Web Site Design

Stanford University Continuing Studies CS 21

Mark Branom
branom@alumni.stanford.edu
http://web.stanford.edu/people/markb/

Course Web Site: http://web.stanford.edu/group/csp/cs03

Week 6 Agenda

- Unfinished business
- Lists
- Tables (how they work, practice using tables)
- Creative use of tables

Lists

There are two main types of lists in HTML:

- Unordered
  - Bulleted lists (disc, circle, square)
- Ordered
  - Numbered lists (1,2,3 or a,b,c or I,II,III, or i,ii,iii)

List tags

- `<ol> </ol>` = Ordered List
  - Attributes: "type=1/a/A/i/I"
- `<ul> </ul>` = Unordered List
  - Attributes: "type=disc/circle/square"

- `<li> </li>` = List Item (in an OL or UL)

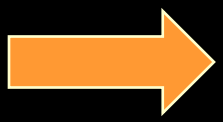
CSS styles for lists

- list-style-type:
  disc | circle | square | decimal |
  lower-roman | upper-roman | lower-alpha |
  upper-alpha | none
- list-style-position:
  inside | outside
- list-style-image:
  <url> | none

<UL> -- Unordered Lists

Note: type="disc" is the default attribute

```
<ul>
   <li> Glory bear </li>
   <li> Osito bear </li>
   <li> Brittania bear </li>
   <li> Maple bear </li>
   <li> Germania bear </li>
</ul>
```



- Glory bear
- Osito bear
- Brittania bear
- Maple bear
- Germania bear

<UL type ="circle">

```
<ul type="circle">
   <li> Glory bear </li>
   <li> Osito bear </li>
   <li> Brittania bear </li>
   <li> Maple bear </li>
   <li> Germania bear </li>
</ul>
```

- Glory bear
- Osito bear
- Brittania bear
- Maple bear
- Germania bear

<UL type="square">

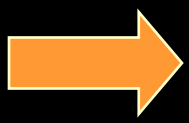
```
<ul type="square">
   <li> Glory bear </li>
   <li> Osito bear </li>
   <li> Brittania bear </li>
   <li> Maple bear </li>
   <li> Germania bear </li>
</ul>
```

- Glory bear
- Osito bear
- Brittania bear
- Maple bear
- Germania bear

<OL> = Ordered Lists

Note: type="1" is the default attribute

```
<ol>
  <li> Glory bear </li>
  <li> Osito bear </li>
  <li> Brittania bear </li>
  <li> Maple bear </li>
  <li> Germania bear </li>
</ol>
```



1. Glory bear
2. Osito bear
3. Brittania bear
4. Maple bear
5. Germania bear

<OL type="a">

```
<ol type="a">
  <li> Glory bear </li>
  <li> Osito bear </li>
  <li> Brittania bear </li>
  <li> Maple bear </li>
  <li> Germania bear </li>
</ol>
```

a. Glory bear
b. Osito bear
c. Brittania bear
d. Maple bear
e. Germania bear

<OL type="A">

```
<ol type="A">
  <li> Glory bear </li>
  <li> Osito bear </li>
  <li> Brittania bear </li>
  <li> Maple bear </li>
  <li> Germania bear </li>
</ol>
```

A. Glory bear
B. Osito bear
C. Brittania bear
D. Maple bear
E. Germania bear

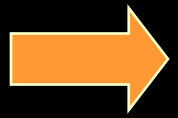
<OL type="i">

```
<ol type="i">
  <li> Glory bear </li>
  <li> Osito bear </li>
  <li> Brittania bear </li>
  <li> Maple bear </li>
  <li> Germania bear </li>
</ol>
```

i. Glory bear
ii. Osito bear
iii. Brittania bear
iv. Maple bear
v. Germania bear

<OL type="I">

```
<ol type="I">
  <li> Glory bear </li>
  <li> Osito bear </li>
  <li> Brittania bear </li>
  <li> Maple bear </li>
  <li> Germania bear </li>
</ol>
```



