 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

✖ **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 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](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

# 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.

# 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.

# “alt” text in code

```

```



# Alternatives to Images



## Alt Text

```

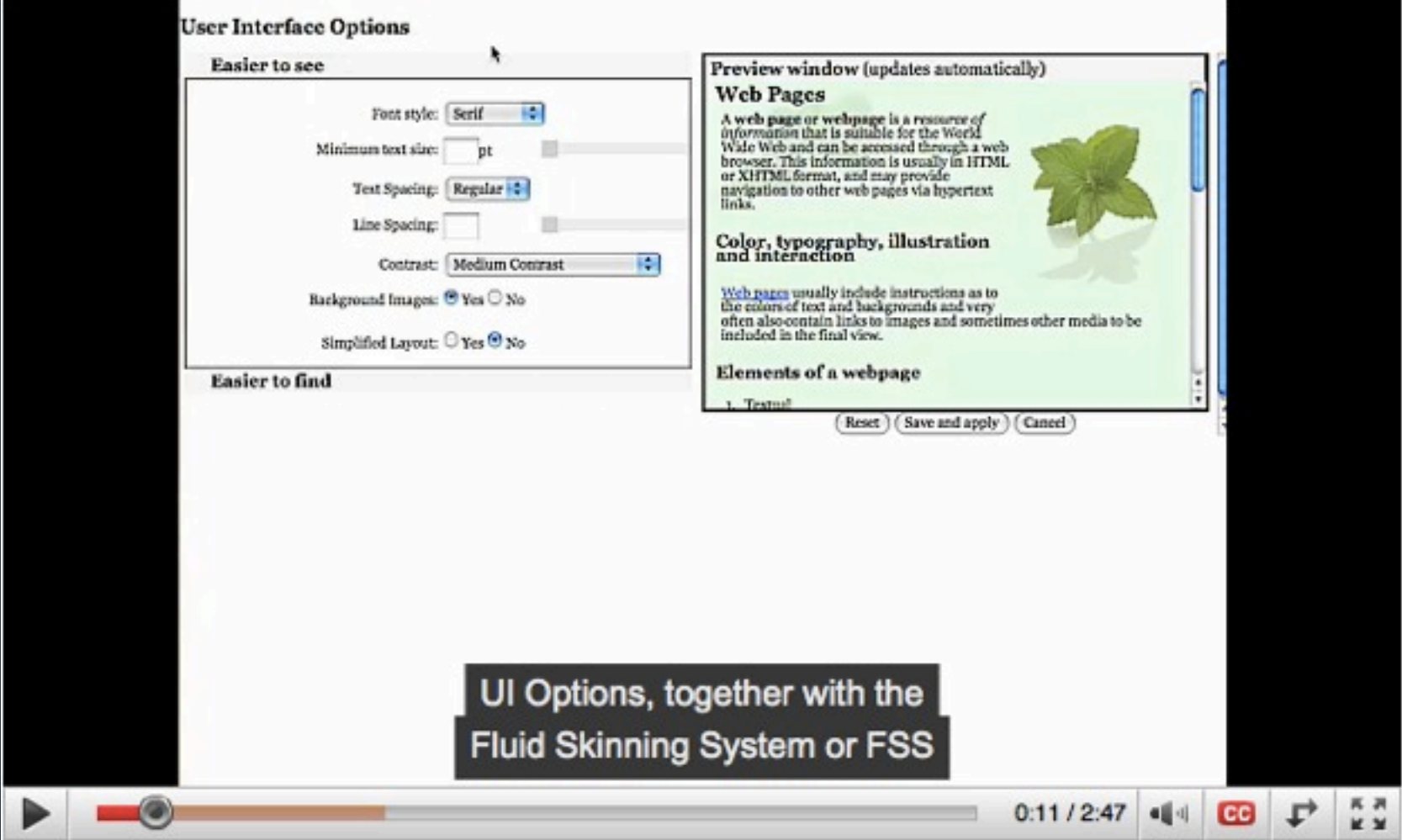
```

## Background Images + Real Text

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

# Alternatives to Video

## YouTube



The screenshot displays a video player interface. The main content area shows a 'User Interface Options' dialog box with two sections: 'Easier to see' and 'Easier to find'. The 'Easier to see' section includes controls for 'Font style' (set to 'Serif'), 'Minimum text size' (a slider), 'Text Spacing' (set to 'Regular'), 'Line Spacing' (a slider), 'Contrast' (set to 'Medium Contrast'), 'Background Images' (radio buttons for 'Yes' and 'No', with 'Yes' selected), and 'Simplified Layout' (radio buttons for 'Yes' and 'No', with 'No' selected). The 'Easier to find' section is currently empty. To the right of the dialog is a 'Preview window (updates automatically)' titled 'Web Pages'. It contains text defining a web page and a small image of a green leaf. Below the preview window are buttons for 'Reset', 'Save and apply', and 'Cancel'. A black text box at the bottom of the video frame reads 'UI Options, together with the Fluid Skinning System or FSS'. The video player controls at the bottom show a progress bar at 0:11 / 2:47, a volume icon, a Creative Commons license icon, and other standard playback controls.

# Captioning

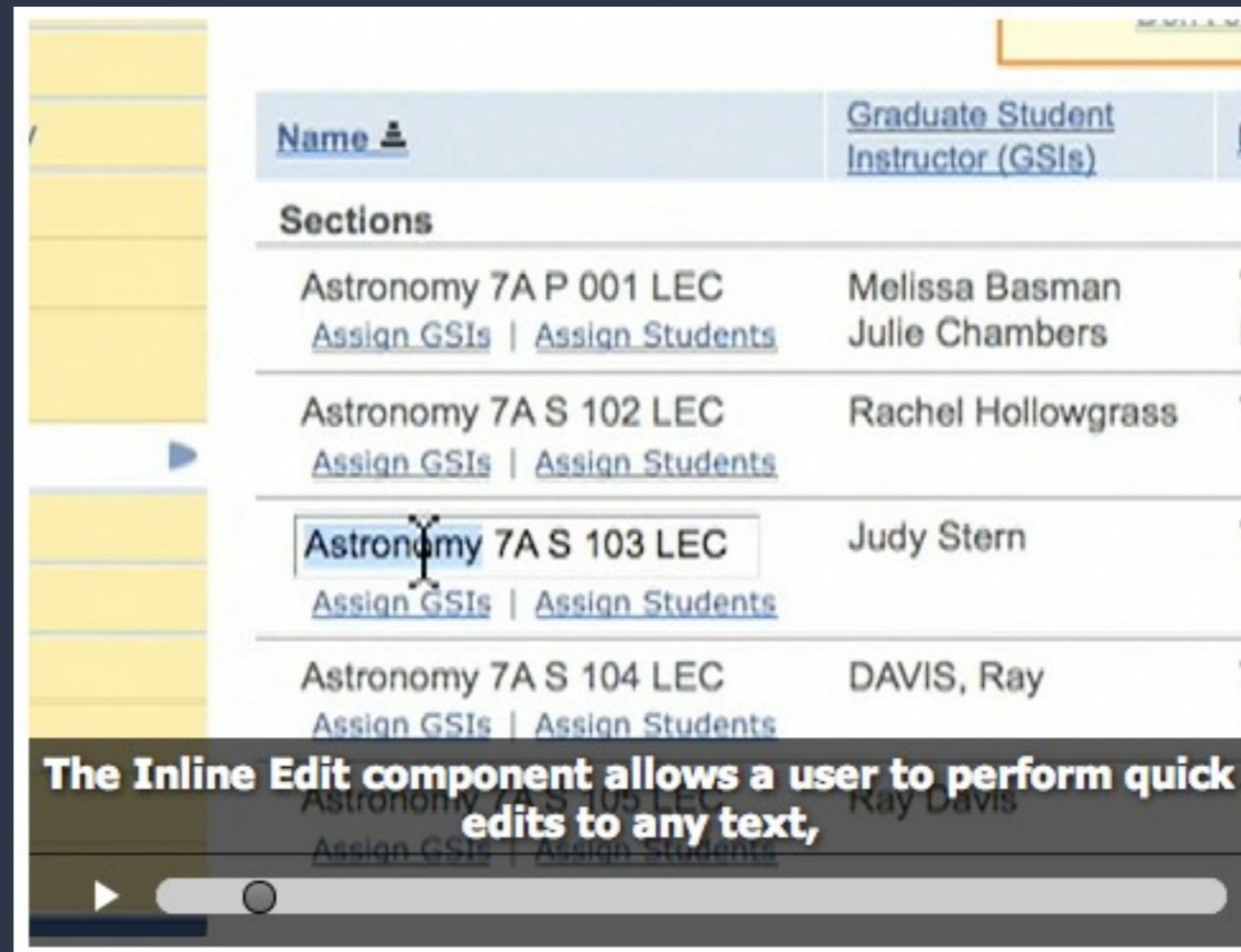
## YouTube and beyond

- Captions are great for everyone
- They make your videos search-friendly
- YouTube Automatic Captioning
  - Speech recognition: pretty awkward
  - Supply a transcript: quick and reliable
- Flash has a l l y problems, but HTML 5 is coming



