This is Week 8

Halloween

Jason Hirschhorn
Fall, 2011
Hi!

Welcome to my home page!

Feel free to browse around!

This is a picture of two dolphins. Dolphins are my favorite animals.

(Btw, that word "dolphin" up there. Ya, it's a hyperlink. Nbd.)

Wait a second . . . here's one more link. Baby monkey!
Agenda

- Announcements
- HTML
  - Tags
  - mypage.html
  - Resources
- chmod
- CSS
  - mypage1.html
  - mypage2.html + mycss.css
  - mypage3.html
- PHP
- SQL
Announcements

• Problem Set 7 Walkthrough (Sun, 7pm, NW B103) – [https://www.cs50.net/psets/](https://www.cs50.net/psets/)
• Problem Set 5’s Scavenger Hunt
  • :(  
• Problem Set 6’s BIG BOARD
  • I Saw You... Adabelle, Irineo, Holly, Stephen, Jimmy, Larmon, Marc + Quan
• Final Project
  • Pre-Proposal due by noon on Mon 11/7
  • Looking for a project? [https://projects.cs50.net](https://projects.cs50.net)
  • Want to learn something new? [https://manual.cs50.net/Seminars](https://manual.cs50.net/Seminars)
HTML
HTML

- HyperText Markup Language
- NOT a programming language
  - Describes content but does not create content
  - No if conditions or for/while loops

Main Purposes
- Content, structure and format
  - How should my content look?
  - Where should it all go?
- Metadata
  - Information about the web page
• A series of embedded elements
  • I.e. “things”
• But “thing” isn’t that descriptive...
• ...so each element has tags that tell the browser exactly what to do with it
• In human-speak

  start tag: I'm as brave as a lion!
  make things bold

  end tag: stop making things bold

• In HTML

  <b> I'm as brave as a lion! </b>
Tags

• Many tags contain attributes, which modify or give specificity to a tag
  • E.g. I have a hyperlink but to where?
• Attributes are specified in key-value pairs
  
  key   value
  
  <a href="https://www.cs50.net">CS50 Home</a>

• Remember to always close your tags!
A valid HTML document also has a particular structure:

- `<!DOCTYPE html>`
  - Tells the browser what type of document this file is
  - Simply a declaration at the beginning of the file
- `<html>`
  - Tells the browser that everything inside is HTML
- `<head>`
  - Tells the browser what to load before the user can interact with the page
- `<body>`
  - Tells the browser that everything inside is the actual content of the page
HTML

• To view your web page from a browser, it must be in a directory called public_html
  • You’ll create one in pset7
• Remember to make sure your HTML is valid!
  • http://validator.w3.org/
• Feel free to check out my home page...
  • https://cloud.cs50.net/~jhirschh/
Resources

• There’s so much I can/want to do! Where do I look for help?
• Google
  • https://www.google.com/
• w3schools
  • http://www.w3schools.com/
  • “Try it Yourself Editor”
• cs50.net Resources
  • https://www.cs50.net/resources/
• Week 8 Cheatsheets
  • https://www.cs50.net/lectures/
chmod
chmod

• “Change mode”
  • Control the permissions for files and directories

• Who can access?
  • You (i.e. user)
  • Everyone else (i.e. group + others)

• How much can someone access?
  • r – read a file or list a directory’s contents
  • w – write to a file or directory
  • x – execute a file or move into a directory

• Why are permissions important?
chmod

- Make a directory
  
  ```
  mkdir ~:/public_html
  ```

- Change the directory’s permissions
  
  ```
  chmod a+x ~:/public_html
  ```

- Alternatively
  
  ```
  chmod 711 ~:/public_html
  ```

<table>
<thead>
<tr>
<th>user</th>
<th>group</th>
<th>others</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>w</td>
<td>x</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
chmod

- For index.html (or any HTML page we create)
  - The user should be able to read and write
  - The rest of the world should be able to read

Quick Quiz

```
chmod ??? index.html
```

- What are the questions marks if...
  - ...using “shorthand”?
  - ...using an octal combination?
CSS
CSS

• Cascading Style Sheets
  • HTML defines our content and structure
  • CSS defines its style
• Give certain style properties to a specific element or group of elements
  • For a specific element, give it an id
  • For a group of elements, give them all a class

