



中山大學
SUN YAT-SEN UNIVERSITY

Lecture 8

HTML Forms

SE-805 Web 2.0 Programming

(<http://my.ss.sysu.edu.cn/wiki/display/W2PSC/Home> , supported by Google;
using some slides of & inspired by Marty Stepp's CSE 190 M courseware)

School of Software, Sun Yat-sen University

Outline

- **Parameterized Pages**
- Form Basics
- Form Controls
- Processing Form data in PHP

Web data

- most interesting web pages revolve around data
 - examples: Google, Baidu, IMDB, Digg, Facebook, YouTube, renren
 - can take many formats: text, HTML, XML, multimedia
- many of them allow us to access their data
- some even allow us to submit our own new data
- most server-side web programs accept **parameters** that guide their execution

Query strings and parameters

URL?name=value&name=value...

`http://www.google.com/search?q=Obama`

`http://example.com/student_login.php?username=stepp&id=1234567`

- query string: a set of parameters passed from a browser to a web server
 - often passed by placing name/value pairs at the end of a URL
 - above, parameter **username** has value **stepp**, and **id** has value **1234567**
- PHP code on the server can examine and utilize the value of parameters
- a way for PHP code to produce different output based on values passed by the user

Query parameters: **\$ _REQUEST**

```
$user_name = $_REQUEST["username"];  
$id_number = (int) $_REQUEST["id"];  
$eats_meat = FALSE;  
if (isset($_REQUEST["meat"])) {  
    $eats_meat = TRUE;  
}
```

PHP

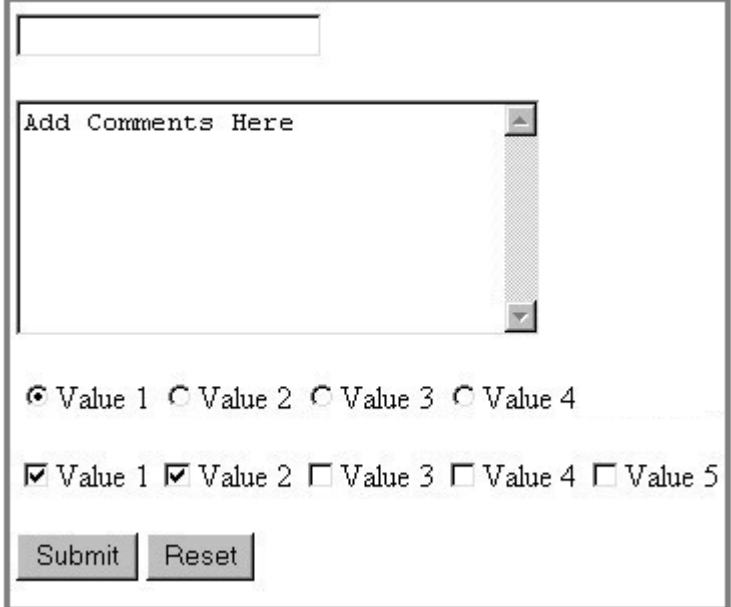
- **\$_REQUEST**["*parameter name*"] returns a parameter's value as a string
- test whether a given parameter was passed with **isset**

Outline

- Parameterized Pages
- **Form Basics**
- Form Controls
- Processing Form data in PHP

HTML forms

- **form**: a group of UI controls that accepts information from the user and sends the information to a web server
- the information is sent to the server as a **query string**



The image shows a web form with the following elements:

- A single-line text input field at the top.
- A multi-line text area below it with the placeholder text "Add Comments Here".
- A row of four radio buttons labeled "Value 1", "Value 2", "Value 3", and "Value 4".
- A row of five checkboxes labeled "Value 1", "Value 2", "Value 3", "Value 4", and "Value 5".
- At the bottom, two buttons labeled "Submit" and "Reset".

HTML form: <form>

```
<form action="destination URL">  
  form controls  
</form>
```

HTML

- required **action** attribute gives the URL of the page that will process this form's data
- when form has been filled out and **submitted**, its data will be sent to the **action**'s URL
- one page may contain many forms if so desired

Form example

```
<form action="http://www.google.com/search">
  <div>
    Let's search Google:
    <input name="q" />
    <input type="submit" />
  </div>
</form>
```

HTML

Let's search Google:

output

- often wrap the form's controls in a block element such as **div**, **fieldset**, etc.

