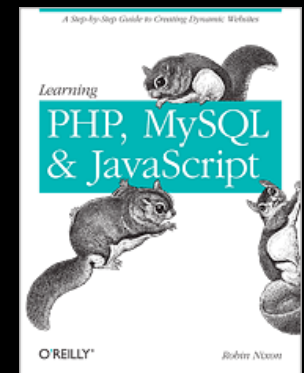


Using JQuery / JSON

Chapter 16

Dr. Charles Severance

To be used in association with the book:
PHP, MySQL, and JavaScript by Robin Nixon



open.michigan

Unless otherwise noted, the content of this course material is licensed under a Creative Commons Attribution 3.0 License.

<http://creativecommons.org/licenses/by/3.0/>.

Copyright 2011, Charles Severance



In Transition...

- In the beginning: HTML 3
- Current technology: HTML 4 + JavaScript + CSS
 - Peeking ahead: JQuery
- Coming soon: HTML 5

Document Object Model

- JavaScript can manipulate the current HTML document
- The “Document Object Model” tells us the syntax to use to access various “bits” of the current screen to read and/or manipulate
- You can even find pieces of the model by **id** attribute and change them
- We can read and write the DOM from JavaScript

http://en.wikipedia.org/wiki/Document_Object_Model

DOM's are not Identical

- Not all browsers represent their page exactly the same.
- This makes it a challenge to keep all of your JavaScript working on all browsers
- Also means you need to test your code over and over on each browser
- Aargh..

JQuery to the rescue

- While the DOM's are not particularly portable, and direct DOM manipulation is a little clunky, there are a number of JavaScript frameworks that handle the myriad of subtle differences between browsers
- <http://jquery.org/>
- With JQuery, instead of manipulating the DOM, we use JQuery functions and everything works much better..

Sites

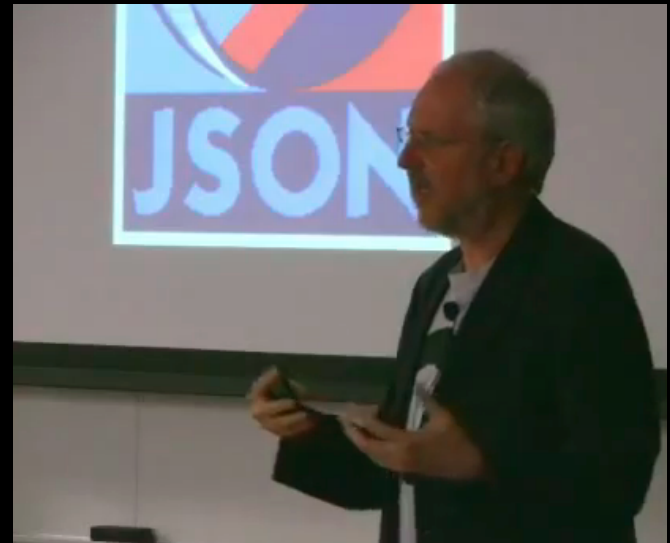
- The web is a wonderful source of JQuery documentation
 - http://docs.jquery.com/Main_Page
 - <http://api.jquery.com/>
 - <http://jqueryui.com/demos/>
 -

DEMO

JavaScript Object Notation

JavaScript Object Notation


- Douglas Crockford - "Discovered" JSON
- Object literal notation in JavaScript



<http://www.youtube.com/watch?v=-C-JoyNuQJs>

JSON

http://www.json.org/ Google



Introducing JSON

العربية Български 中文 Český Niederlandse Dansk English Esperanto Française Deutsch Ελληνικά עברית Magyar Indonesia Italiano 日本 한국어 فارسی Polski Português Română Русский Српски Slovenščina Español Svenska Türkçe Tiếng Việt

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the [JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999](#). JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an *object*, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an *array*, vector, list, or sequence.

These are universal data structures. Virtually all modern programming languages support them in one form or another. It makes sense that a data format that is interchangeable with programming languages also be based on these structures.

In JSON, they take on these forms:

An *object* is an unordered set of name/value pairs. An object begins with { (left brace) and ends with } (right

```

object
  {}
  { members }
members
  pair
  pair , members
pair
  string : value
array
  []
  [ elements ]
elements
  value
  value , elements
value
  string
  number
  object
  
```

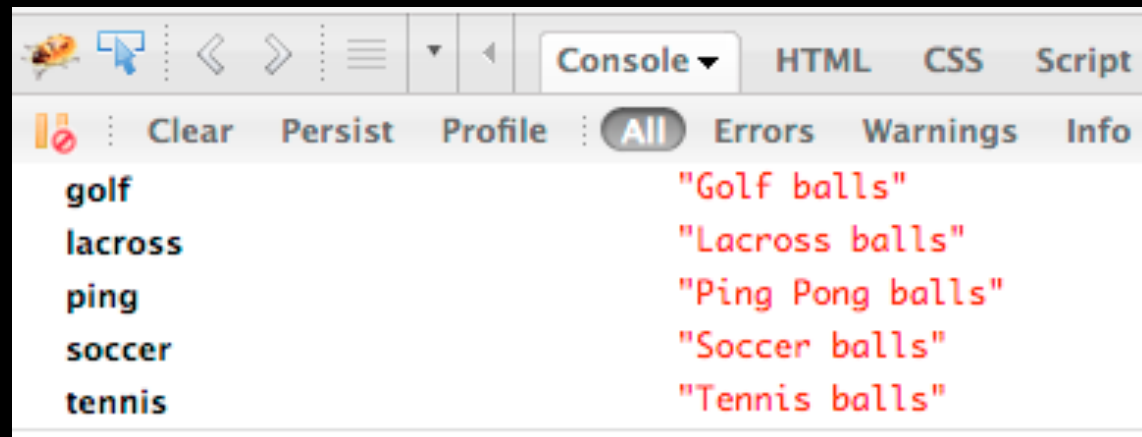
Associative ~~Arrays~~ Objects

- JavaScript Associative Arrays are actually objects with member variables
- They can be accessed with either associative array syntax or object syntax

```
balls = {"golf": "Golf balls",  
        "tennis": "Tennis balls",  
        "ping": "Ping Pong balls"};
```

```
balls.soccer = "Soccer balls";  
balls['lacross'] = "Lacross balls";
```

```
console.dir(balls);
```



JSON Syntax

```
who = {  
  'name': 'Chuck',  
  'age': 29,  
  'college': true,  
  'offices' : [ '3350DMC', '3437NQ' ],  
  'skills' : { 'fortran': 10,  
               'C': 10,  
               'C++': 5,  
               'python' : '7'  
            }  
};
```

String

Integer

Boolean

List/Array

Object

DEMO