

Get Vs. Post

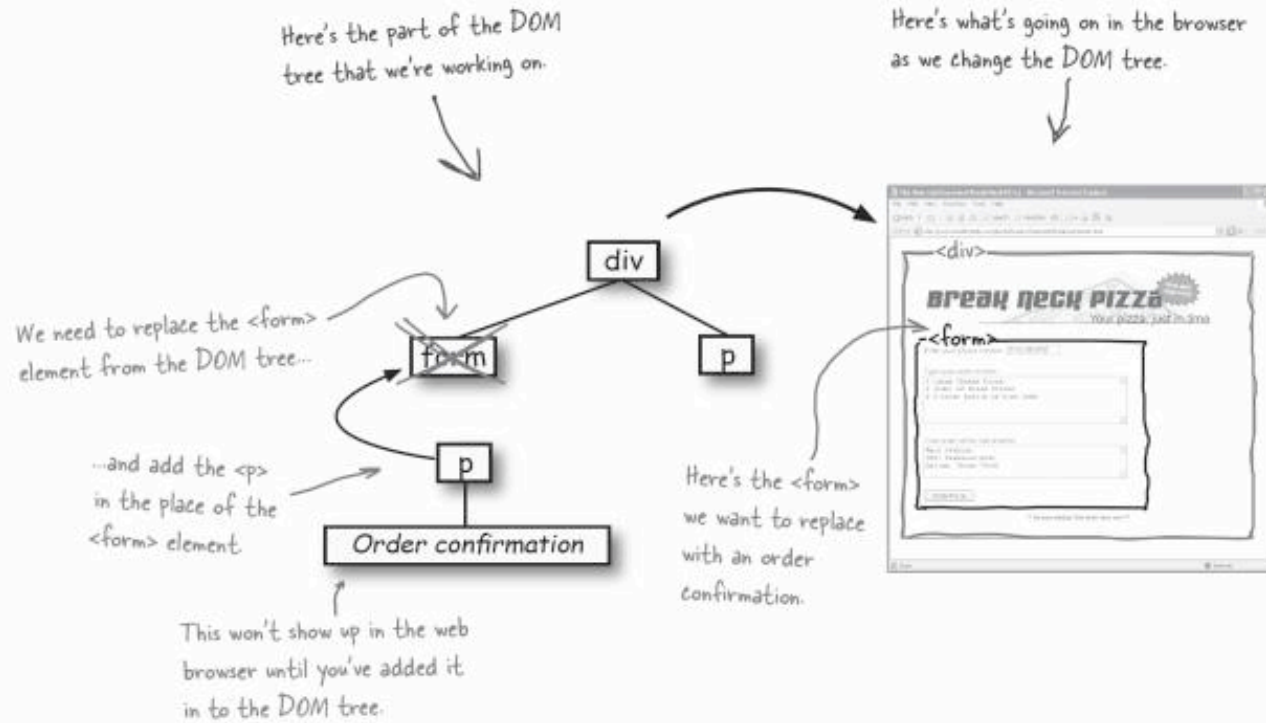
A Question of Volume and Semantics

Blog Review

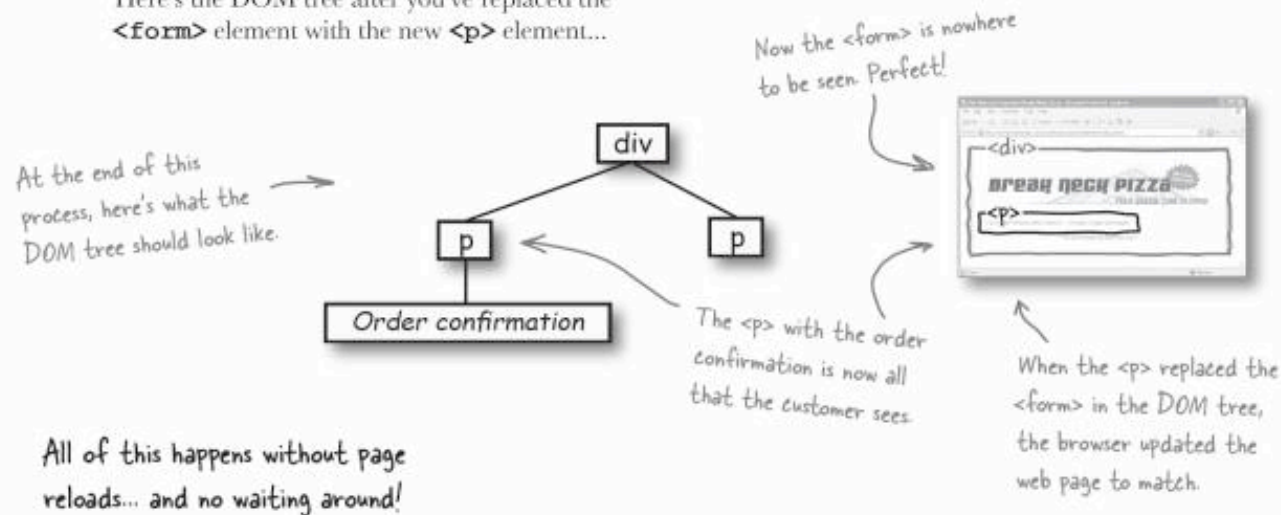
- Tag cloud issues
- Google maps

The DOM is connected to what a customer sees

Let's take a closer look at exactly what happens when you replace an element in the DOM:

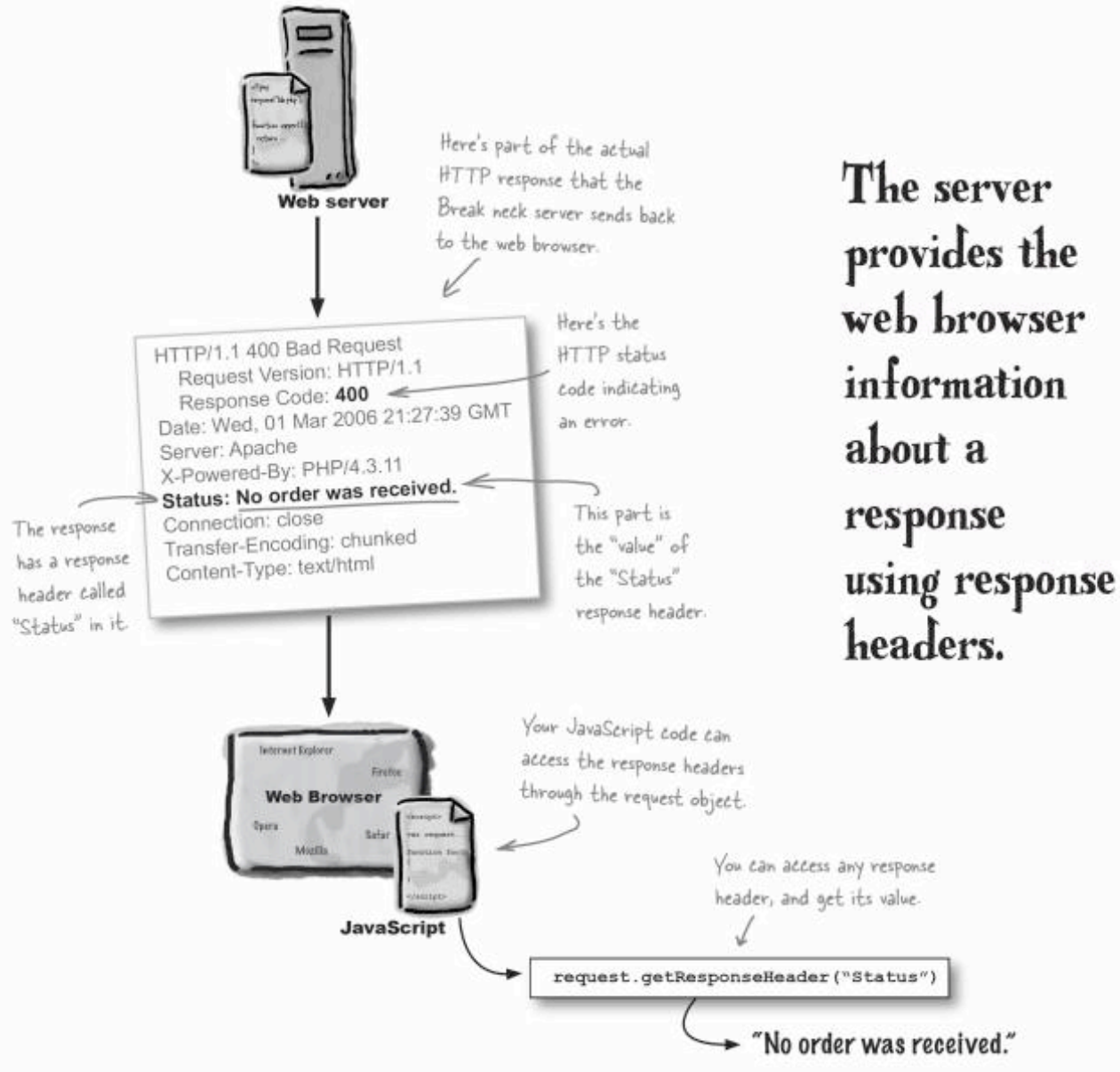


Here's the DOM tree after you've replaced the `<form>` element with the new `<p>` element...



The server talks back

Anytime the server sends a response to your request, it can give you information about its response using **response headers**.



Bud's Observations

- The ajax method strips away the abstraction of web application frameworks
- You are working with open standards
 - Over a decade old (mature)
 - Widely supported
- Have to think about details that frameworks keep from you.

Post Vs. Get

- Post more complicated to set up
 - Requires a request header to tell the server how to interpret the data.
 - Avoids data caching.
- Post implies you are making a change to server data
- Get implies you are getting, not changing

SQL Injection

- Just read pages 317–333
- The problem is that what you put in the form fields can be used as query parameters.
- Reveal the whole contents of your database.
 - Generally bad.

Summary

- This chapter completes the basics
- Next, XML and JSON