

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/cs21>

- Unfinished business
- Forms
 - What are Forms?
 - Creating a Form Processor
 - Creating an HTML form

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>
 - <https://creative.adobe.com/addons>

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 checkbox button
 - password radio hidden
 - reset submit
- Other ways to input data:
 - textarea
 - select
- In HTML5, there are now even more ways to input data, although not all browsers support all of the new input type values.

<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" />
```

Enter your 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" />`

Enter your password:

<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:

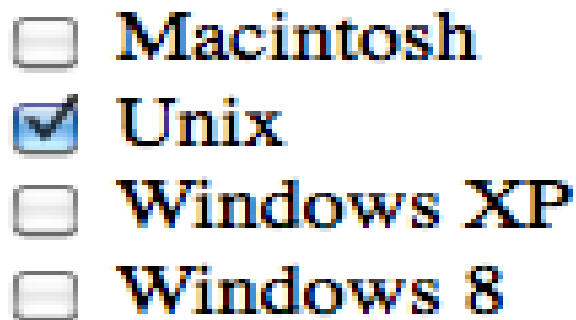
```
<br /><input type="radio" name="univ" value="Stanford" />UCLA  
<br /><input type="radio" name="univ" value="Stanford" />Harvard  
<br /><input type="radio" name="univ" value="Stanford" />Oxford  
<br /><input type="radio" name="univ" value="Stanford"  
    checked="checked" />Stanford
```

```
○ UCLA  
○ Harvard  
○ Oxford  
● 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
```

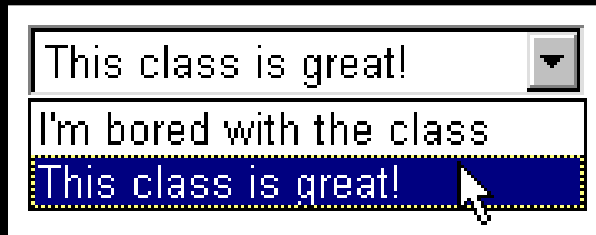


☐ Macintosh
☒ Unix
☐ Windows XP
☐ Windows 8

<select></select>

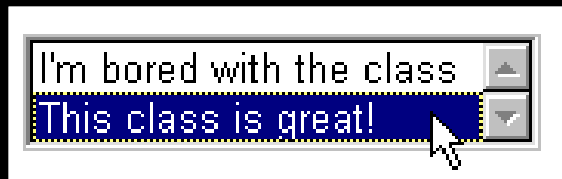
- The select element shows a list of choices in as a pull-down menu.

```
<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" />
```



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" />`

A rectangular button with a light gray background and a thin black border. The text "Calculate Interest" is centered on the button in a black, monospaced font.

Hidden

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

Name=John+Doe&ID=694



Hidden

General info:

Name: John Doe

ID: 694



Part=Widget&Qty=3&Name=John+Doe&ID=694

Order Form

Part Widget

Qty 3

Hidden

Name: John Doe

ID: 694



Client