

Discussion 9

Instructor: Professor Moser
TA: Yung-Ting Chuang
ECE155B



What is HTML?

- *H*yper *T*ext *M*arkup *L*anguage
- A markup language designed for the creation of web pages and other information viewable in a browser
- The basic language used to write web pages
- File extension: .htm, .html



Creating a HTML File

1. Open Notepad
2. Click on File -> Save as...
3. In the File name pull-down box, type in webpage.html
4. Click on Save
5. Type in content for your file
6. Once you finished the content, click on File -> Save



HTML Tags

- For example: `<b>`, `<font>`, `<title>`, `<p>` etc.
- Tag usually goes with pair: an open tag (`<b>`) and an end tag (`</b>`)

Effect	Code	Code Used	What It Does
Bold	B	<code>Bold</code>	Bold
Italic	I	<code><I>Italic</I></code>	<i>Italic</i>

- Single tag: `<hr>`, `<br>`
- Tags are NOT case sensitive



HTML Document Structure

```
<html>
```

```
<head>
```

```
<title> Page Title Goes Here </title>
```

```
</head>
```

```
<body>
```

```
content goes here
```

```
</body>
```

```
</html>
```



Background

- Bgcolor
- Specifies a background-color for a HTML page.

```
<body
```

```
bgcolor="#000000">
```

```
<body
```

```
bgcolor="rgb(0,0,0)">
```

```
<body bgcolor="black">
```

- Background
 - Specifies a background-image for a HTML page
- ```
<body
background="clouds.gif">

<body
background="http://www.
w3schools.com/clouds.gif
>
```

# Text

- Put text on a webpage
  - `<p>Today is my first day at my new job, I'm so excited!</p>`
  - Output: Today is my first day at my new job, I'm so excited!
- Put text in center of a page
  - `<center>Hello</center>`
  - Output: Hello
- Put text on the right of a page
  - `<p align="right">Hello</p>`
  - Output: Hello





# Font

- To change text size
  - `<font size="+3">Hello</font>`  
Tag attribute
  - Output: **Hello**
- To change text color
  - `<font color="red">Hello</font>`
  - Output: **Hello**
- Using both
  - `<font size="+3" color="red">Hello</font>`
  - Output: **Hello**

# Commonly Used Character Entities

Result	Description	Entity Name
	Non-breaking space	&nbsp;
<	Less than	&lt;
>	Greater than	&gt;
&	Ampersand	&amp;
“	Quotation mark	&quot;
©	Copyright	&copy;



# Headings

- There are 6 heading commands.

**<H1>This is Heading 1</H1>**

**<H2>This is Heading 2</H2>**

**<H3>This is Heading 3</H3>**

**<H4>This is Heading 4</H4>**

**<H5>This is Heading 5</H5>**

**<H6>This is Heading 6</H6>**



# List

- Unordered list

- Code:

- ```
<ul>
```

- ```
Coffee
```

- ```
<li>Milk</li>
```

- ```

```

- Output:

- Coffee

- Milk

- Ordered list

- Code:

- ```
<ol>
```

- ```
Coffee
```

- ```
<li>Milk</li>
```

- ```

```

- Output:

- 1. Coffee

- 2. Milk



# Table

```
<table border="1">
<tr>
<th>Heading</th>
<th>Another Heading</th>
</tr>
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td></td>
</tr>
</table>
```

Heading	Another Heading
Row 1, cell 1	Row 1, cell 2
Row 2, cell 1	

# Create Links

- A Hypertext link
  - `< a href="http://www.ucsb.edu">`  
UCSB Home`</a>`
  - Output: [UCSB Home](http://www.ucsb.edu)
- A Email link
  - `<a href="mailto:ece155b.s12@gmail.com">`  
Email me`</a>`
  - Output: [Email me](mailto:ece155b.s12@gmail.com)



# Image Formats

- .gif
  - Graphics Interchange Format
- .jpeg or .jpg
  - Joint Photographic Experts Group
- .bmp
  - bitmap



# Inserting Image

- Place all images in the same directory/folder where you web pages are
- `<img src>` is a single tag
- ``
  - Output: 
- Turn an image into a hyperlink
  - `<a href="http://www.ucsb.edu"></a>`
  - Output: 



# Image Size

- Computer images are made up of “pixels”
- `<IMG HEIGHT=“100” WIDTH=“150” SRC=“image.gif”>`



Height

Width

# Forms

- A form is an area that can contain form elements.
- `<form></form>`
- Commonly used form elements includes:
  - Text fields
  - Radio buttons
  - Checkboxes
  - Submit buttons



# Text Input Fields

- Used when you want the user to type letters, number, etc.

```
<form>
```

```
First name: <input
type="text"
name="firstname">
```

```


```

```
Last name: <input
type="text"
name="lastname">
</form>
```

- Output

First name:

Last name:



# Submission Button

- When user clicks on the “Submit” button, the content of the form is sent to another file.

```
<form name="input"
action="html_form_action
.asp" method="get">
Username: <input
type="text" name="user">

<input type="submit"
value="Submit">
</form>
```

- Output

Username:

Submit



# Checkboxes

- Used when you want the user to select one or more options of a limited number of choices.

