CSE 190M Final Cheat Sheet Summer 2011

This is intended as a reference for the syntax and topics that you should be studying for the final, and it not intended as a complete reference or guide.

HTML

Tags You should know at a minimum…

div, span, p, img, a, ul, ol, li, dl, dt,

dd, inputs, textarea, h1, h2..., button, q,

blockquote

IDs/Classes

Levels of Inheritance

!important > #id > .class > tag (though you will probably never see !important)

Tables

<table>

<tr>

<td>row 1, cell 1</td> <td>row 1, cell 2</td>

</tr>

<tr>

<td>row 2, cell 1</td> <td>row 2, cell 2</td>

</tr>

</table>

CSS

Stylings

Width

Height

Float

Clear

margin,

padding

border

background color

text align

position:relative

-relative to it's normal position

position:fixed

-relative to the browser window

position:absolute

-relative to first non static anscestor element

JavaScript

Prototype

*Accessing ID’s and CSS selectors*

var element = $(“idname”)

var elementValue = $(“idname”).value;

element.setAttribute(“attributename”,value);

element.addClassName(“classname”); //same thing for removeClassName

var elementArray = $$(“css selector”);

var firstElement = elementArray[0];

*Ajax Requests*

new Ajax.Request("webservicename",

{

method: "get", //or "post"

parameters: {"parameter1":value1,"paramtetr2";value2},

onSuccess: success,

onFailure: ajaxFailure,

onException: ajaxFailure

}

);

*Dealing with Ajax responses*

function success(ajax) {

var x = ajax.responseXML.getElementsByTagName(“tagnamehere”);

//returns array of tags

var attribute = x[i].getAttribute("attributename");

//Or

var y = ajax.responseText.split(“\n”);

//returns array of lines

}

*Checkboxes and Stuff*

var myBoolean = $(“checkbox”).checked;

*Modifyingthe DOM*

var newElement = document.createElement(“tagtype”); //makes a new tag

outerElement.appendChild(newElement); //add to the dom

*Random*

for(var i = 0; i < x.length; i++) {

var p = document.createElement(“p”);

p.setAttribute(“class”, “special”);

p.innerHTML = x[i];

$(“output).appendChild(p);

}

PHP

Form Processing

//Use the $\_REQUEST array

//Reference your HW4

Header

*Errors*

*//*At the top

if(!isSet($\_REQUEST[“field”]) {

//throw errors

}

header(“HTTP/1.1 400 Invalid Request”);

*Type*

*//*(for your php to be a web service)

header("Content-type: text/plain");

header("Content-type: text/xml");

Glob

$file = file(“myfile.txt”); //returns array of llines

$file2 = file\_get\_contents(“myfile.txt”); //returns string with \n characters

$path = “documents/”

$files = glob($path . "\*.txt"); //grabs all the files in documents that end with .txt

$files2 = glob($path . “\*cat\*”); //grabs all the files with cat in them  
 $files2 = glob($path . “cat\*”); //grabs all the files starting with cat

foreach ($files as $file) {

?>

<p><?=$file?></p>

<?php

}

?>

MySQLGuide

//Here is what a regular SQL query will look like…

**SELECT** what cols to return

**FROM** what table

**WHERE** what condition 1 on the rows **AND** condition 2 on rows;

//Remember….ALL RESERVED WORDS IN SQL ARE IN CAPS!

//Other things you can do.

**ORDER BY** col name **DESC**;

**LIMIT** how many results you want to see;

//returns rows ordered by the value of the row with the highest first, also ASC

**JOIN** another table **ON** what conditions;

//Sometimes there is information on more than one table that you need to access for a single query.

//this is more complicated, but **JOIN** refers to what table you want to merge with your own and **ON** //matches rows of the different tables together based on what conditions you specified. Multiple joins //are common, and will be your final. Practice, practice, practice!!

//joinscreates a Cartesian coordinate of the two tables, and in some cases can make huge combined tables that might return more results than you originally wanted. That is why when using tables you will often find

**SELECT DISTINCT**

//is the better call to use, because you might have 15 of the same value returned from one column, when one will do.

//The best way to practice SQL is by doing queries yourself. Look at the slides from the week 8 lab, and that will tell you how to get set up. I suggest using “putty” as your SSH client. Once you get on, a few hints I have.

My Hints

//1. Pasting into putty is done by right clicking the indicator box. Use this to avoid having a war with your password on Thursday or Friday.

//2. You can press up and down to cycle through your past queries. This makes it quicker to send a lot of queries and so you don’t have to type the whole thing multiple times.

//3. Prepare your queries in a text editor. They look way better for you and are easier to edit than in the // terminal.

//4. For sql requests in PHP

$db = mysql\_connect("localhost", "name", "password");

mysql\_select\_db("database name");

$queryResults = mysql\_query("your query with variables like '$this' ;");

$nameRow = mysql\_fetch\_array($queryResults);

$value = $nameRow["some col of table"];