# Alternatives to Video

## HTML 5



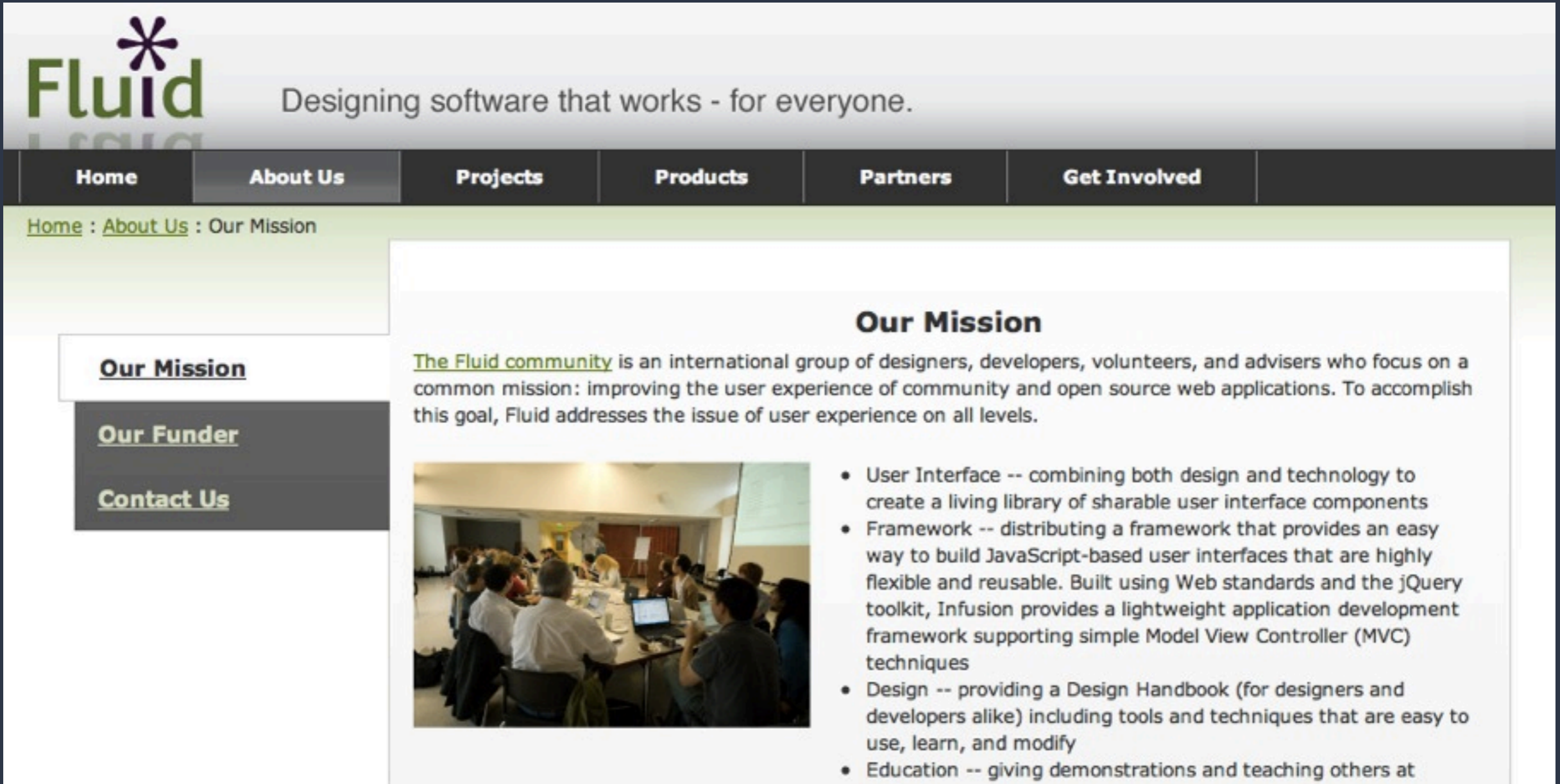
The screenshot shows a web application interface with a table of sections. The table has columns for 'Name', 'Graduate Student Instructor (GSIs)', and a numeric column. The 'Name' column contains entries like 'Astronomy 7A P 001 LEC', 'Astronomy 7A S 102 LEC', 'Astronomy 7A S 103 LEC', and 'Astronomy 7A S 104 LEC'. The 'Graduate Student Instructor (GSIs)' column contains names like 'Melissa Basman', 'Julie Chambers', 'Rachel Hollowgrass', 'Judy Stern', and 'DAVIS, Ray'. The 'Astronomy 7A S 103 LEC' row is highlighted, and the text 'Astronomy 7A S 103 LEC' is selected. A text input field is overlaid on this text, with a cursor at the end, demonstrating the inline edit component. Below the table, a video player interface is visible, showing a play button and a progress bar.

Name	Graduate Student Instructor (GSIs)	
Astronomy 7A P 001 LEC	Melissa Basman Julie Chambers	1
Astronomy 7A S 102 LEC	Rachel Hollowgrass	1
Astronomy 7A S 103 LEC	Judy Stern	1
Astronomy 7A S 104 LEC	DAVIS, Ray	1

**The Inline Edit component allows a user to perform quick edits to any text,**



# Avoiding Repetition



The screenshot shows the Fluid website's 'Our Mission' page. At the top left is the Fluid logo, a purple asterisk above the word 'Fluid' in green. To its right is the tagline 'Designing software that works - for everyone.' Below this is a dark navigation bar with links for Home, About Us, Projects, Products, Partners, and Get Involved. The 'About Us' link is highlighted. Below the navigation bar is a breadcrumb trail: 'Home : [About Us](#) : Our Mission'. On the left side, there is a vertical menu with three items: 'Our Mission' (highlighted), 'Our Funder', and 'Contact Us'. The main content area features the heading 'Our Mission' followed by a paragraph: '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.' Below the text is a photograph of a group of people in a meeting room, sitting around a table with laptops, looking at a presentation screen. To the right of the photo is a bulleted list of three items: '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', and 'Design -- providing a Design Handbook (for designers and developers alike) including tools and techniques that are easy to use, learn, and modify'. The fourth item, 'Education -- giving demonstrations and teaching others at', is partially cut off.

**Fluid** Designing software that works - for everyone.

Home About Us Projects Products Partners Get Involved

Home : [About Us](#) : Our Mission


**Our Mission**

**Our Funder**

**Contact Us**

## 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



Designing software that works - for everyone.

Home

About Us

Projects

**Navigation Bar**

Get Involved

[Our Mission](#)

[Our Funder](#)

[Contact Us](#)

## 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



Designing software that works - for everyone.