I. Glory bear
II. Osito bear
III. Brittania bear
IV. Maple bear
V. Germania bear

More on ordered lists

- In addition to modifying the method of counting, you can also modify the number to start the list with (start="xx" must be a numeric value, even if the type is an alphabetical value):

```
<ol type="1" start="4">
   <li> Glory </li>
   <li> Osito </li>
   <li> Brittania </li>
</ol>
```

4. Glory
5. Osito
6. Brittania

```
<ol type="A" start="4">
   <li> Glory </li>
   <li> Osito </li>
   <li> Brittania </li>
</ol>
```

D. Glory
E. Osito
F. Brittania

(note: NOT start="D")

Nesting a List

- You can nest a list to make organization charts, outlines, etc.

```
<ul>
  <li>Boss</li>
    <ul>
      <li>Supervisor A</li>
        <ul>
          <li>Worker A</li>
          <li>Worker B</li>
        </ul>
      <li>Supervisor B</li>
        <ul>
          <li>Worker C</li>
          <li>Worker D</li>
        </ul>
    </ul>
</ul>
```

- Boss
  - Supervisor A
    - Worker A
    - Worker B
  - Supervisor B
    - Worker C
    - Worker D

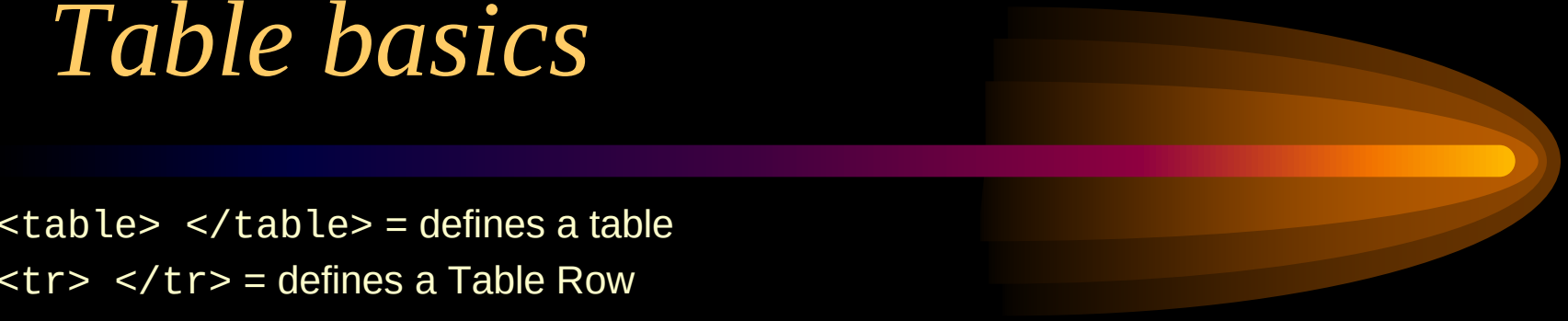
```
<ol type="I">
    <li>Intro</li>
    <li>Thesis 1</li>
      <ol type="A">
        <li>Point A</li>
        <li>Point B</li>
         <ol>
           <li>Subpoint 1</li>
            <ol type="a">
              <li>Proof a</li>
              <li>Proof b</li>
            </ol>
           <li>Subpoint 2</li>
         </ol>
      </ol>
    <li>Thesis 2</li>
    <li>Thesis 3</li>
    <li>Conclusion</li>
</ol>
```

I. Intro
II. Thesis 1
    A. Point A
    B. Point B
        1. Subpoint 1
            a. Proof a
            b. Proof b
        2. Subpoint 2
III. Thesis 2
IV. Thesis 3
V. Conclusion

Tables

- Tables allow web designers to place data into rows and columns of cells.
- Organizing Content
  - Tables are most often used to organize data on a web page. As with spreadsheets, tables allow you to arrange data into rows and columns of cells. Any sort of data can be placed in the table -- even other tables.
- Page Layout
  - Creative web page designers often use tables for page layout. By hiding the border of the table, the cells can hold elements of the page and place them more precisely on the page. This method allows for more creativity and control by the author.
  - The World Wide Web Consortium (W3C) is an organization dedicated to providing guidance and structure to the Internet. In 2002, the W3C recommended that web designers stop using tables to control page layout, and to instead use Cascading Style Sheets (CSS). As web browsers became more compliant with CSS, the need for using tables as a page layout tool fast became an obsolete method. That said, email clients (Outlook, Apple Mail, etc.) do not understand CSS well. If you want to create an HTML email newsletter, you must still use tables to control page layout.