Outline

- Parameterized Pages
- Form Basics
- **Form Controls**
- Processing Form data in PHP

Form controls: <input>

```
<!-- 'q' happens to be the name of Google's required parameter -->
<input type="text" name="q" value="Colbert Report" />
<input type="submit" value="Booyah!" />
```

HTML

output

- **input** element is used to create many UI controls
 - an inline element that **MUST be self-closed**
- **name** attribute specifies name of query parameter to pass to server
- **type** can be **button**, **checkbox**, **file**, **hidden**, **password**, **radio**, **reset**, **submit**, **text**, ...
- **value** attribute specifies control's initial text

Text fields: <input>

```
<input type="text" size="10" maxlength="8" /> NetID <br />
<input type="password" size="16" /> Password
<input type="submit" value="Log In" />
```

HTML

NetID

Password

output

- **input** attributes: **disabled**, **maxlength**, **readonly**, **size**, **value**
- **size** attribute controls onscreen width of text field
- **maxlength** limits how many characters user is able to type into field

Text boxes: <textarea>

```
<textarea rows="4" cols="20">  
Type your comments here.  
</textarea>
```

HTML

Type your comments
here.

output

- initial text is placed inside **textarea** tag (optional)
- required **rows** and **cols** attributes specify height/width in characters
- optional **readonly** attribute means text cannot be modified

Checkboxes: <input>

yes/no choices that can be checked and unchecked (inline)

```
<input type="checkbox" name="lettuce" /> Lettuce
<input type="checkbox" name="tomato" checked="checked" /> Tomato
<input type="checkbox" name="pickles" /> Pickles
```

HTML

☐ Lettuce ☒ Tomato ☐ Pickles

output

- none, 1, or many checkboxes can be checked at same time
- when sent to server, any checked boxes will be sent with value on:
 - <http://ssw2p.3322.org/public/params.php?tomato=on&pickles=on>
- use checked="checked" attribute in HTML to initially check the box

Radio buttons: <input>

sets of mutually exclusive choices (inline)

```
<input type="radio" name="cc" value="visa" checked="checked" /> Visa
<input type="radio" name="cc" value="mastercard" /> MasterCard
<input type="radio" name="cc" value="amex" /> American Express
```

HTML

• Visa • MasterCard • American Express 提交查询

output

- grouped by **name** attribute (only one can be checked at a time)
- must specify a **value** for each one or else it will be sent as value **on**

Think of <input>

- So many types of **input**, why **NOT** use elements instead?
- `<input type="text" ... />` → `<text/>` or `<text></text>`
- `<input type="checkbox" ... />` → `<checkbox .../>`
- In fact, it is just a bad design decision when form was firstly designed and introduced into html in 1996,
- and we follow it so far...,
- another flaw: `checked="checked"` ..., is it weird?
- Lessons:
- Reality is never, ever perfect
- BUT we will try out best to make it perfect

Text labels: <label>

```
<label><input type="radio" name="cc" value="visa" checked="checked" /> Visa</label>
<label><input type="radio" name="cc" value="mastercard" /> MasterCard</label>
<label><input type="radio" name="cc" value="amex" /> American Express</label> HTML
```

• Visa • MasterCard • American Express

output

- associates nearby text with control, so you can click text to activate control
- can be used with checkboxes or radio buttons
- either wrap the input elements or target input elements with id specified via the “for” attribute
- **label** element can be targeted by CSS style rules
- reasons for preferring label than text:
 - **functionality**: can be directly clicked on
 - **styling**: can be styled by CSS rules
 - **accessibility**: screen reader will read it when selected

Drop-down list: <select>, <option>

menus of choices that collapse and expand (inline)

```
<select name="favoritecharacter">
  <option>Jerry</option>
  <option>George</option>
  <option selected="selected">Kramer</option>
  <option>Elaine</option>
</select>
```

HTML

Kramer 提交查询


output

- **option** element represents each choice
- **select** optional attributes: **disabled**, **multiple**, **size**
- optional **selected** attribute sets which one is initially chosen