Home

About Us

Projects

**Navigation Bar**

Get Involved

Home > About Us > Our Mission

**Our Mission**

**Our Funder**

**Contact Us**

## 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.

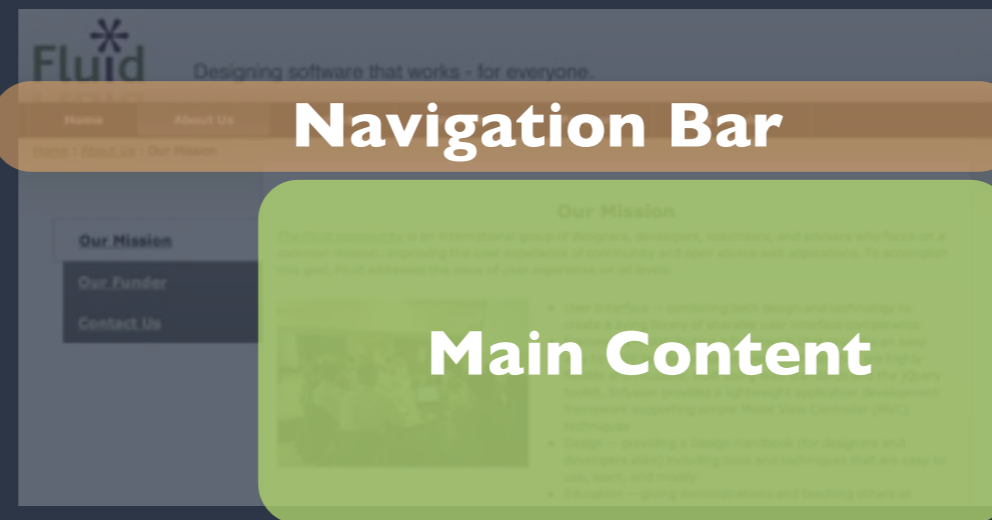


## Main Content

- User Interface -- combining both design and technology to create a living library of sharable user interface components
- 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

```
<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



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

- ["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

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

- "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

## Level Three

These are best viewed in **Safari 4**



# 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

## 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

## Level Three

These are best viewed in **Safari 4**



# Navigable Headings



Designing software that works - for everyone.

<H1>

Fluid Daily Build Resources

<H2>

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

<H3>

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

uPortal site

- 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

<H3>

- Continuum Build Server
- Unit Tests
- Manual Tests

<H3>

- "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

<H3>

These are best viewed in **Safari 4**



# Navigating Headings

The screenshot shows the Fluid website with a 'Heading List' dialog box overlaid. The website header includes the Fluid logo and the tagline 'Designing software that works - for everyone.' The main content area is titled 'Fluid Daily Build Resources' and contains several sections: 'Interactive demos are now available', 'Fluid Integration Examples', 'uPortal Instance', 'Infrastructure', 'Infusion', and 'Engage'. The 'Heading List' dialog box is a blue window with a close button in the top right corner. It contains a list of headings with their