Table basics



`<table> </table>` = defines a table
`<tr> </tr>` = defines a Table Row
`<td> </td>` = defines a Table Cell ("Table Data")
`<th> </th>` = defines a Table Header (special kind of Table Cell)
`<caption> </caption>` = defines a caption above the table

Attributes:

- `summary` - text description of the contents of the table (modifies `<table>`)
- `bgcolor` / `background` - specifies color/background of the table (modifies `<table>`,`<td>`,`<tr>`)
- `border` - specifies border width of table in pixels or % (modifies `<table>`)
- `height` - specifies height of the table or cell in pixels or % (modifies `<table>` and `<td>`)
- `width` - specifies width of the table or cell in pixels or % (modifies `<table>` and `<td>`)
- `cellpadding` - specifies distance between cell and content within cell(modifies `<table>`)
- `cellspacing` - specifies spacing between each cell (modifies `<table>`)
- `align` - specifies horizontal alignment (`center/right/left` - modifies `<caption>`,`<table>`, `<td>`)
- `valign` - specifies vertical alignment (`top/middle/bottom` - modifies `<caption>`,`<tr>`,`<td>`)

Table example

```
<table>
 <caption>
     Beanie Babies!
 </caption>
 <tr>
    <th>Bear Name</th>
    <th>Description</th>
 </tr>
 <tr>
    <td>Glory</td>
    <td>American Bear</td>
 </tr>
 <tr>
    <td>Osito</td>
    <td>Mexican Bear</td>
 </tr>
</table>
```

Beanie Babies!

Bear Name	Description
Glory	American Bear
Osito	Mexican Bear

Creative Use of Tables

- In addition to allowing data to be placed onto a web site, tables allow designers the ability to control page layout.

- You can use tables to wrap text around graphics, to put text in rows and columns (like a newspaper), etc…

Using Tables for Page Layout (HTML Email only)

1) Decide what elements you are going to want on the HTML email page -- both text and graphics.

2) On paper, sketch how you'd like the page to look.

3) Sketch in basic lines to help break your table up into the cells you'll need to define with your hidden table.

4) Now that you have the basic layout of the table, simply create the table code and add the elements to the correct cells.

Forms

- Forms are Fun!!!
  They form the basis for interactivity on the World Wide Web.
  - allow visitors to communicate with the page owner
  - guestbooks (where visitors can leave comments viewable to all)
  - allow page owners to gather information about their visitors
  - for online surveys
  - for online exams
  - to purchase supplies/tickets/room reservations

Parts of a Form

- There are two parts to any form:
  - The HTML portion, where the visitor will interact with the webpage
  - The CGI (Common Gateway Interface) program which will process the form.

Forms processing

- Once a user submits the form, the data sent needs to be processed, usually via a CGI program (often written in PERL, PHP, C++, ASP, Visual Basic, or Java).

- Many CGI scripts are freely available to download and utilize on your server without needing to have much expertise in the programming language:
  - http://www.webmonkey.com/2010/02/perl_tutorial_for_beginners/
  - http://www.webmonkey.com/2010/02/php_tutorial_for_beginners/
  - http://github.com/
  - http://code.google.com/
  - http://www.scriptarchive.com/
  - http://www.cgiscript.net
  - http://www.javascript.com
  - http://www.adobe.com/cfusion/exchange/index.cfm

HTML part

- <FORM>: marks the form
- <INPUT>: tag used to define variables and field types
  - TEXT: single-line textbox
  - PASSWORD: single-line textbox but in hidden text
  - RADIO: radio button single-choice selections
  - CHECKBOX: checkbox button multiple-choice selections
  - RESET: used to reset variables back to default value
  - SUBMIT: used to submit form to the CGI script
  - BUTTON: used to submit form to a JavaScript or other client-side script
  - HIDDEN: used to submit hidden information
- <TEXTAREA>: tag used for multiple-line textbox
- <SELECT>: tag used for pull-down menus

