

FSS Reference - Methods and Limitations for dynamically changing styles

Background

As part of the Fluid Skinning System project and the Transformable 2 project, Fluid is looking at ways to dynamically inject style sheets or otherwise change styles.

Goals are:

- inject our styles into pages that we otherwise know very little about
- change styles on the fly as part of transforming the page for accessibility or user preference.

Overview

This page covers various ways to dynamically change styling on a web page at run time and the various limitations of method based on testing and research.

Some things to test:

- adding a `<link>` tag to the HEAD
- adding an `<import>` tag to HEAD
- the effects of `!important` on CSS hierarchy
- stylesheet. *disabled*

Preliminary Test Results

The good news:

Using a very simple technique of injecting a link tag (`<link href="css/tool.css" type="text/css" />`) to the HEAD using Javascript seems to reliably introduce and render new CSS into the page for Firefox3 and Safari3 on the Mac and FF3, IE6 and IE7 on Windows (the browsers I've currently tested). Sweet!

The bad news:

Only Firefox3 and Safari3 seem to respect the CSS hierarchy.

The order in which style sheets and styles are loaded in the HEAD of the document determines which styles have precedence over other styles. The styles loaded later have precedence over the styles loaded earlier. If the first style sheet says that `` tags are **blue**. And the next style sheet says that all `` tags are **red** then all the `` tagged content should be **red**.

This should work whether the styles sheets were specified in the head before the page is loaded or if the stylesheets are injected after load time with Javascript.

And it does with Firefox3 and Safari3. You can inject the new stylesheet anywhere in the HEAD; beginning, end, middle and Firefox3 and Safari3 will treat the rules in the injected stylesheet as if they had been loaded from the start. So, using our colored strong tags example above, if you *append* the `strong_red.css` file to HEAD at run-time, and *then prepend* the `strong_blue.css` file, the `` text on the page will stay **red**. Did you follow that? Even though you added the stylesheet specifying the blue text after the stylesheet that specifies the red text, you placed the stylesheet higher in the head so the red styling takes precedence.

But in IE, whichever file gets loaded last (in time) gets precedence, the `` text will turn **blue**.

Interestingly, if you inspect the DOM as reported to Debugbar from IE, the stylesheets are getting loaded into the DOM in the correct order, they just aren't being parsed correctly.

This is still a technique that may be useful to us, but one of the things that we could not do is inject a CSS reset before all the other style sheets. We would then crush any subsequent styling.

I still want to test what happens if I inject stylesheets after the HEAD loads but before the BODY loads and a few other things.

Content Subheading

Subheading text.

Content Heading

Content text.

RELATED LINKS

i Use this format to add related links to your page. Remove the "noformat" tags and replace "add,related,labels,here" with a comma separated list of labels related to this page. Delete the panel if there are no related links. Remove this note before publishing.

```
{contentbylabel:add,related,labels,here|key=@all|maxResults=20|showSpace=true|sort=modified}
```

PAGE HIGHLIGHTS

- [Link 1](#)
- [Link 2](#)
- [And so on...](#)

Error rendering macro 'panel'

com.atlassian.rendererv2.macro.

basic.validator.

MacroParameterValidationException:

Color value is invalid