corresponding counts, a 'Sort Headings' section with radio buttons for 'In Tab Order' (selected) and 'Alphabetically', and a 'Display' section with radio buttons for 'All Headings' (selected), 'Level1 Headings', 'Level2 Headings', 'Level3 Headings', 'Level4 Headings', 'Level5 Headings', and 'Level6 Headings'. There are also 'Move To Heading' and 'Cancel' buttons on the right side of the dialog box.

**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 build process are reflected in the demo sites.

**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

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

- Content
- Unit
- Manu

#### Infusion

- "Infu
- Unit
- Unit
- Unit
- Unit
- Unit

#### Engage

These are

• Fluid

### Heading List

Fluid : Designing Software that works - For everyon...  
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

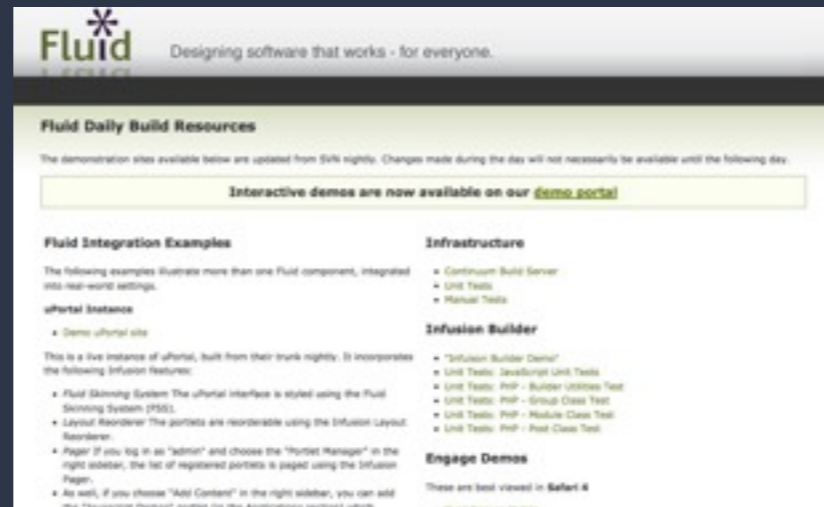
In Tab Order  
 Alphabetically

Display

All Headings  
 Level1 Headings  
 Level2 Headings  
 Level3 Headings  
 Level4 Headings  
 Level5 Headings  
 Level6 Headings

Move To Heading  
Cancel

# 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

**User Interface Options**

▼ Easier to see

Font style:

Minimum text size:

Text Spacing:

Line Spacing:

Contrast:

Background Images:  Yes  No

Simplified Layout:  Yes  No

► Easier to find

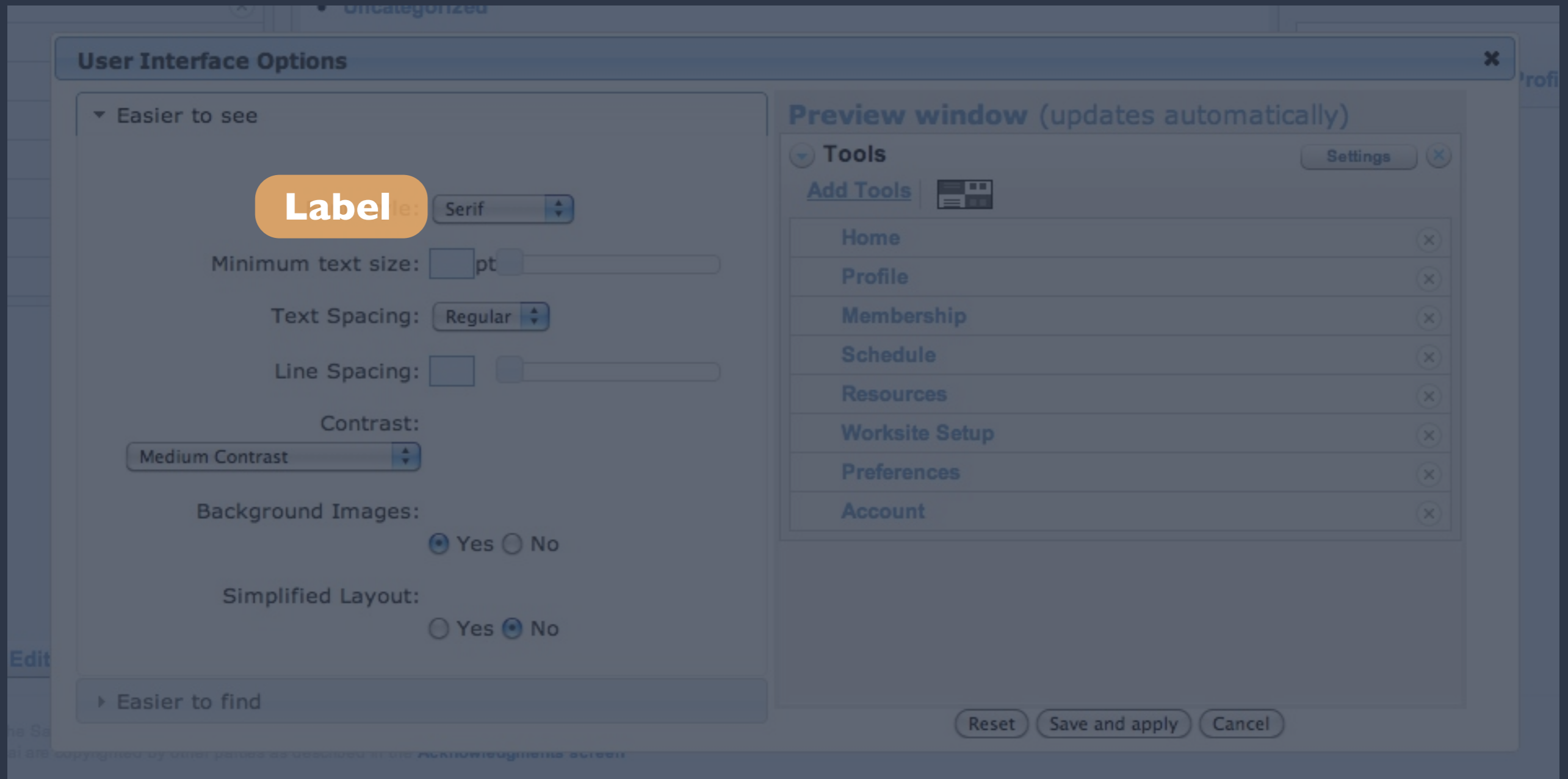
**Preview window (updates automatically)**

Tools

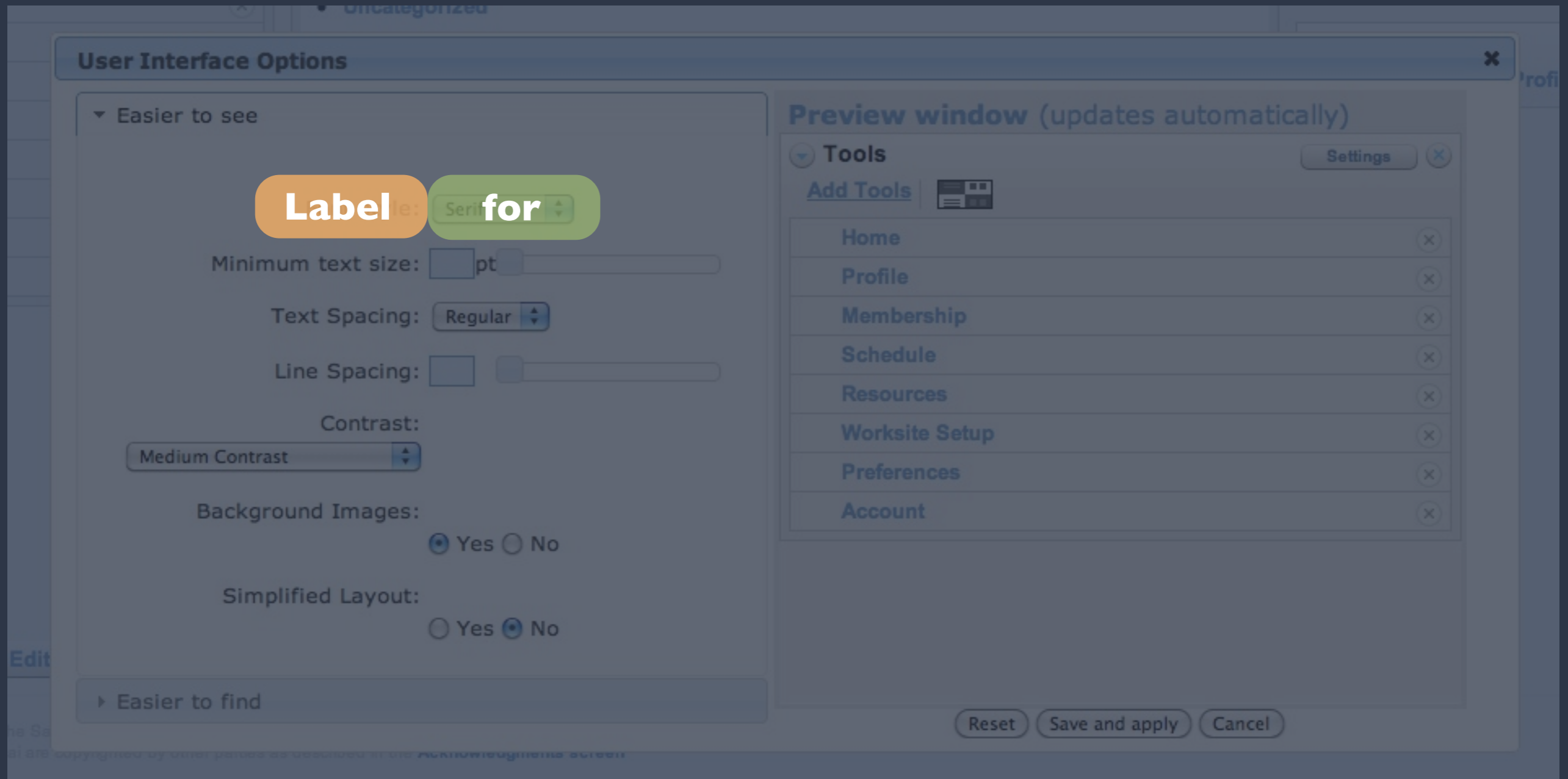
[Add Tools](#)

Home	<input type="button" value="x"/>
Profile	<input type="button" value="x"/>
Membership	<input type="button" value="x"/>
Schedule	<input type="button" value="x"/>
Resources	<input type="button" value="x"/>
Worksite Setup	<input type="button" value="x"/>
Preferences	<input type="button" value="x"/>
Account	<input type="button" value="x"/>

# Labelling Forms

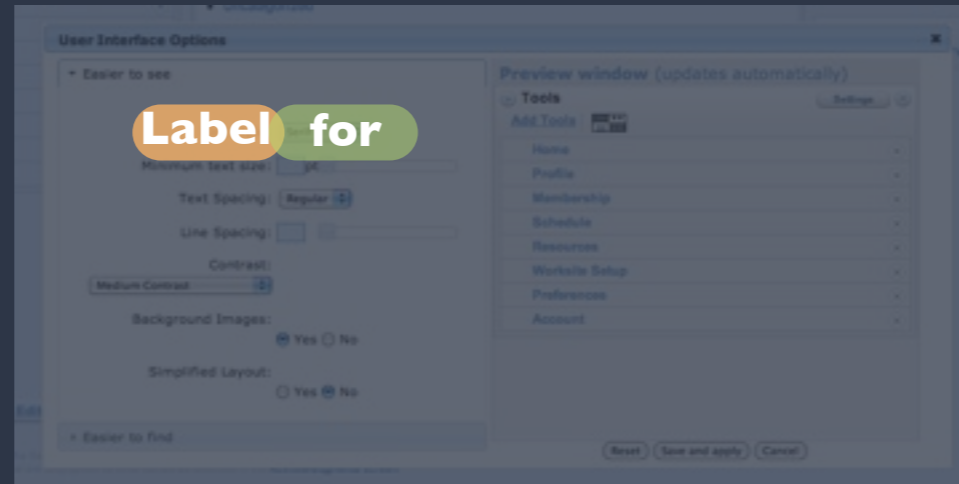


# Labelling Forms





# 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>
```