<FORM> tag

- Encloses all of the form elements.
  - contains METHOD and ACTION attributes
- First, the form needs to know how to send the information to the server. Two methods are available, GET and POST.
  - METHOD="GET"
  - Most HTML documents are retrieved by requesting a single URL from the server. GET tacks on the information from the form to the end of the URL. Spaces are translated into + sign; if there's more than one variable, they're separated by an & sign:
    `http://www.google.com/search?q=Stanford+University`
  - METHOD="POST"
    With the POST method, the information from the form is sent to the CGI script separately from the URL This is the preferred method. It is the method most commonly used in creating forms, as it is less likely to cause problems if there are many variables and data.

More on <FORM>

- Second, the form needs to know *where* to send the information. This is the URL being requested from the form, and is referenced with the `ACTION` tag. This URL will almost always point to a computer script to decode the form results.

  `ACTION="http://www.company.com/cgi/script.pl"`

- Once you put it all together, your form will usually have the following format:

```
<FORM METHOD="POST" ACTION=”http://www.company.com/cgi/script.pl">

[Form input tags and other HTML goes here]

</FORM>
```

- Notice that this form uses the method `POST` and sends the input information to a local script named `script.pl` in the cgi web directory.

<INPUT> tag

- FORM contents consist primarily of INPUT tags, which define the field types and the names of the variables. These INPUT tags allow the visitor to enter information or to select choices.
- Each INPUT tag is given a TYPE and NAME attributes. These attributes determine what kind of information it contains and the name identifier for the field.
- This is the syntax for an input tag:
  ```
  <input type="option" name="text" />
  ```
- Types available:
  - text
  - password
  - reset
  - checkbox
  - radio
  - submit
  - button
  - hidden
- Other ways to input data:
  - textarea
  - select
- In HTML5, there are now even more ways to input data. See the extra handout "A Form Apart" for details.

<input type="text" />

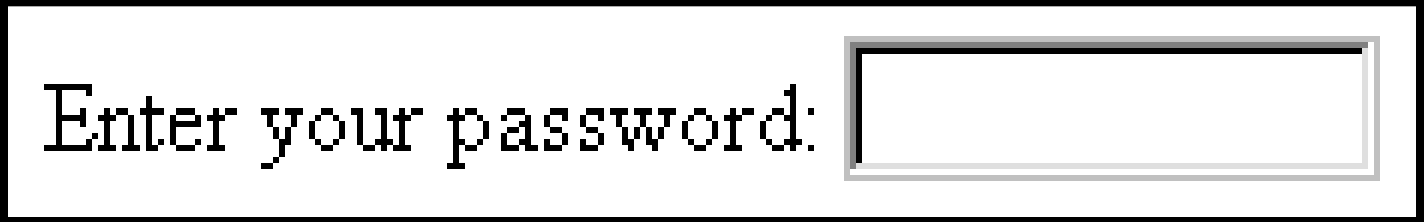
- <input type="text" name="name" size="xx" maxlength="yy" value="default-value" />
- Creates a single-line box for your user to place data. You can set the size, maximum length, and default values if you wish.
- Example:

```
Enter your first name: <input type="text"
name="firstname" size="40" maxlength="40" value="Enter
First Name" />
```

<input type="password" />

- <input type="password" name="name" size="xx" maxlength="yy" value="default-value" />
- Works the exact same way as the text type, but when a user types in the information, asterisks appear instead of text. However, this text is sent to the server in clear-text format!
- Example:

```
Enter your password: <input type="password" name="pass"
size="10" maxlength="10" />
```



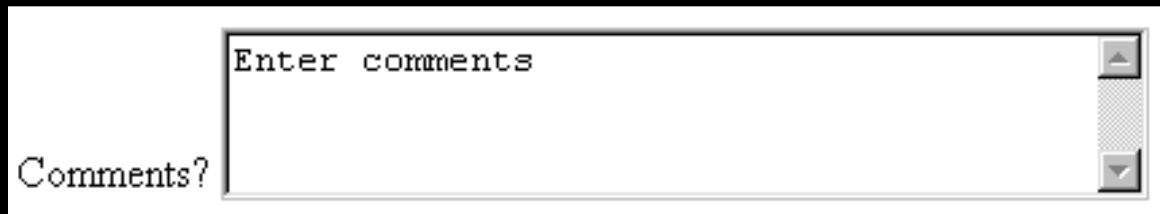
<textarea>default text<textarea>

- Allows user to submit more than one line of text. Usually used for comments or leaving feedback.

