

University of Washington, CSE 190 M, Spring 2007

Lab 10: Final Exam Review (Thursday, May 31th, 2007)

1. Ajax/XML

Write the Ajax Javascript code to fetch and display XML data from the file named `q1.xml` (in the same directory as your code). This file contains data about a rectangle to draw on the page, including its size and color. Your code should process the XML and display the rectangle on the page as a `div`. Add the `div` to the bottom of the page body.

The XML data will be in a format that matches the following abbreviated example:

```
<?xml version="1.0" encoding="UTF-8"?>
<shapes>
  <rectangle width="100" height="30" color="FFFF00" />
</shapes>
```

For the XML data above, your code would produce the following content on the HTML page:



You may assume that your page already contains the following code from lecture and the slides:

```
function ajaxHelper(url, fn) { // calls fn when data arrives
  var ajax = new XMLHttpRequest();
  ajax.onreadystatechange = function() {
    if (ajax.readyState == 4 && ajax.status == 200)
      fn(ajax);
  };
  ajax.open("GET", url, true);
  ajax.send(null);
}
```

2. PHP

Write a PHP script that will look up a name in a company's records and display information about this employee. Your script will be given a full name (first and last) via a GET request parameter named `name`. Read the data from a file named `employees.txt` and attempt to match the given name. The file has one employee's data per line and each piece of information (full name, username, position) is separated by a tab. The `employees.txt` file will look like this:

```
Marla Jeffries      mjeff      Lamination Tzar
Conner O'Reilly    conn       Director of Archives
```

If the name is matched, you must display the employee's name (with any spaces removed) and position. The position name must be a link as follows:

```
http://awesomeco.com/PositionNameWithoutSpaces/username
```

For example, accessing `yourScript.php?name=Marla Jeffries` should give the following output between the horizontal lines, where the text [Lamination Tzar](#) is a link to `http://awesomeco.com/LaminationTzar/mjeff`.

Marla Jeffries is the [Lamination Tzar](#).

If the name is not matched, you should display a message in the following format:

Tyler Durden is not an AwesomeCo employee.

3. SQL

Write an SQL query that will match up all actors who share the same last name and appeared in a movie together. Display the actors' first names, shared last name, and movie name. You should not match up an actor with him/herself, and you should show the results such that the person whose name comes first in ABC order is listed first. The following is a subset of the results returned:

```
+-----+-----+-----+-----+
| Carrie      | Christopher | Henn   | Aliens  |
| Matthew Michael | Taylor     | Goodall | Apollo 13 |
| Clint       | Rance      | Howard  | Apollo 13 |
| ...        |            |        |          |
| Tommy (VI)  | Zachary    | Lee     | Vanilla Sky |
| Alice Marie | Cindy      | Crowe   | Vanilla Sky |
+-----+-----+-----+-----+
133 rows in set (0.11 sec)
```

Recall that the imdb database contains the following tables:

Actor				Movie			Cast		
id	fname	lname	gender	id	name	year	aid	mid	Role
433259	William	Shatner	M	112290	Fight Club	1999	433259	313398	James T. Kirk
797926	Britney	Spears	F	209658	Pi	2000	433259	407323	T.J. Hooker
831289	Jenny	Weaver	F	210511	Memento	2000	797926	342189	Herself
...					