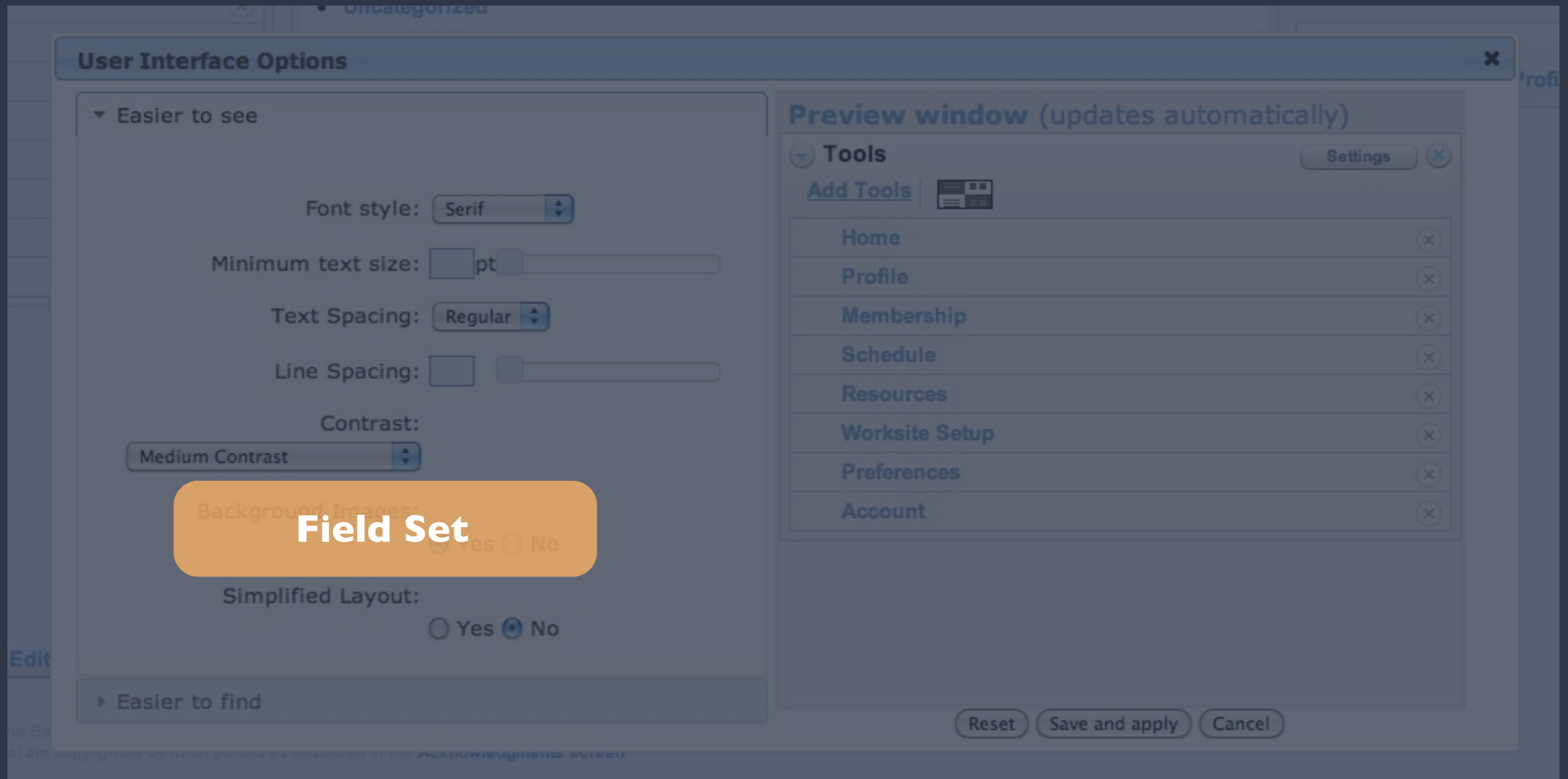
```
  <option value="courier">Courier</option>
```

```
  <option value="times">Times</option>
```

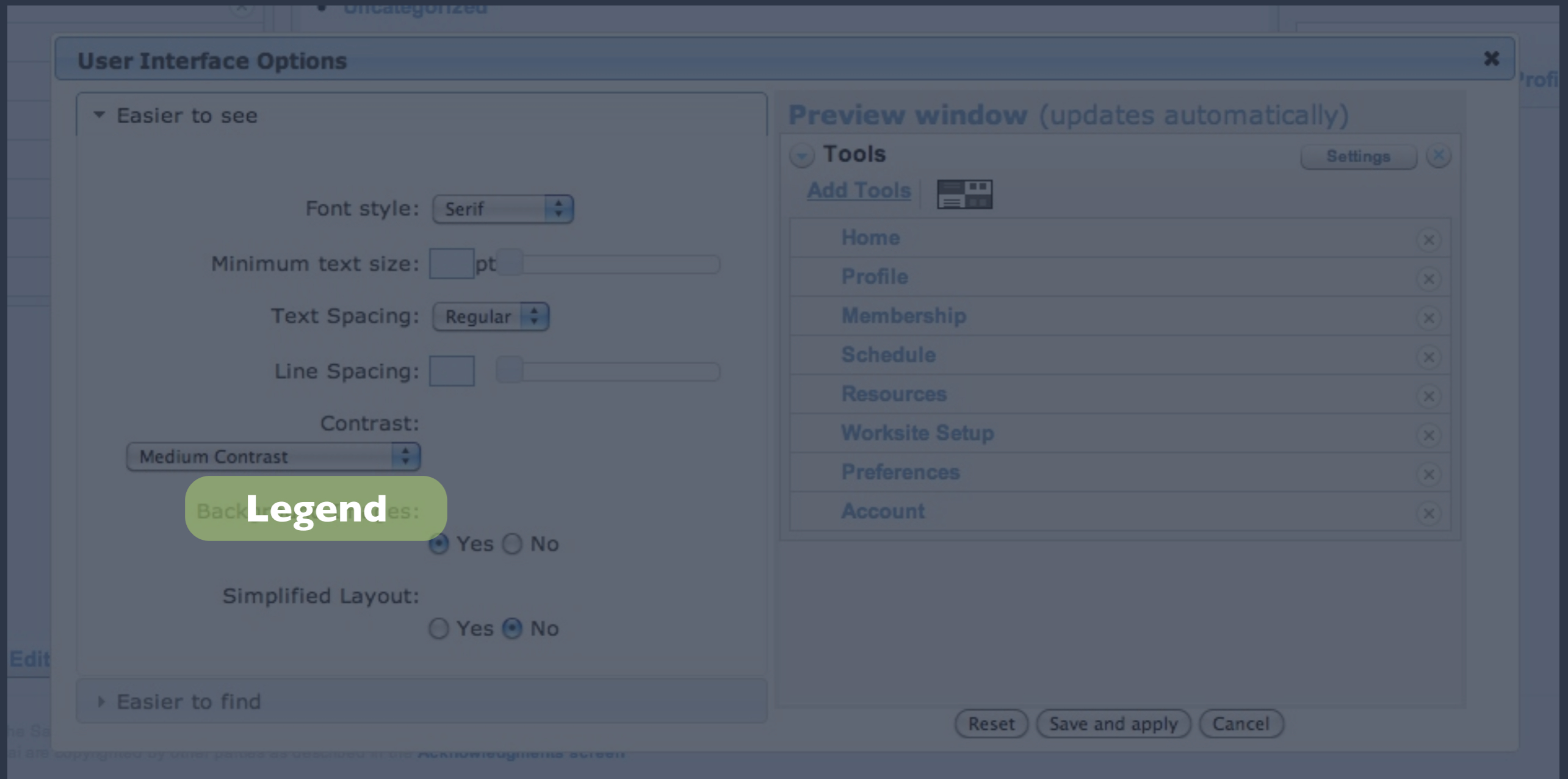
```
</select>
```

<li>

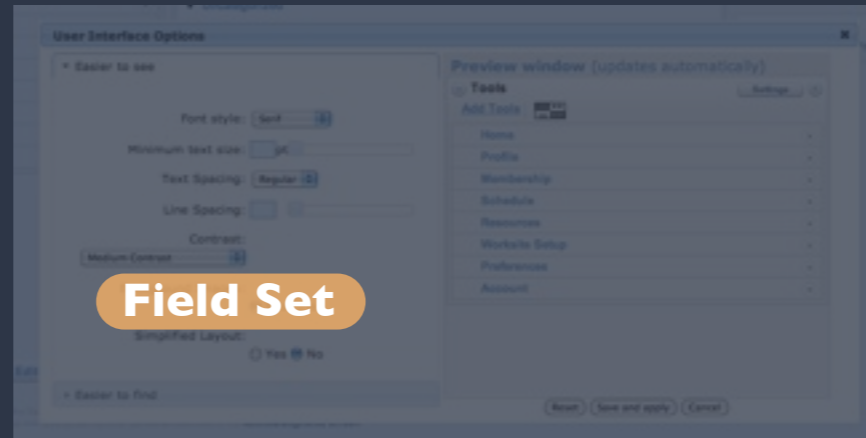
# Grouping Forms



# 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

1 2 3 4 5 6 7 (last) < previous [next >](#) Show  per page

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

1 2 3 4 5 6 7 (last) < previous [next >](#) 1-5 of 32 items

**Header**

**Body**

# Table Structure

The image shows a screenshot of a web application table. The table has a header row highlighted in orange and a body of rows highlighted in green. The header row contains the following text: 'Name', 'User ID', 'Points', and 'Comments'. The body rows contain the following data: 'Ahn, Jason', '15234314', '87', and 'Lorem ipsum dolor sit amet.'. The table is part of a larger interface with pagination controls at the top and bottom.