<textarea rows="xx" cols="yy" name="text" wrap="off|soft/virtual|hard/physical">default text</textarea>

  - Rows: number of rows
  - Cols: number of columns
  - Name: variable's name
  - Wrap:
    - Off=no wrapping of text allowed
    - Soft or Virtual=wrapping is on, but text is sent to the CGI script as a single line of text, without line breaks
    - Hard or Physical=wrapping is on, and when the text is sent to the CGI script, it is sent with line breaks

```
– Comments?
  <textarea rows="3" cols="40" name="comments”
   wrap="virtual">
  Enter comments
  </textarea>
```



<input type="radio" />

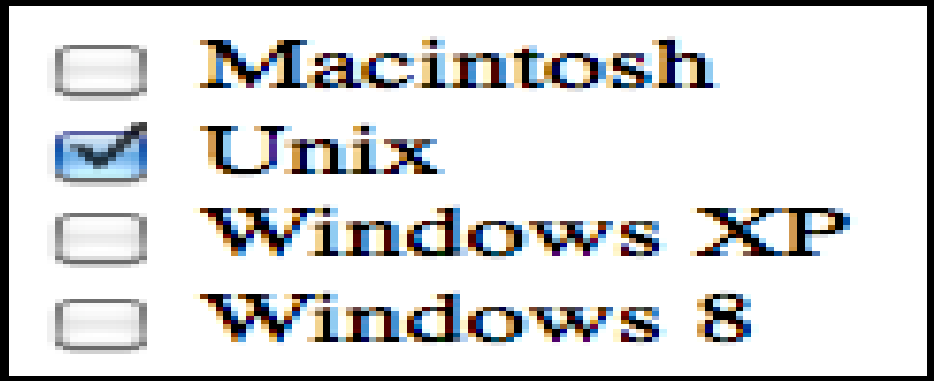
- Provides a group of buttons from which only one may be chosen
- <input type="radio" name="text" value="value" />
- If you want a radio button to be pre-selected, add the attribute "checked".
- Example:

```
<br /><input type="radio" name="univ" value="Stanford" />UCLA
<br /><input type="radio" name="univ" value="Stanford" />Harvard
<br /><input type="radio" name="univ" value="Stanford" />Oxford
<br /><input type="radio" name="univ" value="Stanford"
      checked="checked" />Stanford
```

<input type="checkbox" />

- Works just like the radio button, except that more than one value can be selected.
- <input type="checkbox" name="name" value="value" />
- Example:

```
<br /><input type="checkbox" name="computer" value="Mac" />Macintosh
<br /><input type="checkbox" name="computer" value="Unix"
       checked="checked" />Unix
<br /><input type="checkbox" name="computer" value="WinXP" />
       Windows XP
<br /><input type="checkbox" name="computer" value="Win8" />
       Windows 8
```

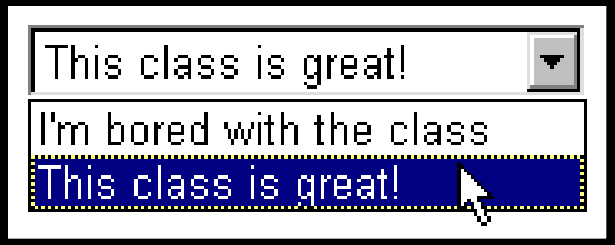


<select></select>

- The select element shows a list of choices in as a pull-down menu.

