Mobile Web Apps

Harnessing the Power of Infusion

**Jacob Farber,**
Web User Interface Developer,
Adaptive Technology Resource Centre,
University of Toronto

**Justin Obara,**
Web User Interface Developer and Tester,
Adaptive Technology Resource Centre,
University of Toronto
Topics We’ll Cover

• About Fluid and Infusion
• Benefits of Mobile Web Apps
• Mobile Web App Pain Points
• How Infusion Can Help
What is Fluid

http://fluidproject.org

- Is an Open Source community of
  - Designers
  - Developers
  - Accessibility Experts
- Helps other open communities
- Consists of universities, museums, and individuals
What is Fluid Infusion

- Application Framework built on top of jQuery
- Lightweight CSS Framework (FSS)
- UI Components you can reuse and adapt
- Open architecture: everything is configurable
- Designed for Usability and Accessibility
Infusion Components

http://build.fluidproject.org/infusion/demos/
Why Mobile Web Apps?
Simplify development

Objective-C
Cocoa Touch

Java
Android SDK

C++
Carbide
Why Mobile Web Apps?
Simplify development
HTML, CSS, JS
Benefits of Mobile Web Apps

- Modern phones have great browsers
- Web is ubiquitous and familiar
- HTML, CSS, and JavaScript

Google Latitude
Mobile Web Pain Points

- Sparse documentation
- Few successful Open Source examples
- Existing frameworks have limitations
Mobile Web Pain Points
Sparse Documentation

- Lots of uncharted territory....
Mobile Web Pain Points

Existing Frameworks

Frameworks:
jQTouch   iUI   WebApp.Net   Magic Framework

• Workflow
• JavaScript Footprint
• Flexibility of themes
How Infusion Can Help

The Fliquor App

http://tinyurl.com/fliquor

- Demonstration of Using Fluid Infusion
- Search public images on Flickr
How Infusion Can Help

- Loose coupling (DOM Agnosticism)
- Template driven
- Fast and flexible styling (FSS)
DOM Agnosticism

Component code is maximally unaware of the details of the markup layout that it operates on.
DOM Binder

- Relaxes the coupling between the code and the markup
DOM Binder

- Uses named selectors
- Explicit selectors never appear in code
- Code is free of baked dependences on markup
- Attached to a component with call to `fluid.initView`
DOM Binder

Options

```
fluid.defaults("fliquor.imageViewer", {
    selectors: {
        searchButton: ".flc-fliquor-searchButton",
        searchBox: ".flc-fliquor-search",
        template: ".flc-fliquor-template",
        images: ".flc-fliquor-images",
        ...
    },
...
});
```

```
that.locate("searchBox").fluid("activatable",
    that.search);

that.locate("searchButton").click(function (evt) {
    that.search();
    return false;
});
```
HTML Template

• Components render based on Template
• Code has no dependence on markup structure
• Markup is not in code
HTML Template

```html
<body class="flc-fliquor fl-theme-coal">
  <div class="fl-centered fl-navbar fl-table search-bar">
    ...
  </div>
  <div class="fl-centered results">
    <ol class="flc-fliquor-images fl-list-menu fl-list-thumbnails fl-thumbnails-expanded">
      <li class="flc-fliquor-template">
        <a href="#">
          <span class="wrapper">
            <span class="flc-fliquor-item-title fl-fliquor-item-title">Title</span>
            <img title="Title" alt="Alt" src="#" class="fl-icon" />
          </span>
        </a>
      </li>
    </ol>
  </div>
  ...
</body>
```
HTML Template

- Fully customizable
- As semantic as possible
- ARIA added via JavaScript
- Presentation class names separated from Javascript selector names
Rapid CSS using FSS

- Modular, extensible class-name CSS system
- Easy to read class names
- Plays nice with other CSS systems (jQuery themes, other frameworks, etc)
- Modularity means it’s meant for you to hack
- Lots of support, fully cross browser
Reset:

normalize the little things for a consistent starting point
FSS Applied

Layout:
placement & containers
FSS Applied

Themes:

style, colors and graphics
Mobile FSS

- Highly focused on target platforms
- Follows basic touchscreen UI guidelines from Apple and Android dev centres
- Repurposes FSS class names, enabling multiplatform content
Mobile FSS Applied
Make it your own

 фл-theme-myTheme
Presentation Summary

- Loose coupling (DOM Agnosticism)
- No baked dependence on markup
- Template Driven
- You own the markup
- Rapid CSS
- Get up and running fast, easily customized
Fluid Engage
Thank You

Fluid Project

http://www.fluidproject.org/

Demos

http://build.fluidproject.org/

Join the community

http://fluidproject.org/getinvolved/

jacob.farber@utoronto.ca  justin.obara@utoronto.ca