FSS - Advanced Layout

⚠ **Deprecated**

This page is currently deprecated, and will be removed shortly. For more up-to-date documentation, see FSS Advanced Layout.

On This Page

FSS Documentation

Fluid Skinning System (FSS)
FSS Naming Conventions
FSS Cheat Sheet
Mobile FSS Cheat Sheet

Demos: Columns
Demos: Tabs, Menus, etc.
Demos: Text
Demos: Themes

HTML and CSS Background
CSS Bugs, Gotchas

Still need help?

Join the infusion-users mailing list and ask your questions there.

Overview

This walk-through describes how to use the Helpers functionality of the Fluid Skinning System (FSS). Here, "helpers" refer to the convenience class names and ready-made GUI setups such as Tabs, Widgets, Menus, etc. It is written under the assumption that readers have reviewed the Fluid Skinning System (FSS) introductory article and understand what the FSS is.

Learn how to customize FSS Layout classes and override values with the HTML and CSS Background article.

For more information about the walk-throughs in general, links to functional demos and FSS style sheets, please refer to FSS - Getting Started.

Before continuing with this article, it is **highly** recommended that you glance over the following walk-throughs as they explain certain concepts in detail:

1. FSS - Layout to give you an overview of how layout in the FSS works.
2. FSS - Colors and Themes to help give these helpers a unified appearance (a.k.a. themes) or customize their look.

Incorporating the Style Sheet

In order to use the FSS helper class names, you will need to include the following files in the `<head>` of your document:

```
fss-layout.css
```

All FSS class names use the `fl-` prefix. In our demos, if you see other class names without this prefix they are either: complimentary CSS; overriding CSS; or specific extensions of the FSS CSS. This means if you don't like the way a certain class names behaves you could easily override or extend them.

What is a Helper?

When you read the FSS - Layout article, you might notice thats it's mostly a lot of "convenience" class names which do several things at once to make laying out a page easier. These conveniences are taken further to create FSS Helpers. Instead of laying out intangible concepts like invisible wrappers and grid containers, you would use FSS Helpers to make GUI things like toolbars, widgets, menus and tabs.

To use FSS Helpers, you need to loosely follow the markup model we have provided, and then apply the appropriate helper class name.

Please note: this document assumes you have already read FSS - Layout article.

Utility Helpers

Layout Fix

```
<div class="fl-fix">
    <h1>A default header "leaks" its margins outside it's parent container</h1>
    <p>Same thing for any block element with margins</p>
</div>
```

Background info: Sometimes you will have a parent container whose children have large margins, like putting an H1 or P tag inside a DIV. The problem is the margin values on these children extend beyond the parent, giving the illusion the parent is being moved around when really it's the margin values of the children that are forcing the parent to shift.

`fl-fix` changes the `overflow` and `hasLayout` properties of a container. In the code sample above, it would force the container DIV to completely encapsulate the children, including the children's margin values.

Icons

```
<a href="#_bottom" class="fl-icon myCustomToolIcon" title="Tool Description">Tool</a>
```

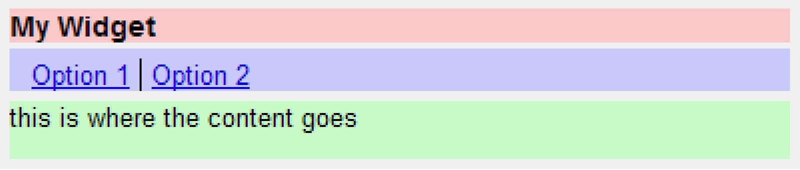
`fl-icon` is a small icon helper, which is useful in when you want to make nicely accessible icons. Often when you want to make icons you would use either: an empty anchor with a background image; or an anchor with an image as the content. For people who have low-vision, these icons are not helpful since they cannot always adjust the size of icon properly. These icons also don't allow for effective search engine indexing since there is little useful content associated with the icon (just the title and href) - see "Use descriptive anchor text" in the Official Google Blog.

What the `fl-icon` does is set up your icons at a certain size (16 x 16 pixels), and allow you to place text in your anchors without it showing or compromising the visual aesthetic. Its does this by making the links `block` level elements with a large negative text indent to "pull" the text outside of their clipping region.

When used in conjunction with the FSS high contrast themes, the icon image is effectively replaced with the icon text, thus maintaining icon useability.

Widget Helper

Basic Widget



My Widget

Option 1 | Option 2

this is where the content goes

Widgets in the FSS have been generally modelled after the MyCamTools UX enhancement model which has 3 main components: a titlebar, options, and content (in that order).