`<form>`

```
<input type="checkbox" name="bike" value="bike"> I have a bike
```

`<br>`

```
<input type="checkbox" name="car" value="car"> I have a car </form>
```

- Output

- ☐ I have a bike
- ☐ I have a car



# Radio Buttons

- Used when you want the user to select one of a limited number of choices.

```
<form>
```

```
<input type="radio"
name="sex"
value="male"> Male
```

```


```

```
<input type="radio"
name="sex"
value="female"> Female
</form>
```

- Output

- Male
- Female



# Text Box

- Used when you want to get input from user.

```
<form>
```

```
<p>Please provide your
suggestion in the text box
below:</p>
```

```
<textarea row="10"
cols="30">
```

```
</textarea>
```

```
</form>
```

- Output

Please provide your  
suggestion in the text box  
below:

A large, empty rectangular text box with a thin white border, intended for user input. It is positioned below the text 'Please provide your suggestion in the text box below:'.

# Pull-down Menu

- Used when you want user to respond with one specific answer with choices you given.
- Output

<p>Select a fruit:</p>

<select name"Fruit">

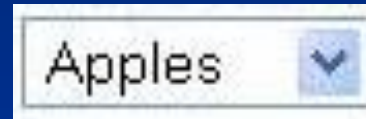
<option selected> Apples

<option> Bananas

< option > Oranges

</select>

Select a fruit:

A screenshot of a web browser showing a pull-down menu. The text 'Select a fruit:' is above the menu. The menu itself is a rectangular box with a light blue border. Inside, the word 'Apples' is written in a dark blue font. To the right of the text is a small blue square button with a white downward-pointing arrow.



# Form (Post vs. Get)

- POST is the preferred method for sending lengthy form data. When a form is submitted POST the user does not see the form data that was sent. For example, this form uses POST:

```
<FORM METHOD=POST ACTION="newDoc.jsp">
```



# Form (Post vs Get)

- GET sends the data as part of the URL.
- When the form data (or "query data") is added to the end of the URL it is "URL encoded" so that the data can be used in a standard URL.
- Example:
- [http://www.ucsb.edu/newDoc.jsp?newPass=puppies  
&newLogin=newPuppy](http://www.ucsb.edu/newDoc.jsp?newPass=puppies&newLogin=newPuppy)
- The neat thing about GET is that these unique URLs can be used directly in links without any form being involved.



# Form (Post vs. Get)

- If you use the "GET" method, the results of your form will appear in the URL box in the top of the browser and name value pairs, separated by "&".
- The POST method passes information back by another method which does not show on the URL.
- GET is good for debugging, but POST should probably be used in production.



# Form (post)

Where the page will direct to when user hit "submit" button

```
<form method="Post" action="NewDoctor.jsp">
 <table>
 <tr>
 <td>Name:</td>
 <td><input type="text" name="pname" ></td>
 </tr>
 <tr>
 <td>Last name:</td>
 <td><input type="text" name="pname" ></td>
 </tr>
 <tr>
 <td></td><td> <input type="submit" value="Submit"> </td>
 </tr>
 </table>
</form>
```

# Form (post)

- Below is how we retrieve the Name and LastName from the form using “POST” method in previous slide, in “newDoc.jsp” page:

```
String lastname = request.getParameter("pname");
String name = request.getParameter("pname");
```



# JSP Elements

- Directives: `<%@ dirname attr=value %>`
- Actions: `<pref:name attr=value ... />`
- Scripting:
  - Scriptlet: `<% scripting code %>`
  - Expression: `<%= expression %>`
  - Declaration: `<%! declaration %>`



# JSP Page Example

```
<%@ page contentType="text/html" %>
```

```
<%@ include file="header.jsp" %>
```

```
<h1>Hello World!</h1>
```

```
<jsp:useBean id="date" class="java.util.Date" />
```

```
It's <%= date %>
```

```
<%@ include file="footer.jsp" %>
```



# Directives Example

```
<%@ page contentType="text/html" %>
```

```
<%@ include file="header.jsp" %>
```

```
<h1>Hello World!</h1>
```

```
<jsp:useBean id="date"
 class="java.util.Date()"/>
```

```
It's <%= date %>
```

```
<%@ include file="footer.jsp" %>
```





# JavaBeans Class Example

```
public class UserInfoBean {
 private String firstName;
 public void setFirstName(String fName) {
 firstName = fName;
 }
 public String getFirstName() {
 return firstName;
 }
}
```



# useBean Example

```
<jsp:useBean id="user"
class="com.foo.UserInfoBean" />
```



# Scripting Elements

- Scriptlet: `<% scripting code %>`
- Expression: `<%= expression %>`
- Declaration: `<%! declaration %>`



# Scriptlet Example

```
<% if (user.isValid()) { %>
```

Do something

```
<% } else { %>
```

Do something else

```
<% } %>
```



# Expression Example

- First Name: `<%= user.getFirstName() %>`
- `1 + 1 = <%= 1 + 1 %>`



# Declaration Example

```
<%! int pageCount = 0; %>
```

```
<%! int requestCounter = 0; %>
```

```
<h1>Counters</h1>
```

Instance variable counter:

```
<%= pageCount++ %>
```

Local variable counter:

```
<%= requestCounter++ %>
```



# Session example

```
<html>
<head> <title> Sessions </title> </head>
<body bgcolor="#ddeedd">
<%! String name="" , pass="";
 HttpSession session; %>
<%
 name=request.getParameter("uname");
 pass=request.getParameter("pass");
 session.putValue("ID", pass);
 session.putValue("NAME", uname);
%>
</body>
</html>
```



# Session example (cont.)

```
<html>
```

```
<head><title> Retriving Session </title></head>
```

```
<body bgcolor="#34aabb">
```

```
<%
```

```
String pass= (String) session.getValue("ID");
```

```
String name= (String) session.getValue("NAME");
```

```
%>
```

```
</body>
```

```
</html>
```

