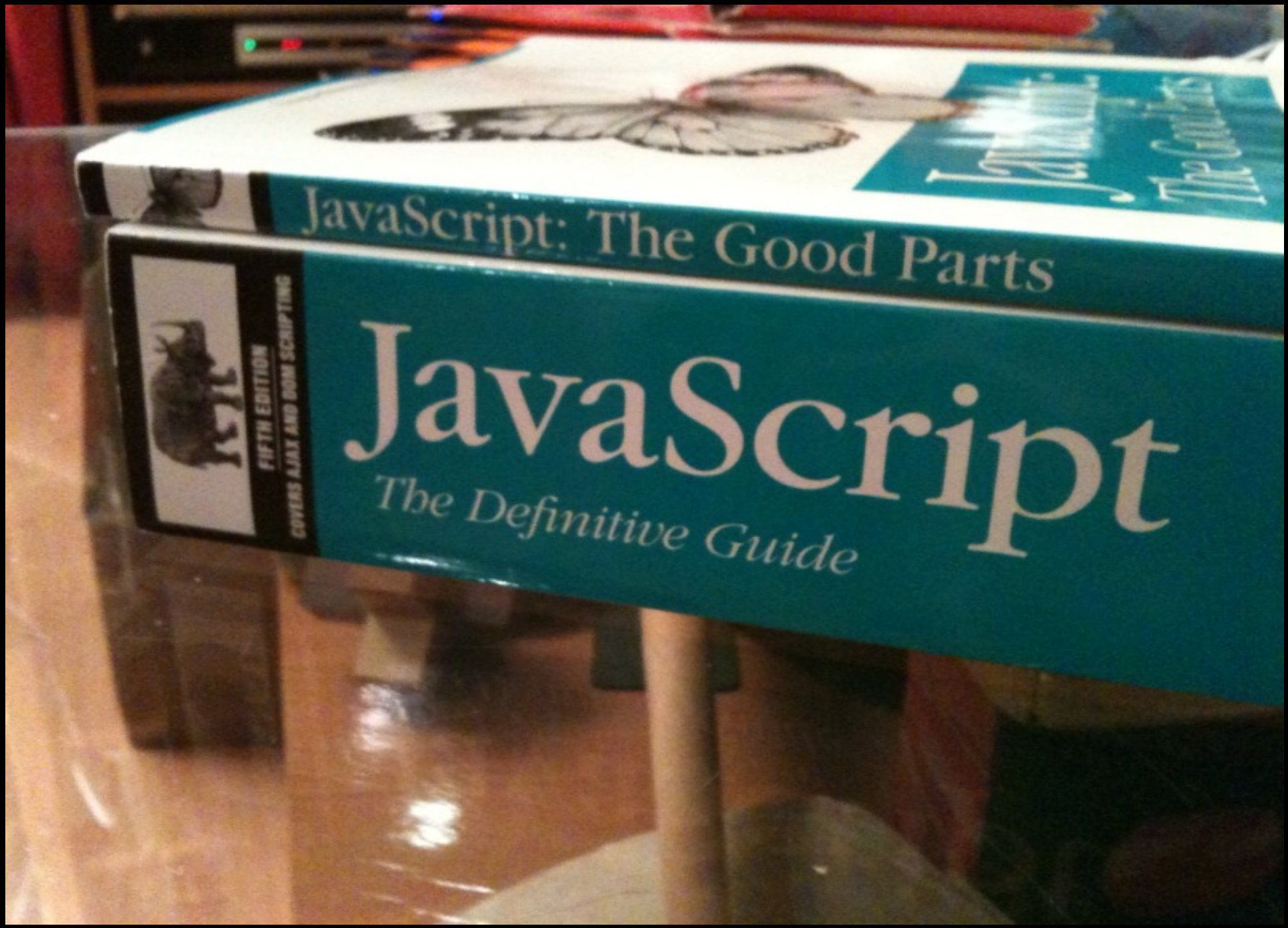
Working with Javascript Building Responsive Library apps

Computers in Libraries
April 15, 2010
Arlington, VA

Jason Clark
Head of Digital Access & Web Services
Montana State University Libraries

Overview

- Why Javascript?
- Smart Design and Markup
- Progressive Enhancement
- Ajax and User Interfaces
- Code Libraries - jQuery
- Demos and Code
  - UX Problems and Solutions
- Questions



From alikins' photostream (http://www.flickr.com/photos/alikins/4439062727/)

Why Javascript?

client-side scripting FTW

Why Javascript?

- Want to make your applications more interactive
- Want to incorporate data from external Web Services
- Don’t want your users to have to download a plugin
- Simplest form of dynamic functionality on a web page

Javascript App Components

HTML and CSS
These familiar Web standards are used for styling the look and feel of a page and to markup those areas on a page that will be targeted for data updates.

DOM (document object model)
The DOM is used to manipulate dynamic page views for data and to walkthrough documents to “cherrypick” data. The DOM enables certain pieces of an Ajax page to be transformed and updated with data.

XML, JSON (Javascript Object Notation), HTML, or plain text
Any of these standards are used to provide structure to the data it passes to and from a page.

XMLHttpRequest object
The heavy lifter for Ajax: It’s a javascript object embedded in most modern browsers that sets up data request/response pipelines between client and server.

Client-side Scripting (Javascript)

Client versus Server

- **Client-side scripting**
  - **Web browser does all the work**
- **Server-side Scripting**
  - **Web server does all the work**

Ajax leverages both client and server side scripting

Smart Design and Markup

Think semantically, code hooks

Coding for App Functionality

- Place hooks using ids and classes
- Create semantic divisions
- Document can be recognized by human

Markup sample...

```
<dd class="toggle">
```

Progressive Enhancement

Failing gracefully...

