cis20.2-spring2010-sklar, lab III.4
XML lab

This lab will give you an opportunity to get some hands-on experience with XML. First you will run and modify some examples. Then you will define your own XML format and try using it with a file you generate by hand. Finally you will automatically generate a file in your format by querying your MySQL database and grabbing data from your table(s) and storing it in the file.

You will also get a chance to use file handling in PHP.

A. Parsing XML

1. Log onto mingus (146.245.250.181).
2. Go to your `public_html` directory and copy the following 3 files from my directory into your directory:

```
sh$ cd public_html
sh$ cp /home/sklar/public_html/test.xml .
sh$ cp /home/sklar/public_html/xml-example.php .
sh$ cp /home/sklar/public_html/fac.xml .
```

3. In your web browser, go to the "PHP XML Expat Parser" page:
   `http://www.w3schools.com/php/php_xml_parser_expat.asp`
   NOTE that this is also linked on the class home page

   There you will find a TUTORIAL on using XML in PHP.
   You'll see that both the `test.xml` and `xml-example.php` were grabbed from that page.

   Try running your copy of `xml-example.php` in your browser and see that your output matches the output on the Tutorial web page.
4. Now MODIFY `xml-example.php` so that instead of opening `test.xml`, it opens `fac.xml`.
5. Then run `xml-example.php` again in your browser.
   What happens when you run it now? What does the output look like?
6. Modify `xml-example.php` so that the output has nice headings, like in the example copied from the Tutorial web page.

B. Designing XML

1. Pick one (or more) of the tables in your MySQL database that you created for your project. Design an XML format for storing that data (like we did in the last class). Create an XML file using your format and storing some of the data from your table.
2. Modify `xml-example.php` so that it displays your sample file.
3. Now compose a *Document Type Definition (DTD)* for your data, based on the XML file you just created. Put that at the beginning of your data file and make sure `xml-example.php` still runs okay.
4. Try Validating your XML and DTD by going to:
   `http://www.w3schools.com/xml/xml_validator.asp` and pasting your code in the box labeled **Validate Your XML Against a DTD**.

C. Generating XML

1. Now try generating the content of the XML file automatically, by using PHP to read the MySQL database table and write an XML file that follows your format and contains the data you read from the database.

   You will need to use PHP File Handling, which you can read about on the following pages, if you haven't used it before:
   `http://www.w3schools.com/php/php_file.asp`
   `http://www.w3schools.com/php/php_ref_filesystem.asp`

   Here is an example:

   ```
   <?php
   if ( $file = fopen( "data/mytest.txt","w" )) {
     echo fwrite( $file,"good morning sunshine!" );
     fclose( $file );
   }
   else {
     die( "Could not open file" );
   }
   ?>
   ```

   Note that in order to write to a file from a web browser, the file you are writing to needs to be in a publicly writeable folder. The safest way to do this is to create a subfolder under your `public_html` folder and set the subfolder's permissions to 777.

   So, from your `public_html` directory, do this:

   ```
   sh$ mkdir data
   sh$ chmod 777 data
   ```

   Now you should be able to run the example (above) if you type it into a PHP file and address it in your browser.

2. Finally, try reading the XML file you created (above) with the modified version of `xml-example.php`.

NOTE that in a way this lab is a silly example, because it does not demonstrate a reason to use XML. Why not just display the data that you read from the MySQL database directly from the PHP file that reads the database, without using XML in the middle?

The answer is that you want to be able to grab the data from the database and transmit it to *another application* using a well-defined and easily-understood format: hence XML. The example is only silly because you wrote both the PHP file to read the database and generate the XML AND the PHP file that reads the XML and displays the data.

Next time, now that you fully grasp the basics of XML, we'll demonstrate a more useful example. :-)