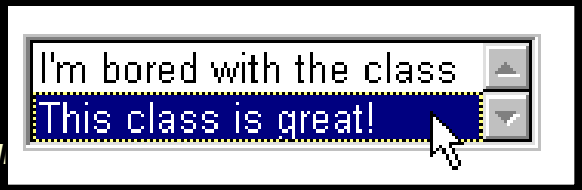
```
<select name="class-enjoyment">
   <option value="bored">I'm bored with the class</option>
   <option value="great" selected="selected">This class is great!</option>

</select>
```



- If instead of a pull-down menu, a list box is desired, add the SIZE attribute (size refers to the number of items):
-

```
<select name="class-enjoyment" size="2">
    <option value="bored">I'm bored with the class</option>
    <option value="great">This class is great!</option>
</select>
```



More <select>

- To allow a user to select more than one item, the MULTIPLE attribute is used:

```
<select name="favorite-food" multiple="multiple">
      <option value="hamburger">Hamburgers</option>
      <option value="pizza">Pizza </option>
      <option value="tacos">Tacos </option>
      <option value="vegetables">Vegetables </option> </select>
```

<input type="reset" />

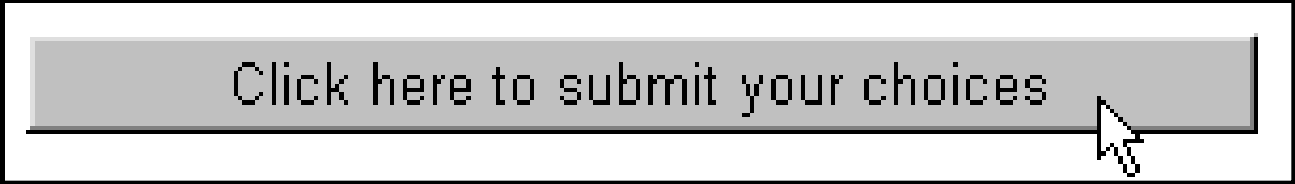
- This choice allows the user to reset and clear all of the data fields to their default values.
- <input type="reset" value="text" />
- Whatever is put in the text for the value attribute will be what is seen as the Reset button.

```
<input type="reset" value="Clear all choices and start over
   again" />
```

<input type="submit" />

- The submit value displays a push button with the preset function of sending the data in the form to the server to be processed, as defined by the action attribute in the <form> tag.
- <input type="submit" value="text" />
- Whatever is put in the text for the value attribute will be what is seen as the Submit button.

```
<input type="submit" value="Click here to submit your choices" />
```

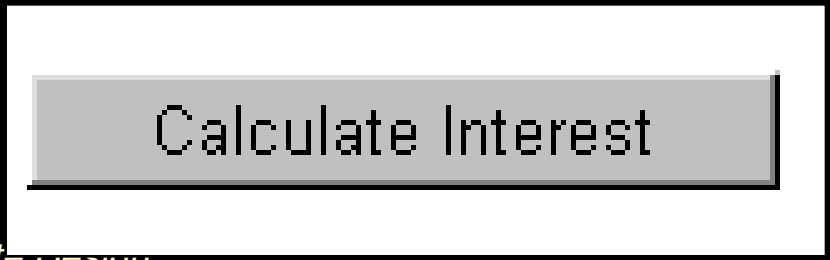


<input type="image" />

- Sometimes, designers wish to create their own submit button using a graphical program. The image value works exactly the same way the submit value works. The SRC attribute defines the URL of the image; the ALT attribute will be displayed if the browser is incapable of displaying images; the width and height attributes define the width and height of the image; the border attribute defines whether a border is desired.
- <input type="image" src="URL-of-image" alt="text" width="xx" height="yy" border="0" />
-
```
<input type="image"
src="http://www.company.com/images/ clickme.gif"
alt="Click Me to Submit" width="30" height="10"
border="0" />
```

<input type="button" />

- Sometimes you'll want to create a webpage that will do an action but won't submit the entire data. For example, you might create a Javascript program to calculate the shipping cost or the tax of a potential order. Whatever is placed in the value attribute will be displayed as the button; the name will be the name called by the script.
- <input type="button" value="text" name="name" />
- This requires the use of a scripting language to tie an event to the button and create an action.
- `<input type="button" value="Calculate Interest" name="calculator" />`



Hidden

- Allows webmasters to submit default or previously specified text that is hidden from the user.
- <input type="hidden" name="name" value="text" />

Hidden (continued)