CSE 154

LECTURE 24: JSON

JSON data

```
{
  "private": "true",
  "from": "Alice Smith (alice@example.com)",
  "to": [
    "Robert Jones (roberto@example.com)",
    "Charles Dodd (cdodd@example.com)"
  ],
  "subject": "Tomorrow's \"Birthday Bash\" event!",
  "message": {
    "language": "english",
    "text": "Hey guys, don't forget to call me this weekend!"
  }
}
```

JSON

Browser JSON methods

method	description
JSON.parse(*string*)	converts the given string of JSON data into an equivalent JavaScript object and returns it
JSON.stringify(*object*)	converts the given object into a string of JSON data (the opposite of JSON.parse)

- you can use Ajax to fetch data that is in JSON format
- then call `JSON.parse` on it to convert it into an object
- then interact with that object as you would with any other JavaScript object

JSON example: Books

Suppose we have a service books_json.php about library books.

- If no query parameters are passed, it outputs a list of book categories:

```
{ "categories": ["computers", "cooking", "finance", ...] }
```
JSON

- Supply a category query parameter to see all books in one category:
  http://webster.cs.washington.edu/services/books/books_json.php?category=cooking

```
{
  "books": [
    {"category": "cooking", "year": 2009, "price": 22.00,
     "title": "Breakfast for Dinner", "author": "Amanda Camp"},
    {"category": "cooking", "year": 2010, "price": 75.00,
     "title": "21 Burgers for the 21st Century", "author": "Stuart Reges"},
    ...
  ]
}
```
JSON

JSON exercise

Write a page that processes this JSON book data.

- Initially the page lets the user choose a category, created from the JSON data.
  - ○ Children ○ Computers ○ Finance List Books

- After choosing a category, the list of books in it appears:

Books in category "Cooking":

- Breakfast for Dinner, by Amanda Camp (2009)
- 21 Burgers for the 21st Century, by Stuart Reges (2010)
- The Four Food Groups of Chocolate, by Victoria Kirst (2005)

Bad style: the eval function

```
// var data = JSON.parse(this.responseText);
var data = eval(this.responseText);   // don't do this!
...
```
JS

- JavaScript includes an `eval` keyword that takes a string and runs it as code
- this is essentially the same as what `JSON.parse` does,
- but `JSON.parse` filters out potentially dangerous code; `eval` doesn't
- `eval` is evil and should not be used!

What is a web service?

web service: software functionality that can be invoked through the internet using common protocols

- like a remote function(s) you can call by contacting a program on a web server
  - many web services accept parameters and produce results
- can be written in PHP and contacted by the browser in HTML and/or Ajax code
- service's output might be HTML but could be text, XML, JSON or other content
  - examples seen in CSE 154: `quote.php`, `animalgame.php`, `books_json.php`, `urban.php`, `weather.php`

Setting content type with header

```
header("Content-type: type/subtype");
```
PHP

```
header("Content-type: text/plain");
print "This output will appear as plain text now!\n";
```
PHP

- by default, a PHP file's output is assumed to be HTML (text/html)
- use the header function to specify non-HTML output
  - must appear before any other output generated by the script

Recall: Content ("MIME") types

MIME type	related file extension
text/plain	.txt
text/html	.html, .htm, ...
text/xml	.xml
application/json	.json
text/css	.css
text/javascript	.js
image/gif	.gif

- Lists of MIME types: by type, by extension

Example: Exponent web service

Write a web service that accepts a `base` and `exponent` and outputs `base` raised to the `exponent` power. For example, the following query should output `81` :

```
http://example.com/exponent.php?base=3&exponent=4
```

solution:

```php
<?php
header("Content-type: text/plain");
$base = (int) $_GET["base"];
$exp  = (int) $_GET["exponent"];
$result = pow($base, $exp);
print $result;
?>
```