<img src="logo.gif" id="logo" class="top">

<h1 class="top">My Page Title</h1>
Styling a Page

1. Use the style attribute
   ```html
   <p style="text-align: center;">
   ```

2. Create an external stylesheet and link it in
   • In the head of the HTML file
     ```html
     <link rel="stylesheet" type="text/css" href="style.css">
     ```
   • In the CSS file
     ```css
     p { text-align: center; }
     ```

3. Use an internal stylesheet
   • In the head of the HTML file
     ```html
     <style>
     p { text-align: center; }
     </style>
     ```
PHP
PHP

- **PHP Hypertext Preprocessor**
- Dynamically generates a web page
  - A “backend” for your web page
  - Code is executed by the server not the browser
  - Plays nicely with HTML
- All PHP code (if in an HTML file) must be inside the `<?php ?>` tag
- It’s really similar to C...
  - `if (condition) {} else if (condition) {} else {}`
  - `while (condition) {}`
  - `for (<pre>; <condition>; <post>) {}`
PHP

- ...for the most part

Variables
- Don’t have types, and you can mix data types
  - Loosely-typed (PHP) vs. strongly-typed (C)
- Must start with the dollar sign
  - $x = 42;
  - $s = "cs50";

Compilation
- All PHP code on a page is interpreted line by line by the server then the result is displayed to the user
  - Interpreted (PHP) vs. compiled (C)
• A browser “requests” a web page in two ways

GET
• Information is displayed in the URL
• For information that is not too sensitive
• E.g. a page on YouTube
  http://www.youtube.com/watch?v=GI6CfKcMhjY

POST
• Information is not displayed in the URL
• Marginally more private
• E.g. whether you are/are not logged into cs50.net, the URL is the same
  https://www.cs50.net/
PHP

• Whether it’s a GET or POST request, PHP can access the information
  • $_GET
  • $_POST

• Built-in arrays that store key-value pairs
  • (Like hash tables from last week)

• E.g.
  • If a user accesses this URL
    
    http://www.youtube.com/watch?v=GI6CfKcMhjY
  • In watch.php referencing $_GET("v") returns GI6CfKcMhjY
  • So we could look up $_GET("v") in the database where we store all the videos and then display the right video
PHP

• If a user submits a form, variables are stored as key-value pairs in arrays

   where variables should be sent

```html
<form action="register.php" method="post">
    <table>
        <tr>
            <td>Captain:</td>
            <td><input name="captain" type="checkbox"></td>
        </tr>
    </table>
    <input type="submit" value="Register!">
</form>
```

what type of request (get or post)

name attribute is the key; user input is value
SQL

- **Structure Query Language**
- Allows you to interact with a database
  - Insert new values
  - Retrieve values meeting certain parameters
  - Delete or update old values
- Tightly integrated with PHP
Database

- Database = collection of tables
- Table = many records of the same type
- Record = single instance of a type that fills a number of fields

<table>
<thead>
<tr>
<th>fields</th>
<th>id</th>
<th>username</th>
<th>password</th>
</tr>
</thead>
<tbody>
<tr>
<td>records</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SQL Queries

• Insert a new row

\[
\text{INSERT INTO } \text{<table>}(\text{<column 1>, <column 2>})
\]
\[
\text{VALUES (}\text{<value1>, <value 2>})
\]

\[
\text{INSERT INTO users(id, username, password)}
\]
\[
\text{VALUES (1, 'djm', 'bowden')}
\]

• Delete a row

\[
\text{DELETE FROM } \text{<table> WHERE } \text{<column> = <value>}
\]

\[
\text{DELETE FROM users WHERE username = 'djm'}
\]
SQL Queries

• Retrieve a row

```sql
SELECT <column> FROM <table> WHERE <other column> = <value>
```

SELECT username FROM users WHERE id = 1

• Modify an existing row

```sql
UPDATE <table> SET <column> = <value> WHERE <other column> = <value>
```

UPDATE users SET password = 'chartier' WHERE username = 'djm'
• These queries are just examples
  • There are many more keywords you can add to your queries!
• In pset7, you get to create your own database
  • phpMyAdmin, a web-based tool with which you can manage MySQL databases
• As we’ve discussed, it is essential to “escape” the contents of any string given by the user before you put it in a database
HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.

OH, DEAR - DID HE BREAK SOMETHING?
IN A WAY-

DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?

OH, YES. LITTLE BOBBY TABLES, WE CALL HIM.

WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.

AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.
That was Week 8

http://xkcd.com/627/