Name	User ID	Points	Comments
Ahn, Jason	15234314	87	Lorem ipsum dolor sit amet.
Alarman, Krista	19206726	79	Consectetur adipiscing elit.
Almeida, Elaine	19286942	92	Suspendisse ut arcu.
Azer, Tamer	19446940	88	
Bandak, Salim	19283721	89	Ut egestas, urna ut rhoncus gravida, risi nisi condimentum pede.

```
<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>
      <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>
```



# Web 2.0 Accessibility



# 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

# 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



# 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

# Supporting Assistive Technology



# Opaque Markup

```
// These are tabs. How would you know?  
<ol>  
  <li><a href="#cats">Cats</a></li>  
  <li><a href="#dogs">Dogs</a></li>  
  <li><a href="#gators">Gators</a></li>  
</ol>  
<div>  
  <div id="cats">Cats meow.</div>  
  <div id="dogs">Dogs bark.</div>  
  <div id="gators">Gators bite.</div>  
</div>
```



# Opaque Markup: Tabs

Cats

Dogs

Hamsters

Alligators

Cats [meow](#).

The cat (*Felis catus*), also known as the domestic cat or housecat to distinguish it from other felines and felids, is a small carnivorous mammal that is valued by humans for its companionship and its ability to hunt vermin and household pests. It has been associated with humans for at least 9,500 years and is currently the most popular pet in the world.