Progressive Enhancement

Strategy for web design and development
Adding functionality for user's with most current browsers.

Graceful Degradation
Building apps that still work with older browsers and clients

Finding the BALANCE...

See Progressive Enhancement:

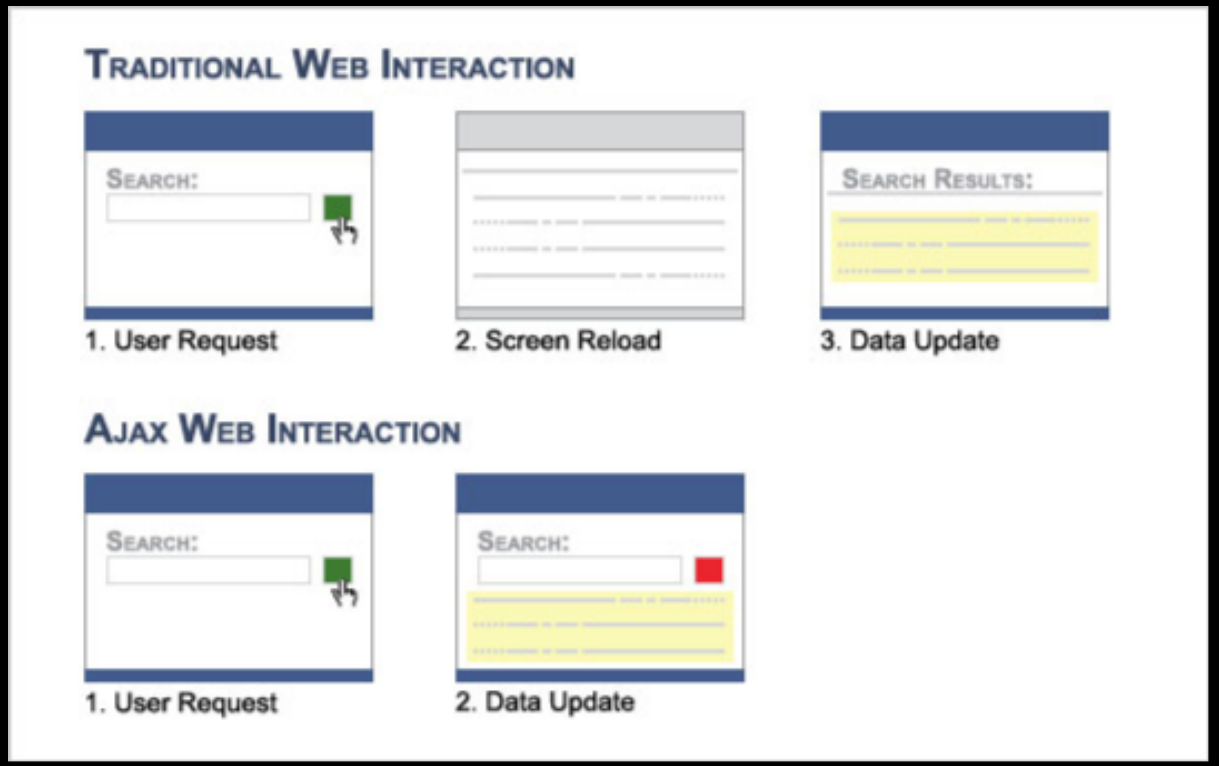
Ajax and User Interfaces

Desktop functionality in the browser

What is Ajax?

- Asynchronous Javascript and XML
  - Not all AJAX apps involve XML
- Combination of technologies
  - HTML, CSS, DOM
  - XML, JavaScript
- Some server scripting language
- A method for building more responsive and interactive applications

How Ajax Works...



Ajax is Ubiquitous

- Google Maps
- Flickr
- Delicious

Code Libraries - jQuery

Makin' it easier for all of us

Why Use a Code Library?

- Cross Browser Troubleshooting = DONE
- Access to high level functions
- Cleaner code
- Plugin architecture

Which Code Library?

- jQuery
  - jquery.com
- MooTools
  - mootools.net
- Prototype
  - www.prototypejs.org

Demos and Code

Learning & building from examples

UX Problem

Need to load data into page without reloading page

Solution

Use Ajax to load pieces of data when user makes request

People @ Your Library (Ajax)
www.lib.montana.edu/~jason/files/javascript/showpeople/

UX Problem

Want to use data from outside your site to enhance site content

Solution

Use a call to web service and process response client-side

Flickr API - Display Photos (Javascript)
www.lib.montana.edu/~jason/files/javascript/flickr/

UX Problem

You have a long list of information and you want to present pieces of list to a user only when requested

Solution

Use jQuery to hide/display information when clicked

Dynamic Definition List
(toggle with jQuery)
www.lib.montana.edu/~jason/files/javascript/togglelist/

UX Problem

Have sets of data that users have to scroll and scroll to see

Solution

Use jQuery to create a search/browse filter interface

Livesearch of a Department and Subject List (jQuery)
www.lib.montana.edu/~jason/files/javascript/livesearch/

UX Problem

Have essential user interface elements to include, but can't clutter interface

Solution

Show and hide search interface based on user's request (click)

Toggle Search Form (Javascript)
www.lib.montana.edu/~jason/files/javascript/togglesearch/

Final thoughts - What's Next?

- Consider Javascript advantages and disadvantages
- Fundamentals of method are there
- Keep practicing and learning

Questions?

twitter.com/jaclark
www.lib.montana.edu/~jason/talks.php