Web Accessibility and Design

Colin Clark, Fluid Technical Lead & Architect
Justin Obara, Fluid QA Lead
Jess Mitchell, Fluid Project Manager

Tuesday, March 16, 2010
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
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.
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

Tuesday, March 16, 2010
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/spymuseumfctsht.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
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: Serif
- Minimum text size: 0
- Text Spacing: Regular
- Line Spacing: 0
- Contrast: Medium Contrast
- Background Images: Yes
- Simplified Layout: Yes

Easier to find

Reset, Save and apply, Cancel

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

Preview window (updates automatically)

Tuesday, March 16, 2010
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

Tuesday, March 16, 2010
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/
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:

1. Use “alt” text
How to start:

1. 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...
1. The Tab key moves focus to the next control
Keyboard Navigation Conventions

1. The Tab key moves focus to the next control

2. Shift-Tab moves focus backwards
Keyboard Navigation Conventions

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://validator.w3.org/)
  - [http://jigsaw.w3.org/css-validator/](http://jigsaw.w3.org/css-validator/)

---

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

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

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

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

aChecker

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

Wave

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

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

- No Magic
  - Is the alt text meaningful 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

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
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>
Alternatives to Video

YouTube

UI Options, together with the Fluid Skinning System or FSS
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 accessibility problems, but HTML 5 is coming
Alternatives to Video

HTML 5

The Inline Edit component allows a user to perform quick edits to any text.
Avoiding Repetition

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

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

Our Mission

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

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

Our Mission

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

- User Interface -- combining both design and technology to create a living library of sharable user interface components; a framework that provides an easy way to build 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...
Avoid Repetition

Skip Links

<!-- Navigation bar goes here -->

<a id="nav" title="navigation menu"></a>

<!-- Main page content goes here -->

<a title="content area" name="content"></a>
Designing Navigation

- Keep in mind that keyboard navigation is:
  - not just for screen reader users
  - is linear and 1-dimensional
  - can be slow and tedious
- Skip links should be available and visible to all
- Place them as high in the page as possible
Navigable Headings

Fluid Daily Build Resources

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

Interactive demos are now available on our demo portal

Fluid Integration Examples

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

uPortal Instance
- Demo uPortal site

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

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

Infrastructure
- Continuum Build Server
- Unit Tests
- Manual Tests

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

Engage Demos
These are best viewed in Safari 4
- Fluid Engage Mobile
Navigable Headings

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

Interactive demos are now available on our demo portal

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

uPortal Instance
- Demo uPortal site

This is a live instance of uPortal, built from 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

Tuesday, March 16, 2010
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 portals are reorderable using the Infusion Layout Reorderer.
- **Pager** If you log in as "admin" and choose the "Portlet Manager" in the right sidebar, the list of registered portlets is paged using the Infusion Pager.
- As well, if you choose "Add Content" in the right sidebar, you can add the "Javascript Demos" portlet (in the Applications section) which

**Level Three**

- Continuum Build Server
- Unit Tests
- Manual Tests

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

These are best viewed in Safari 4

- Fluid Focus Mobile
Navigable Headings

Level One

Level Two

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

Interactive demos are now available on our demo portal

Level Three

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

Level Four

- Demo uPortal site

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

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

Level Three

- Continuity Build Server
- Unit Tests
- Manual Tests

Level Three

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

These are best viewed in Safari 4

Level Three

- Fluid Fisheye Mobile
Navigable Headings

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 Demonstration Examples

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

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

These are best viewed in Safari 4

- Fluid Forms Mobile
Navigating Headings

Fluid Daily Build Resources

The demonstration sites available below are updated from SVN nightly. Changes made during the day are reflected here.

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

Infraserv

- Contiki
- Unit 1
- Unit 2
- Unit 3

Infusion

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

Engage

These are the Engage demo sites...

- Fluid Skinning System
- Layout Reorderer
- Pager
- "Add Content"

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

[Heading List]

Sort Headings:

- In Tab Order
- Alphabatically

Display:

- All Headings
- Level 1 Headings
- Level 2 Headings
- Level 3 Headings
- Level 4 Headings
- Level 5 Headings
- Level 6 Headings

Move To Heading

Cancel
Fluid: Designing Software that works - For everyone.

Fluid Daily Build Resources

Fluid Integration Examples
- uPortal Instance
- Sakai Mock-up

Infrastructure...

