

Functional Examples

Work in Progress

This section is a work-in-progress. We're still fleshing out the definition of Functional Examples. If you have questions, please contact [Jacob Farber](#) or [Eli Cochran](#).

Functional Examples: plug and play component samples for application integration

(formerly *Springboards*)

Overview

A "one-stop shop" for developers, containing simple, working, well-styled versions of the Fluid components ready for cut-and-paste integration.

Goals

1. To help developers better understand Fluid components and how they work
2. To help foster widespread adoption of Fluid components
3. To assist developers in writing semantic markup and applications that are [AT](#) friendly
4. To provide useful and easily customizable themes for Fluid components and beyond

Guidelines

- clean semantic markup with wrappers for styling
- namespaced CSS
- no "context"
- unified CSS theme
- cut-and-paste-able
- well documented (excellent inline comments)

On this Page

- [Overview](#)
- [Goals](#)
- [Guidelines](#)
- [HTML Patterns](#)
- [CSS Patterns](#)
- [Components and html elements](#)
- [elements and states](#)
 - [Conceptual States:](#)
 - [HTML States:](#)
 - [CSS States:](#)
 - [ARIA States:](#)
 - [Generic HTML elements:](#)
 - [ARIA defined meta elements \(Roles\)](#)
 - [Fluid component specific objects:](#)

HTML Patterns

CSS Patterns

```
<div class="theme">
  <div class="selector state">foo</div>
</div>
```

Notes:

- all Fluid style definitions will be name spaced to the theme (ie. `.fluid-infusion button.disabled`) so that they will not collide with some other button style already defined for the page, but can inherit from previously defined buttons

- state: where possible use the ARIA state definition as the state
- we will need to avoid the very useful (but buggy [IE6]) pattern of chaining multiple classes on a single element as a way of modifying one class with another ie. foo.className.anotherClassName [<http://sonspring.com/index.php?id=102>] and [<http://www.ryanbrill.com/archives/multiple-classes-in-ie>]

Components and html elements

a quick inventory of what we have

General

- mouseover highlight
- element focus highlight

[Layout Reorderer](#) (see also:)

element	states	notes
<ul style="list-style-type: none"> • layout 		is the layout an "element"?
<ul style="list-style-type: none"> • pane <ul style="list-style-type: none"> • pane header • pane body 	<ul style="list-style-type: none"> • locked • keyboard focus • drag state (avatar) • placeholder 	
<ul style="list-style-type: none"> • drop target 		
<ul style="list-style-type: none"> • drop remediation 		

[Image Reorderer](#) (see also: [Image Reorderer API](#), [Lightbox Tutorial](#), [Drag and Drop Design Pattern](#))

element	states	notes
<ul style="list-style-type: none"> • image container <ul style="list-style-type: none"> • thumbnail • figure text 	<ul style="list-style-type: none"> • keyboard focus • drag state (avatar) • placeholder 	
<ul style="list-style-type: none"> • drop target 		
<ul style="list-style-type: none"> • instructional text 		

[Reorderer](#) (see also: [Reorderer API](#))

element	states	notes
<ul style="list-style-type: none"> • list item 		

[Uploader Design Overview](#) (see also: [Tutorial - Uploader](#), [Uploader API](#), [Uploader Design Pattern](#))

element	states	notes
<ul style="list-style-type: none"> • dialog 		
<ul style="list-style-type: none"> • file queue 		

<ul style="list-style-type: none"> • queue item 	<ul style="list-style-type: none"> • ready • uploaded • error • mouseover highlight 	
<ul style="list-style-type: none"> • scrolling area 		
<ul style="list-style-type: none"> • buttons 	<ul style="list-style-type: none"> • active • disabled 	
<ul style="list-style-type: none"> • progress 		

[Inline Edit](#) (see also: [Simple Text Inline Edit API](#), [Tutorial - Simple Text Inline Edit](#), [Inline Edit Design Pattern](#))

element	states	notes
<ul style="list-style-type: none"> • text 		
<ul style="list-style-type: none"> • field 		

[Pager](#) (see also: [Pager Tutorial - Markup-driven](#), [Pager API](#), [Pagination Design Pattern](#))

elements and states

functionality, presentation and accessibility

Mapping out what are all the elements that we want to support in components and what states and attributes we need to support for those elements

Brainstorming some lists here....

Conceptual States:

- focused
- disabled/enabled
- editable/un-editable
- movable/locked
- draggable/locked
- open/closed
- active
- checked
- selected
- read only
- ready
- changed
- uploaded
- success
- error
- hover
- default
- hidden
- expanded

<aside>Interesting conversation here about uploaded and success. "Uploaded" is really just "success" in the Uploader context. One could also see "checked" and "selected" as synonymous even if HTML does not. </aside>

HTML States:

- disabled
- selected
- readonly

CSS States:

css2

- focus

- linked
- active
- hover
- pseudo-elements (first-child, first-letter, before, after, etc.)

css3

- checked
- indeterminate
- disabled
- open
- default
- hidden
- more pseudo-elements

ARIA States:

- autocomplete
- checked
- disabled
- expanded
- haspopup
- invalid
- level
- multiline
- multiselectable
- pressed
- readonly
- required
- selected
- valuemax
- valuemin
- valuenow
- valuetext

Generic HTML elements:

- a [link]

Form elements

- input type="text"
- input type="image"
- input type="button"
- input type="checkbox"
- input type="radio"
- input type="file"
- input type="reset"
- input type="password"
- textarea
- button
- select
- optgroup
- option

Text elements

- label

ARIA defined meta elements (Roles)

- alert
- alertdialog
- application
- button
- checkbox
- columnheader
- combobox
- description
- dialog
- directory
- document
- grid
- gridcell
- group
- label
- link

- list
- listbox
- listitem
- log
- menu
- menuitem
- menuitemcheckbox
- menuitemradio
- option
- presentation
- progressbar
- radio
- radiogroup
- region
- row
- rowheader
- separator
- slider
- spinbutton
- status
- tab
- tablist
- tabpanel
- textbox
- timer
- toolbar
- tooltip
- tree
- treegrid
- treeitem

Fluid component specific objects:

- avatar
- vertical scrolling area
- portlet
- draggable element