

Fluid Skinning System (FSS)

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

The Fluid Skinning System is a modular CSS framework: a group of CSS files with a specially designed structure and class-specific format. The modular format allows you to add, remove, and mix classes to get the desired effect.

With FSS and User Interface Options, you can:

- control the appearance of your website using pre-existing modular CSS classes,
- support user-customization of the appearance of a website or Infusion components,
- deliver a look-and-feel for Infusion components
- and much more.

Features

The FSS offers:

- Wrappers and containers for quickly laying out your page, including fixed- and flexible-width columns
- Convenience classnames to quickly and easily create GUI elements such as tabs, menus, etc.
- Convenience classnames for basic font handling, such as font family, size, and formatting
- Theming
- Accessibility support, including
 - easy page linearization
 - tested for usability effectiveness, specifically colorblindness
 - works with Assistive Technologies, specifically with off-screen AT readable content

Why Use FSS?

In addition to making it easy to lay out your websites, FSS is designed to work with the Infusion ([Floe](#)) [UI Options \(2008-2009\)](#) component to allow users to personalize the display of content to their own individual needs. This empowers users with visual or mobility impairments, for example, to alter the colour schemes or layout, enabling them to access content that otherwise might be completely inaccessible to them.

Getting Started With FSS

The FSS is intended to be used as the basis for your website, extended and customized by your own styles. The short-version of how to use FSS is:

1. Include the FSS `.css` files you want, in the right order, ahead of your own `.css` file(s).
2. Add FSS classes to your mark-up, ahead of your own class names

3. If desired, override or extend any FSS classes as needed.

A slightly longer version of how to use FSS is provided in the sections below. For more details on the specific classes available in the FSS, see the [FSS Cheat Sheet](#) page.

1. Including FSS files

FSS includes several separate files, each providing particular functionality:

1. **fss-reset-global.css** or **fss-reset-context.css**: [new in v1.4](#) normalize visual discrepancies between browsers. Optional, but recommended. (For earlier versions, use **fss-reset.css**.)
 - **fss-reset-global.css**: The global file will affect the entire page.
 - **fss-reset-context.css**: Changes are scoped to the `fl-reset` class.
2. **fss-base-global.css** or **fss-base-context.css**: [new in v1.4](#) applies a default set of styles. Optional, but recommended, and work well when used with the **fss-reset** files. (For earlier versions, use **fss-reset.css**.)
 - **fss-base-global.css**: The global file will affect the entire page.
 - **fss-base-context.css**: Changes are scoped to the `fl-reset` class.
3. **fss-layout.css**: contains classes for defining margins, paddings, positions, etc., as well as classes useful for laying pages out with columns, etc.
 - For examples, see the [Columns demos](#) and the [more advanced layout demos](#).
 - For detailed information, see [FSS Cheat Sheet](#).
4. **fss-text.css**: includes classes for setting font size, font family, line height, etc.
 - For examples, see the [Text demos](#) demos.
 - For detailed information, see [FSS Cheat Sheet](#).
5. **fss-theme-<themeName>.css**: [new in v1.1](#) contains classes defining sets of colors, background colors, etc.
 - For examples, see the [Themes demos](#) demos.
 - For detailed information, see [FSS Cheat Sheet](#).

File Order is Critical

The order of the files in your document is important: FSS stylesheets should be included before any Infusion component stylesheets, and in the order described above. Any custom stylesheets should be included last.

Your declaration of FSS files should look somewhat like the following:

```
<link type="text/css" rel="stylesheet" href="fss-reset-global.css" />
<link type="text/css" rel="stylesheet" href="fss-base-global.css" />
<link type="text/css" rel="stylesheet" href="fss-layout.css" />
<link type="text/css" rel="stylesheet" href="fss-text.css" />
<link type="text/css" rel="stylesheet" href="fss-theme-themeName.css" />
<link type="text/css" rel="stylesheet" href="ComponentName.css" />

<link type="text/css" rel="stylesheet" href="YourStyleSheetGoesHere.css" />
```

The above example uses all the different FSS CSS functions. If you have no use for one of the files, you do not need to include it.

2. Adding FSS classes to your mark-up

Note also that you should always put any FSS class names at the beginning of the class attribute in your HTML declarations, before any of your custom classes. For example:

```
<div class="fl-container-250 fl-font-size-110 myCustomClass"> </div>
```

3. Overriding FSS classes

While the FSS classes are generally well thought-out and pretty comprehensive, it is likely that you will want to customize your site by overriding some of the styles. This is how we expect the FSS to be used.

There are generally two ways of overriding predefined styles:

1. Overriding an entire class
2. Overriding only one instance of a class

In all cases, you should override FSS classes in your own custom stylesheet. Do not modify the original FSS files.

Overriding an entire class

Overriding an entire class doesn't mean that you have to change all of the properties specified. To override `fl-para`, you can copy and paste the code into your imported or embedded style sheet and modify it however you like. Here's an example:

Original Style

```
.fl-para {
  font-family: "Times New Roman";
  font-size: 1.2em;
  text-align: left;
}
.fl-footer {
  font-family: "Arial";
  font-size:
  0.9em;
}
```

Override

```
.fl-para {
  font-family: "Arial";
  font-size: 1.4em;
}
```

This "re-styles" `fl-para` by overriding the `font-family` and `font-size`. Whatever is left out (`text-align` in this example) is inherited from the original `fl-para`; so the text remains aligned to the left.

If this override is pasted into an imported style sheet, please note that the code for importing that style sheet would have to come **after** the code for importing the original (see [Order of overrides #4](#)).

Overriding an instance of a class

In some cases, you may only wish to override an FSS style for one element that the class is applied to. This can be accomplished by using a selector that is specific to only that element, for example by adding a class name, or using other specifics.

HTML

```
<div class="fl-col-flex3">
  <div class="fl-col">...</div>
  <div class="fl-col">...</div>
  <div class="fl-col">...</div>
</div>
```

Override

```
.fl-col-flex3 .fl-col:first {  
  margin-left: "10px";  
  margin-right: "10px";  
}
```

In this example, the custom stylesheet overrides the default margins for only the first column of the three-column layout.

Extending FSS classes

The FSS defines many variations of a given style - for example, it defines several widths for use with fixed-width columns. But it is likely that someone will want a width (or font size, or anything else) that is not specifically defined in the FSS files. In these cases, you are encouraged to copy the style, rename it and customize it to your specific needs.