# 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 id="animalTabs" role="tablist" tabindex="0">  
  <!-- Individual Tabs shouldn't be focusable -->  
  <!-- We'll focus them with JavaScript instead -->  
  <li role="tab"><a href="#" tabindex="-1">Cats</a></li>  
  <li role="tab"><a href="#" tabindex="-1">Dogs</a></li>  
  <li role="tab"><a href="#" tabindex="-1">Gators</a></li>  
</ol>  
<div id="panels">  
  <div role="tabpanel" aria-labelledby="cats">Cats meow.</div>  
  <div role="tabpanel" aria-labelledby="dogs">Dogs bark.</div>  
  <div role="tabpanel" aria-labelledby="gators">Gators bite.</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);  
});
```



# 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](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

Cats

Dogs

Hamsters

Alligators

Cats [meow](#).

The cat (*Felis catus*), also known as the domestic cat or housecat to distinguish it from other felines and felids, is a small carnivorous mammal that is valued by humans for its companionship and its ability to hunt vermin and household pests. It has been associated with humans for at least 9,500 years and is currently the most popular pet in the world.

# 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/](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: that.options.selectors.tabs,
  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/Accessible_DHTML)

[http://developer.mozilla.org/en/docs/Key-navigable\\_custom\\_DHTML\\_widgets](http://developer.mozilla.org/en/docs/Key-navigable_custom_DHTML_widgets)

[http://developer.mozilla.org/en/docs/AJAX:WAI\\_ARIA\\_Live\\_Regions](http://developer.mozilla.org/en/docs/AJAX:WAI_ARIA_Live_Regions)



# Toolkit Accessibility





# Dojo

- The first to tackle DHTML accessibility
- Accessibility features include:
  - Support for Windows High Contrast
  - Comprehensive ARIA library
  - Keyboard navigation
- Most core widgets are covered

# Dojo High Contrast



A standard Dojo dialog box titled "Sample Dialog" with a close button (X) in the top right corner. The dialog has a white background and contains three text input fields labeled "Name:", "Location:", and "Description:". Below the fields is an "OK" button.



A high-contrast version of the "Sample Dialog" with a black background and yellow text and borders. The dialog has a close button (X) in the top right corner. It contains three text input fields labeled "Name:", "Location:", and "Description:". Below the fields is an "OK" button.

# jQuery

- A truly community-driven effort
- A step-by-step approach
- Features include:
  - Several accessible widgets, more to come
  - Comprehensive ARIA library
  - Keyboard navigation in progress

# jQuery

## Dialog

The screenshot displays a web page with a jQuery Dialog plugin. A dialog window titled "Basic dialog" is open, containing the text: "This is the default dialog which is useful for displaying information. The dialog window can be moved, resized and closed with the 'x' icon." The dialog has a yellow title bar with a close button (X) and a resize handle (three lines) in the bottom right corner. In the background, a form is visible with the following elements:

- Text: "Sed vel diam id l faucibus ante. Et amet auctor elit"
- Text input field containing "text input"
- Checkbox labeled "checkbox"
- Radio button labeled "radio"
- Select dropdown menu labeled "select"
- Textarea field labeled "textarea"

On the right side of the page, there is a "New window" button and some text: "s rhoncus cies neque, sit".



# infusion

<http://fluidproject.org/products/infusion/>



# Meet Infusion

- Application framework built on top of jQuery
- The culmination of our work helping others
- Designed for usability and accessibility
- Open architecture: everything is configurable



# What's in Infusion?

- A development framework for building apps
- UI components you can reuse and adapt
- Lightweight CSS framework for styling
- Accessibility tools and plugins for jQuery





# a11y: Infusion Goes Deeper

- jQuery Keyboard Navigation Plugin
- ARIA everywhere
- Everything is highly adaptable and flexible
- UI Options and the Fluid Skinning System:
  - Users can customize their environment



# Building Great UIs Is Hard

- Your code gets unruly as it grows
- UIs are hard to reuse or repurpose
- Design change requires big code change
- Accessibility is confusing
- Combining different code/libraries doesn't always work



# Flexible User Interfaces

Infusion is an application framework designed to provide unprecedented flexibility while preserving interoperability.



# Types of JavaScript Tools

- Foundational Toolkits
- Widget libraries
- Application Frameworks

... compare and contrast

# Foundational toolkits

- Totally presentation focused
- DOM manipulation
- Event binding
- Ajax

*jQuery*  
*Prototype*  
*Dojo core*



# Widget Libraries

- Reusable user interface widgets
  - Drag & Drop
  - Tabs
  - Sliders
  - Accordions

*jQuery UI*  
*Ext*  
*Scriptaculous*

  
Fluid

# Application frameworks

- **Model notifications** “something changed here”
- **Views** to help keep your presentational code clean
- **Data binding** to sync the display with your model

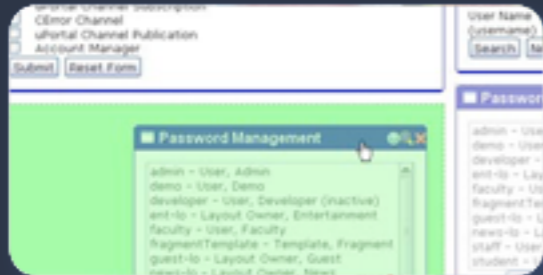
*SproutCore*

*Dojo/Dijit/  
Dojox*

*Cappuccino*

  
Fluid

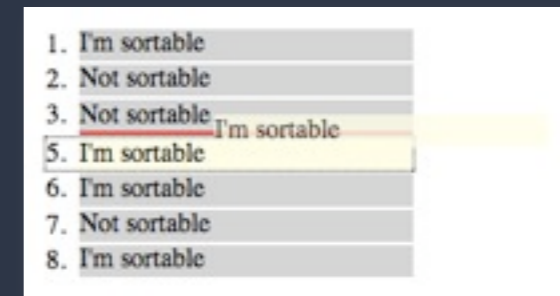
# The Reorderer Family



layouts



grids



lists

# Uploader

Upload Files ✕

File Name	Size	
LICENSE.txt	12.1 KB	✓
maven.xml	1.8 KB	✓
pom.xml	0.9 KB	✓
project.properties	0.1 KB	✓
project.xml	1.2 KB	
README.txt	5.4 KB	

Uploading: 5 of 6 files (15.9 KB of 21.3 KB) [Add more](#)



# Pager

The screenshot shows a Firefox browser window displaying the 'Site Settings' page for 'Members' in Sakai. The browser's address bar shows the URL: [http://build.fluidproject.org/fluid/sample-code/pager/sakai-site-setting/site\\_setting\\_members.html](http://build.fluidproject.org/fluid/sample-code/pager/sakai-site-setting/site_setting_members.html). The page title is 'Site Settings' and the sub-page is 'Members'. There are navigation tabs for 'General', 'Members', 'Roles & Permissions', and 'Site Backup & More'. The 'Members' tab is active, and there is an 'Add Members' button. Below the tabs, there are links for 'Select All', 'Select All Visible', 'Unselect All', and '0 Selected'. On the right, it shows '1-10 of 187 items' and a 'View 10 Per Page' dropdown. A pagination bar shows 'Prev 1 2 3 ... 17 18 19 (last) Next'. The main content is a table with columns: 'Members', 'Email', 'Role', and 'Active'. The 'Active' column contains dropdown menus. A tooltip is visible over the first dropdown, showing the names 'Albert Zimmerman' and 'Rob van der Voo'. The table lists the following members:

Members	Email	Role	Active
<input type="checkbox"/> Joe I. Instructor	<a href="mailto:joe@sakai.university.edu">joe@sakai.university.edu</a>	Instructor	Active
<input type="checkbox"/> Mike A. Smith	<a href="mailto:msmith@yahoo.com">msmith@yahoo.com</a>	TA	Active
<input type="checkbox"/> Jane Doe	<a href="mailto:j.doe@sakai.university.edu">j.doe@sakai.university.edu</a>	Student	Active
<input type="checkbox"/> Laura S. Roberts	<a href="mailto:roberts_laura@gmail.com">roberts_laura@gmail.com</a>	Guest	Active
<input type="checkbox"/> Erin A. Wilson	<a href="mailto:thewilsones@hotmail.com">thewilsones@hotmail.com</a>	Student	Active
<input type="checkbox"/> Ben Singer	<a href="mailto:singer21@hotmail.com">singer21@hotmail.com</a>	Student	Active
<input type="checkbox"/> Bhaktavatsalam Bhayakridbhayanashanachar	<a href="mailto:bb@local.host">bb@local.host</a>	Student	Active
<input type="checkbox"/> Dept Admin	<a href="mailto:da1@local.host">da1@local.host</a>	Instructor	Active
<input type="checkbox"/> Angeolo Haslip	<a href="mailto:ah@local.host">ah@local.host</a>	Student	Active



# UI Options & FSS

The screenshot displays the 'My Dashboard' interface with the 'User Interface Options' dialog box open. The dialog is divided into two main sections: 'Easier to see' and 'Easier to find'. The 'Easier to see' section includes settings for font style (Serif), minimum text size (with a slider), text spacing (Regular), line spacing (with a slider), contrast (Medium Contrast), background images (Yes/No), and simplified layout (Yes/No). The 'Easier to find' section is currently collapsed. A 'Preview window' on the right side of the dialog shows a live preview of the dashboard with the selected settings applied. At the bottom of the dialog are 'Reset', 'Save and apply', and 'Cancel' buttons. The background dashboard shows a 'Tools' sidebar with options like Home, Profile, Membership, Schedule, Resources, Worksite Setup, Preferences, and Account. At the bottom of the dashboard are 'Add Widgets' and 'Edit Appearance' buttons. The Sakai logo and copyright information are visible at the bottom left.

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  No

Simplified Layout:  Yes  No

► Easier to find

**Preview window (updates automatically)**

**Tools**

Add Tools

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

Reset Save and apply Cancel

Add Widgets Edit Appearance

**Sakai** © 2004-2008 The Sakai Foundation  
Portions of Sakai are copyrighted by other parties as described in the [Acknowledgments screen](#)

**Fluid**

# UI Options: High Contrast Theme

My Dashboard

**User Interface Options** [Close]

▼ Easier to see

Font style:

Minimum text size:  px

Text Spacing:

Line Spacing:

Contrast:

Background Images:  Yes  No

Simplified Layout:  Yes  No

► Easier to find

**Preview window (updates automatically)**

More Tools [Settings] [Close]

Add Tools Toggle View

Home	[remove]
Profile	[remove]
Membership	[remove]
Schedule	[remove]
Resources	[remove]
Worksite Setup	[remove]
Preferences	[remove]
Account	[remove]

Reset Save and apply Cancel

Add Widgets Edit Appearance

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



# Infusion is Different

- Accessibility baked right in
- Carefully designed interactions
- Markup is in your control
- Not the same old MVC
- Supports portals, mashups and CMS's



# 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>



# Building with FSS



Code!



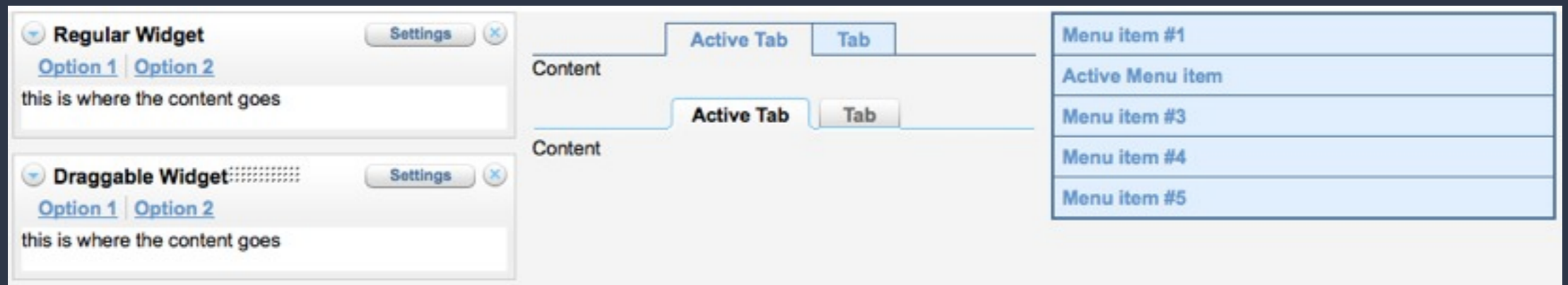


# FSS Themes

## Slate



## Mist



## High Contrast



# 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>
```

