Fluid Infusion

What Is Infusion?

Infusion is a different kind of JavaScript framework. Built for creating applications that are highly usable and accessible, Infusion provides a rich set of APIs for creating loosely-coupled models and views using a declarative and event-driven style.

Infusion includes:

- an application framework for developing applications with JavaScript and jQuery,
- a growing collection of user interface components, and
- a modular CSS framework that allows you to add, remove and mix classes to get effect you want.

Infusion embraces unobtrusive, functional techniques that promote less code and greater flexibility. Infusion takes the pain out of developing accessible, high performance, clean and nimble front-ends for applications that want to do more. Our approach is to leave you in control - it's your interface, using your markup, your way.

Infusion is developed by the Fluid Project, an international group of designers, developers, volunteers, and advisers dedicated to improving the user experience of open and community source projects.

Infusion is available for download and is open to use and modify however you like. Infusion is released under both the ECL 2.0 and BSD licenses.

Why Use Fluid Infusion?

Fluid components differ from other javascript library components in that they are built from the ground up with several core requirements:

- Accessibility (a11y), usability, and internationalization (i18n) are key
- Must strive to be DOM-agnostic
- Must provide a good user experience in any situation
- Must be completely customizable

For More Information

Infusion Documentation

Download the code from Github

Demos

Component Library

Daily Builds

Project Coordination

Testing Infusion

Core Component Requirements

- Accessibility (a11y)
  - Accessible Rich Internet Applications (ARIA) primer (http://www.w3.org/TR/wai-aria-primer/)
  - Accessibility on web in a nutshell (http://www.alistapart.com/articles/wiwa)
- Usability
  - Usability on the web (http://en.wikipedia.org/wiki/Web_usability)
  - Jakob Neilson (usability specialist) (http://www.useit.com)
- Internationalization (i18n)(http://en.wikipedia.org/wiki/Internationalization_and_localization)
- DOM Agnosticism and playing safe
  - Being DOM agnostic means further abstracting your code from the presentation and markup layers. This means the behaviour layer is made to accomodate many different situations (in the markup and behavioural layers) and deal with them all gracefully. Our components cannot make assumptions about the surrounding (x)html markup. They can be placed in situations where normal operation might otherwise break the environment (ie. its not safe to traverse the DOM when it's in flux). While being markup agnostic cannot happen perfectly 100% of the time, it is something for Fluid components to strive for.
- Customization
  - Part of being DOM agnostic is to allow for extensive customization on many levels. Something Fluid components try to achieve is allowing the implementer to easily change as much about the component as possible, allowing for a simpler and safer integration process on their part.