

Fearless JavaScript Workshop

What is Fearless JavaScript?

Fearless JavaScript is an ongoing series of workshops offered by members of the Fluid community. It is designed to provide attendees with a grounding in the essential techniques of modern JavaScript programming. Topics include:

- Why JavaScript?
- Fundamentals of JavaScript
- Using jQuery
- JavaScript in Sakai
- Portal and mash-up friendliness
- Accessibility

Everyone is welcome to attend the Fearless JavaScript workshop. It's assumed that you have some understanding of at least one programming language, and that you'll bring a laptop to follow along with the hands-on exercises.

When is the Next Fearless JavaScript Workshop?

Fearless JavaScript 2.0

Sakai Summer 2008 Conference, Paris, France
Monday, June 30, 2008
13:30 - 16:30

Presentation

- [FearlessJSv2_Post.key.zip](#) [.zip 6.2MB - archive of the Keynote 4 presentation]
- [FearlessJSv2_Post.pdf](#) [.pdf 12.6MB - the final presentation with some of the presenters notes]

Presenters

- Colin Clark, Fluid Project Technical Lead, ATRC, University of Toronto
- Eli Cochran, User Interaction Developer, ETS, UC Berkeley
- Nicolaas Matthijs, Researcher, CARET, University of Cambridge

Goals

- Provide developers with a strong foundation in client-side development with JavaScript, HTML, and CSS
- Train prospective volunteers and collaborators in quality, accessible JavaScript programming techniques
- Help communities avoid common JavaScript programming pitfalls by offering viable alternatives
- Encourage the early adoption of Fluid products and philosophy

Audience

- Web developers of all stripes (experience with at least one programming language, basic understanding of the Web's architecture)
- Working knowledge of portlets and portals
- Basic understanding of HTML and CSS

Philosophy

- Unobtrusiveness
 - Separation of code and content
 - Graceful degradation
 - Using the DOM as a API for components, leveraging jQuery's strengths
- Accessibility
 - Multimodal: works with keyboard and mouse
 - Configurable: doesn't hardcode expectations about control or bindings
 - Flexible: works with a variety of presentations and layouts
 - *Accessibility doesn't have to be hard*
- Portal-friendliness
 - Comprehensive namespacing: JavaScript code and markup
 - Privacy
- Javascript isn't Java
 - It's buggy, but it's powerful
 - Prototypal inheritance
 - Dynamic language
 - Objects as general containers

Workshop Outline

- **Introductions**
 - Getting setup
 - Using the example code
- **Why Javascript? (Eli)**
 - Why JavaScript? Why DHTML? Why AJAX? Why bother?
 - Why DHTML/client-side at all?
 - Why not Flash or Silverlight?
 - Modern JavaScript
- **Javascript 101 (Colin)**
 - JavaScript is different
 - History
 - Part 1: The Basics
 - Variables
 - Numbers
 - Strings
 - null
 - undefined
 - Objects and Arrays
 - Part 2: functions and scope
 - Functions are first class
 - Ways to call a function
 - Determining types
 - Prototypal Inheritance
 - Understanding `this`
 - Scoping rules
 - Closures
 - Objects as general containers
 - Type system: `typeof/instanceof`; `truthy & falsey`; coercion
 - Functions are first-class objects
 - Functions have context: how "this" works in JavaScript
 - Inheritance
 - Instantiating functions: the "new" keyword
 - Prototypes, not classes
 - Why extending built-in types sucks
 - Closures: encapsulating, hiding, and currying
 - Using closures for namespacing and privacy
- **Learning jQuery (Eli)**
 - Why use a JavaScript toolkit?
 - Why jQuery?
 - Other frameworks (these techniques work everywhere)
 - [Comparison of Event Binding Without a Toolkit and With jQuery](#)
 - Finding stuff in the DOM: querying using selectors
 - Binding events
 - DOM manipulation
- **MyCamTools: JavaScript in Sakai (Nico)**
 - Goals
 - Backend: `sData` and `EntityBroker`
 - JSON data feeds
 - Front end: Widgets
 - How they're structured
 - Libraries used
 - Examples
- **JavaScript for Portals and Mashups (Colin)**
 - Portals and mash-ups put extra constraints on your JavaScript
 - Multiple instances
 - Lots of different JavaScript code running
 - High chance of collisions
 - Can't expect control of the document
 - Put code in a unique namespace
 - Use closures for privacy
 - Be unobtrusive
 - Support multiple instances
 - Constrain selectors to a specific fragment
 - How Fluid components are built
 - Unobtrusiveness
 - Markup agnosticism
 - Highly configurable
- **Javascript Accessibility (Colin)**
 - What is accessibility?
 - Fluid's accessibility vision
 - Why most DHTML is inaccessible

- Keyboard accessibility
- ARIA and assistive technologies
- jQuery plugins: keyboard-a11y and jARIA
- **Putting it all together: building a widget with MyCamTools and Fluid**
- **Where to Go Next**
 - [Fluid DHTML Developer's Checklist](#)
 - [Suggested reading](#)
 - Fluid community

Previous Editions of Fearless JavaScript

- [Fearless JavaScript 1.0](#): Writing Fearless Javascript for Portlets, Widgets, and Portals
 - Presented on Sunday, April 27, 2008 at the JA-SIG Spring 2008 conference in St. Paul, MN.