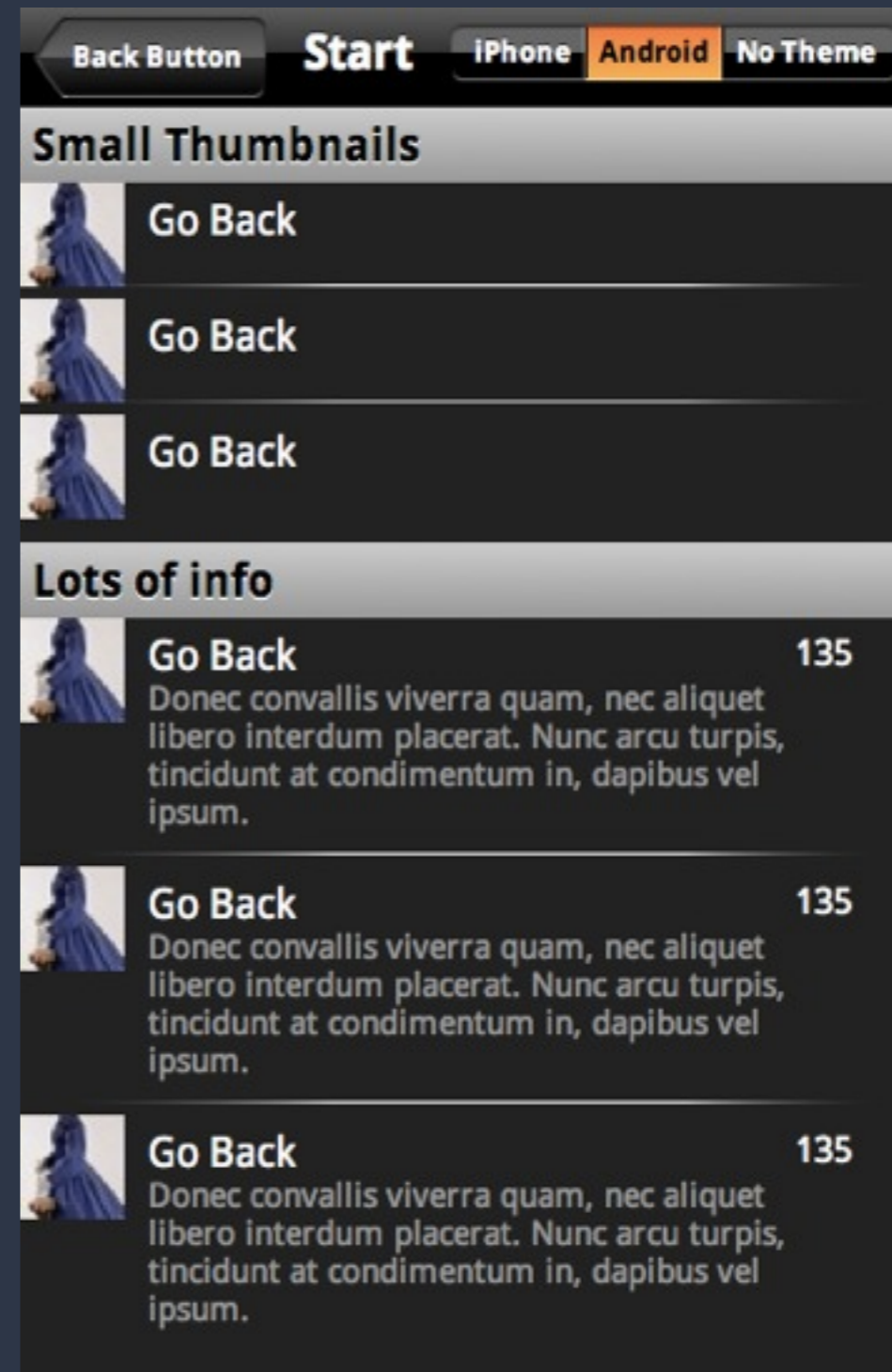


# mFSS Themes

## iPhone



## Android



# 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>
```

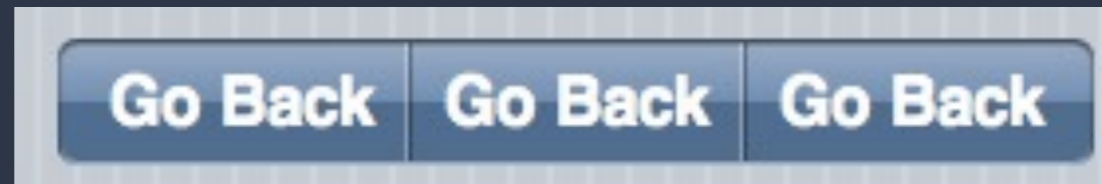


# mFSS: Android Theme

```
<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>
```

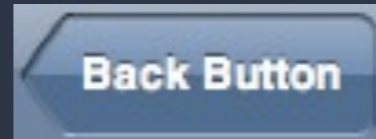


# FSS: Tabs



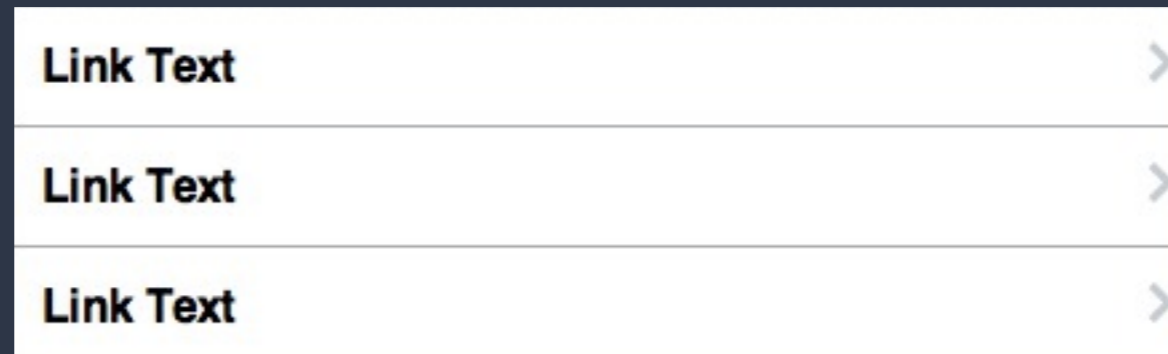
```
<ul class="fl-tabs">
  <li class="fl-tabs-active">
    <a href="#">Go Back</a>
  </li>
  <li>
    <a href="#">Go Back</a>
  </li>
  <li>
    <a href="#">Go Back</a>
  </li>
</ul>
```

# mFSS: Back Button



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

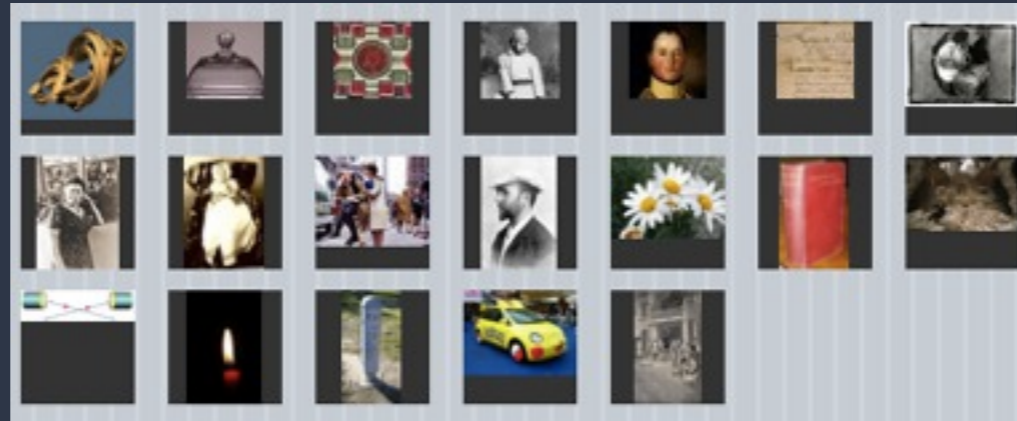
# FSS: Lists



```
<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>
```

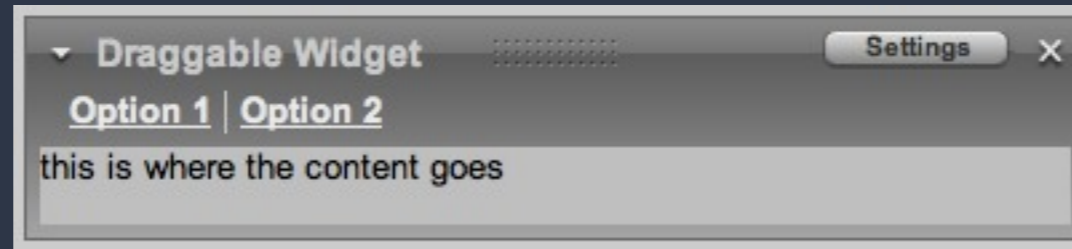


# mFSS: Image Grids



```
<ul class="fl-list-menu fl-list-thumbnails fl-grid">
  <li>
    <a href="20.png">
      
    </a>
  </li>
  <li>
    <a href="2.png">
      
    </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>
```

# Links to get started

- <http://www.alistapart.com>
- <http://webaim.org>
- <http://wiki.fluidproject.org/display/fluid/Design+Handbook>

Questions?