The markup looks like:

```
<div class="fl-container-400">
    <div class="fl-widget">
        <div class="fl-widget-titlebar">
            <h2>My Widget</h2>
        </div>
        <div class="fl-widget-options">
            <ul>
                <li><a href="#_bottom">Option 1</a></li>
                <li><a href="#_bottom">Option 2</a></li>
            </ul>
        </div>
        <div class="fl-widget-content">
            <p>this is where the content goes</p>
        </div>
    </div>
</div>
```

Here is an exploded view of what's going on:

```
<div class="fl-container-400">
```

This is the basic example container, which is 400 pixels wide.

```
<div class="fl-widget">
```

This class name is required on the widget container.

```
<div class="fl-widget-titlebar">
```

`fl-widget-titlebar` identifies the container for the titlebar. This section is not required, but it is a good place to put a title (obviously) and title-level functionality (more on that a little later). An `H2` is the default tag used to identify the title itself- but this can be modified to be any element.

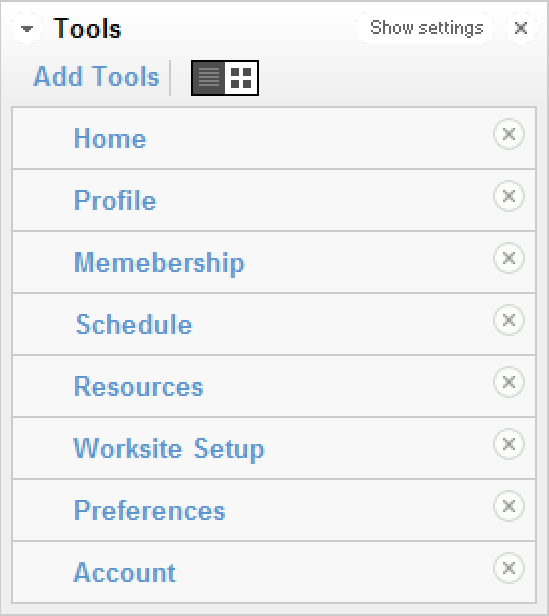
```
<div class="fl-widget-options">
   <ul>
      <li><a href="#_bottom">Option 1</a></li>
      ...
```

`fl-widget-options` identifies the options for the widget, which is outlined as an unordered list of things. This section is not required.

```
<div class="fl-widget-content">
```

`fl-widget-content` identifies the content for the widget, which can be anything - even another widget. If you need to pull in external web pages, an iframe would work well here.

Advanced Themed Widget



In this example, we can see a fully formed widget. There are plenty of options bound to the widget in multiple places. This example also uses other helpers, such as the `fl-icon` helper class for accessible, searchable icons and the `fl-listmenu` for quick navigation menus.

```
<div class="fl-container-300 fl-theme-mist">
    <div class="fl-widget">
        <div class="fl-widget-titlebar">
            <a class="fl-force-left fl-icon more" href="#_bottom">More</a>
            <a class="fl-force-right fl-icon close" href="#_bottom">Close</a>
            <a class="fl-force-right fl-icon settings" href="#_bottom">Settings</a>
            <h2>Tools</h2>
        </div>
        <div class="fl-widget-options">
            <ul>
                <li><a href="#_bottom">Add Tools</a></li>
                <li><a class="fl-icon viewOptions" href="#_bottom">Toggle View</a></li>
            </ul>
        </div>
        <div class="fl-widget-content">
            <ul class="fl-listmenu">
                <li><a href="#_bottom">Home          <span class="fl-icon delete">[remove]</span></a></li>
                <li><a href="#_bottom">Profile       <span class="fl-icon delete">[remove]</span></a></li>
                <li><a href="#_bottom">Memebership   <span class="fl-icon delete">[remove]</span></a></li>
                <li><a href="#_bottom">Schedule      <span class="fl-icon delete">[remove]</span></a></li>
                <li><a href="#_bottom">Resources     <span class="fl-icon delete">[remove]</span></a></li>
                <li><a href="#_bottom">Worksite Setup<span class="fl-icon delete">[remove]</span></a></li>
                <li><a href="#_bottom">Preferences   <span class="fl-icon delete">[remove]</span></a></li>
                <li><a href="#_bottom">Account       <span class="fl-icon delete">[remove]</span></a></li>
            </ul>
        </div>
    </div>
</div>
```

Here is an exploded view of what's going on:

```
<div class="fl-container-300 fl-theme-mist">
```

This is a smaller container and it has a theme applied. For more information on theming, please see FSS - Colors and Themes.

```
<div class="fl-widget">
```

This class name is required on the widget container, same as in the previous example.

```
<div class="fl-widget-titlebar">
     <a class="fl-force-left fl-icon more" href="#_bottom">More</a>
     <a class="fl-force-right fl-icon close" href="#_bottom">Close</a>
     <a class="fl-force-right fl-icon settings" href="#_bottom">Settings</a>
     <h2>Tools</h2>
```

`fl-widget-titlebar` identifies the container for the titlebar. Now, we can see some useful functionality embedded into the titlebar such as more options, close widget, and widget settings all surrounding the h2 title itself.