Tuesday, March 16, 2010
Labelling Forms
Labelling Forms
Labelling Forms

User Interface Options

- Easier to see
  - Label
  - for
  - Minimum text size: [pt]
  - Text Spacing: [Regular]
  - Line Spacing: [ ]
  - Contrast: [Medium Contrast]
  - Background Images: [Yes] [No]
  - Simplified Layout: [Yes] [No]

Preview window (updates automatically)

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

Easier to find

Reset Save and apply Cancel

Tuesday, March 16, 2010
<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

```html
<fieldset>
  <legend>Background Images</legend>
  <span>
    <input type="radio" value="true" checked="checked"
    name="background-images-selection" id="background-yes">
    <label for="background-yes">Yes</label>
  </span>
  <span>
    <input type="radio" value="false" name="background-images-selection"
    id="background-no">
    <label for="background-no">No</label>
  </span>
</fieldset>
```
Tables

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

**Header**

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

**Body**
<table 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
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 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
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.
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:

// 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/Key-navigable_custom_DHTML_widgets

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
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

Basic dialog

Sed vel diam id luctus faucibus ante. Etiam auctor elit.

- checkbox
- radio
- select

textarea

This is the default dialog which is useful for displaying information. The dialog window can be moved, resized and closed with the 'x' icon.
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
Widget Libraries

- Reusable user interface widgets
  - Drag & Drop
  - Tabs
  - Sliders
  - Accordions
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
The Reorderer Family

layouts  grids  lists
## Site Settings

### Members

<table>
<thead>
<tr>
<th>Members</th>
<th>Email</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joe I. Instructor</td>
<td><a href="mailto:joe@sakai.university.edu">joe@sakai.university.edu</a></td>
<td>Instructor</td>
</tr>
<tr>
<td>Mike A. Smith</td>
<td><a href="mailto:msmith@yahoo.com">msmith@yahoo.com</a></td>
<td>TA</td>
</tr>
<tr>
<td>Jane Doe</td>
<td><a href="mailto:j.doe@sakai.university.edu">j.doe@sakai.university.edu</a></td>
<td>Student</td>
</tr>
<tr>
<td>Laura S. Roberts</td>
<td><a href="mailto:roberts_laura@gmail.com">roberts_laura@gmail.com</a></td>
<td>Guest</td>
</tr>
<tr>
<td>Erin A. Wilson</td>
<td><a href="mailto:thewilsones@hotmail.com">thewilsones@hotmail.com</a></td>
<td>Student</td>
</tr>
<tr>
<td>Ben Singer</td>
<td><a href="mailto:singer21@hotmail.com">singer21@hotmail.com</a></td>
<td>Student</td>
</tr>
<tr>
<td>Bhaktavatsalam Bhayakridhyanashanachar</td>
<td><a href="mailto:bb@local.host">bb@local.host</a></td>
<td>Student</td>
</tr>
<tr>
<td>Dept Admin</td>
<td><a href="mailto:da1@local.host">da1@local.host</a></td>
<td>Instructor</td>
</tr>
<tr>
<td>Angelo Haslip</td>
<td><a href="mailto:ah@local.host">ah@local.host</a></td>
<td>Student</td>
</tr>
</tbody>
</table>

[View full list](http://build.fluidproject.org/fluid/sample-code/pager/sakai-site-setting/site_setting_members.html#)
UI Options & FSS
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

Tuesday, March 16, 2010
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**

- Regular Widget
  - Option 1
  - Option 2
  - this is where the content goes

- Draggable Widget
  - Option 1
  - Option 2
  - this is where the content goes

**Mist**

- Regular Widget
  - Option 1
  - Option 2
  - this is where the content goes

- Draggable Widget
  - Option 1
  - Option 2
  - this is where the content goes

**High Contrast**

- More Regular Widget
  - Option 1
  - Option 2
  - this is where the content goes

- More Draggable Widget
  - Option 1
  - Option 2
  - this is where the content goes
FSS: Desktop

<head>
    <link type="text/css" rel="stylesheet" href="fss-layout.css" />
    <link type="text/css" rel="stylesheet" href="fss-theme-slate.css" />
</head>
<body class="fl-theme-slate">
</body>
mFSS Themes

iPhone

Android

Small Thumbnails

Go Back

Go Back

Go Back

Lots of info

Go Back

Donec convallis viverra quam, nec aliquet libero interdum placerat. Nunc arcu turpis, tincidunt at condimentum in, dapibus vel ipsum.

Go Back

Donec convallis viverra quam, nec aliquet libero interdum placerat. Nunc arcu turpis, tincidunt at condimentum in, dapibus vel ipsum.

Go Back

Donec convallis viverra quam, nec aliquet libero interdum placerat. Nunc arcu turpis, tincidunt at condimentum in, dapibus vel ipsum.

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>
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

```html
<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="#">Settings</a>
  </div>
  <h2>Regular Widget</h2>
  <div class="fl-widget-options">
    <ul>
      <li><a href="option1.html">Option 1</a></li>
      <li><a href="option2.html">Option 2</a></li>
    </ul>
  </div>
  <div class="fl-widget-content">
    <p>this is where the content goes</p>
  </div>
</div>
```

Tuesday, March 16, 2010
Links to get started

- http://www.alistapart.com
- http://webaim.org
Questions?