

A Brain-Friendly Guide

Head First Ajax

Fireside Chats



Learn how to make
your web pages
talk and listen at
the same time



Make your clunky
web apps feel like
dynamic, responsive
desktop applications



Learn how Sally did
two things at the same
time with asynchronous
programming



Transfer your
data with plain text,
XML, and JSON

Get a handle
on trees and
the Document
Object Model



O'REILLY®

Rebecca M. Riordan

Table of Contents

Chapter 1. using ajax..... 1

Section 1.1. Web pages: the old-fashioned approach..... 2

Section 1.2. Web pages reinvented..... 3

Section 1.3. So what makes a page "Ajax"?..... 5

Section 1.4. Rob's Rock 'n' Roll Memorabilia..... 6

Section 1.5. Ajax and rock 'n' roll in 5 steps..... 12

Section 1.6. Step 1: Modify the XHTML..... 14

Section 1.7. Step 2: Initialize the JavaScript..... 16

Section 1.8. Step 3: Create a request object..... 20

Section 1.9. Step 4: Get the item's details..... 22

Section 1.10. Let's write the code for requesting an item's details..... 24

Section 1.11. Always make sure you have a request object before working with it..... 25

Section 1.12. The request object is just an object..... 26

Section 1.13. Hey, server... will you call me back at displayDetails(), please?..... 27

Section 1.14. Use send() to send your request..... 28

Section 1.15. The server usually returns data to Ajax requests..... 30

Section 1.16. Ajax is server-agnostic..... 31

Section 1.17. Use a callback function to work with data the server returns..... 35

Section 1.18. Get the server's response from the request object's responseText property..... 36

Section 1.19. Goodbye traditional web apps..... 38

Section 1.20. AjaxAcrostic..... 39

Table of Contents

Chapter 2. designing ajax applications..... 1

Section 2.1. Mike's traditional web site sucks..... 2

Section 2.2. Let's use Ajax to send registration requests ASYNCHRONOUSLY..... 4

Section 2.3. Update the registration page..... 9

Section 2.4. Event Handlers Exposed..... 11

Section 2.5. Set the window.onload event handler... PROGRAMMATICALLY..... 12

Section 2.6. Code in your JavaScript outside of functions runs when the script is read..... 14

Section 2.7. What happens when..... 15

Section 2.8. And on the server..... 16

Section 2.9. Some parts of your Ajax designs will be the same... every time..... 18

Section 2.10. createRequest() is always the same..... 19

Section 2.11. Create a request object... on multiple browsers..... 22

Section 2.12. Ajax app design involves both the web page AND the server-side program..... 24

Section 2.13. The request object connects your code to the web browser..... 30

Section 2.14. You talk to the browser, not the server..... 31

Section 2.15. The browser calls back your function with the server's response..... 34

Section 2.16. Show the Ajax registration page to Mike..... 36

Section 2.17. The web form has TWO ways to send requests to the server now..... 37

Section 2.18. Let's create CSS classes for each state of the processing..... 40

Section 2.19. ...and change the CSS class with our JavaScript..... 41

Section 2.20. Changes? We don't need no stinkin' changes!..... 42

Section 2.21. Only allow registration when it's appropriate..... 43

Table of Contents

Chapter 3. javascript events..... 1

Section 3.1. It all started with a downward-facing dog..... 2

Section 3.2. Ajax apps are more than the sum of their parts..... 9

Section 3.3. Here's Marcy's XHTML..... 10

Section 3.4. Events are the key to interactivity..... 12

Section 3.5. Connect events on your web page to event handlers in your JavaScript..... 15

Section 3.6. Use the window.onload event to initialize the rest of the interactivity on a web page..... 16

Section 3.7. Change those left-side images to be clickable..... 21

Section 3.8. Use your XHTML's content and structure..... 22

Section 3.9. Add the code for hideHint(), too..... 25

Section 3.10. Tabs: an optical (and graphical) illusion..... 26

Section 3.11. Use a for... loop to cycle through the images..... 27

Section 3.12. CSS classes are the key (again)..... 28

Section 3.13. Ummm... but the tabs aren't <a>'s!..... 29

Section 3.14. This broke our JavaScript, too, didn't it?..... 30

Section 3.15. Use a request object to fetch the class details from the server..... 35

Section 3.16. Be careful when you have two functions changing the same part of a web page..... 36

Section 3.17. When you need to change images in your script, think "change CSS classes" instead..... 41

Section 3.18. Links in XHTML are represented by <a> elements..... 42

Section 3.19. We need a function to show an active button and hide a button, too..... 43

Table of Contents

Chapter 4. multiple event handlers..... 1

Section 4.1. An event can have only one event handler attached to it (or so it seems)..... 2

Section 4.2. Event handlers are just properties..... 3

Section 4.3. A property can have only ONE value..... 3

Section 4.4. Assign multiple event handlers with addEventListener()..... 4

Section 4.5. Your objects can have multiple event handlers assigned to a single event in DOM Level 2..... 6

Section 4.6. What's going on with Internet Explorer?..... 10

Section 4.7. Internet Explorer uses a totally different event model..... 11

Section 4.8. attachEvent() and addEventListener() are functionally equivalent..... 11

Section 4.9. addEventHandler() works for ALL apps, not just Marcy's yoga page..... 16

Section 4.10. Let's update initPage() to use our new utility function..... 17

Section 4.11. Use an alert() to troubleshoot..... 19

Section 4.12. So what else could be going wrong?..... 19

Section 4.13. Event handlers in IE are owned by IE's event framework, NOT the active page object..... 21

Section 4.14. attachEvent() and addEventListener() supply another argument to our handlers..... 22

Section 4.15. We need to name the Event argument, so our handlers can work with it..... 23

Section 4.16. You say target tomato, I say srcElement tomato..... 24

Section 4.17. So how do we actually GET the object that triggered the event?..... 28