`fl-icon` is a small icon helper, which is useful in when you want to make highly accessible and readable icons. Please see #Icons for more information.

```
<div class="fl-widget-options">
            <ul>
                <li><a href="#_bottom">Add Tools</a></li>
                <li><a class="fl-icon viewOptions" href="#_bottom">Toggle View</a></li>
      ...
```

`fl-widget-options` is the container for the widget options, which is a list. In this example each list item is a text link, and one has an `fl-icon` class name, which is modified to a larger size (32 x 16 pixels) by `viewOptions`

```
<div class="fl-widget-content">
```

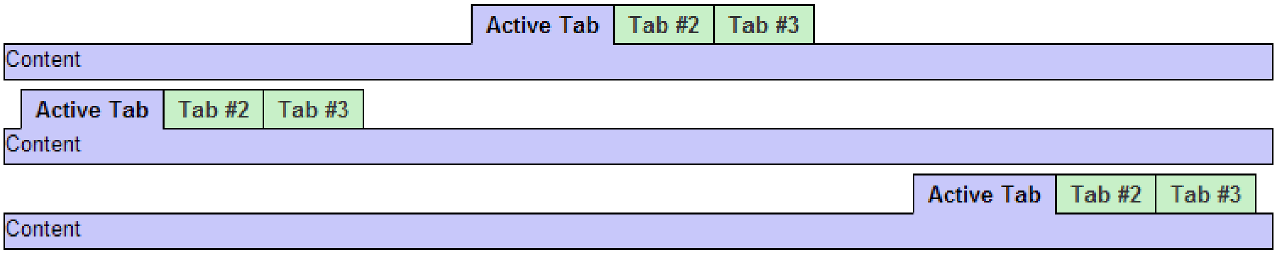
`fl-widget-content` is the content container for the widget, which in this case is another helper system:

```
<ul class="fl-listmenu">
         <li><a href="#_bottom">Home <span class="fl-icon delete">[remove]</span></a></li>
         ...
```

`fl-listmenu` is another helper class name. It is a basic navigation menu. Please see #Menu Helper for more information. This list menu contains links, as well as nested icons inside the menu links.

Tabs Helper

Basic Tabs



Tabs are a very common GUI paradigm, but they can be tricky to implement across all Grade-A browsers. Fortunately, FSS ships a simple Tabs Helper to help solve this problem. The Tabs helper has only one requirement, which is the tabs must be in a list.

The markup looks like:

```
<div class="fl-container-750 fl-centered">
            <ul class="fl-tabs fl-tabs-center">
                <li class="fl-activeTab"><a href="#_bottom">Active Tab</a></li>
                <li><a href="#_bottom">Tab #2</a></li>
                <li><a href="#_bottom">Tab #3</a></li>
            </ul>
            <div class="fl-tab-content">
                Content
            </div>
        </div>
```

Here is an exploded view of what's going on:

```
<div class="fl-container-750 fl-centered">
```

This is the usual container, 750 pixles wide and centered on the page

```
<ul class="fl-tabs fl-tabs-center">
```

`fl-tabs` is the required class name to start, placed on the required list container. The second class `fl-tabs-center` is optional, and is only for tab alignment. The available options for alignment are `fl-tabs-left` for left alignment (default), `fl-tabs-right` for right alignment, `fl-tabs-center` for centered alignment.

```
<li class="fl-activeTab"><a href="#_bottom">Active Tab</a></li>
```

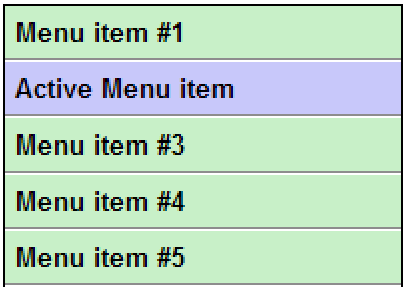
`fl-activeTabs` is used to hilight the currently active tab, and is optional.

```
<div class="fl-tab-content">
```

`fl-tab-content` is an optional peice of the tabs helper system, and is simply styled to nicely "snap" to the tabs themselves. It is intended to work in conjunciton with the `fl-activeTab` class to complete the look of activated tab content block.

Menu Helper

Basic Menu



The Menu helper is a simple vertical menu. It requires a list with an anchor inside each list item. You can place children inside or siblings alongside the anchors.

The basic markup may look like:

```
<div class="fl-container-200 fl-centered">
    <ul class="fl-listmenu">
        <li><a href="#_bottom">Menu item #1</a></li>
        <li class="fl-activemenu"><a href="#_bottom">Active Menu item</a></li>
        <li><a href="#_bottom">Menu item #3</a></li>
        <li><a href="#_bottom">Menu item #4</a></li>
        <li><a href="#_bottom">Menu item #5</a></li>
    </ul>
</div>
```

Here is an exploded view of what's going on:

```
<div class="fl-container-200 fl-centered">
```

This is the usual container, 200 pixles wide and centered on the page

```
<ul class="fl-listmenu">
        <li><a href="#_bottom">Menu item #1</a></li>
        <li class="fl-activemenu"><a href="#_bottom">Active Menu item</a></li>
```

Here, the `fl-listmenu` class name goes on the list container. Inside the container are list items, and then anchors.
Just like for `fl-activeTab`, `fl-activemenu` is an optional class name that identifies a list item to stand out and look "active".