Using `<select>` for lists

```
<select name="favoritecharacter[]" size="3" multiple="multiple">  
  <option>Jerry</option>  
  <option>George</option>  
  <option>Kramer</option>  
  <option>Elaine</option>  
  <option selected="selected">Newman</option>  
</select>
```

HTML



output

- optional **multiple** attribute allows selecting multiple items with shift- or ctrl-click
 - must declare parameter's name with **[]** if you allow multiple selections
- **option** tags can be set to be initially **selected**

Option groups: <optgroup>

```
<select name="favoritecharacter">
  <optgroup label="Major Characters">
    <option>Jerry</option>
    <option>George</option>
    <option>Kramer</option>
    <option>Elaine</option>
  </optgroup>
  <optgroup label="Minor Characters">
    <option>Newman</option>
    <option>Susan</option>
  </optgroup>
</select>
```

HTML

Jerry

提交查询

output

Outline

- Parameterized Pages
- Form Basics
- Form Controls
- **Processing Form data in PHP**

"Superglobal" arrays

- PHP superglobal arrays (global variables) contain information about the current request, server, etc.:

Array	Description
<u>\$ _GET</u> , <u>\$ _POST</u>	parameters passed to GET and POST requests
<u>\$ _REQUEST</u>	parameters passed to any type of request
<u>\$ _SERVER</u> , <u>\$ _ENV</u>	information about the web server
<u>\$ _FILES</u>	files uploaded with the web request
<u>\$ _SESSION</u> , <u>\$ _COOKIE</u>	"cookies" used to identify the user (seen later)

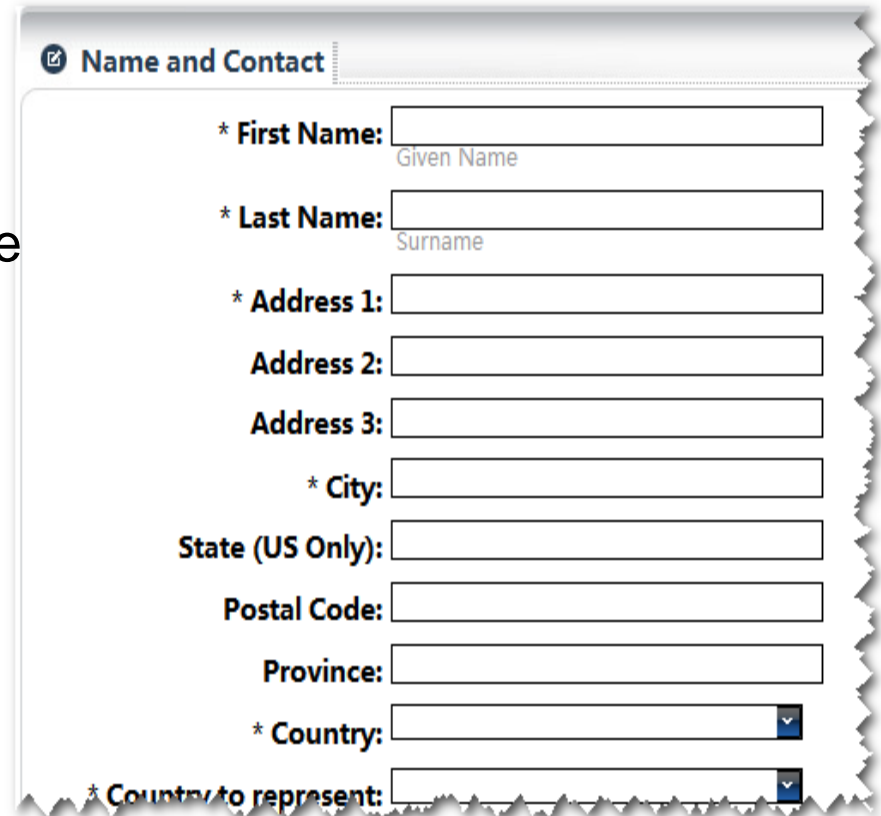
- These are special kinds of arrays called associative arrays.

Summary

- Query String & Parameters
- Form Basics
- Form Controls
 - **input**
 - **type**: text, password, textarea, checkbox, radio
 - **label**
 - **select, option**
- Processing Form data in PHP
 - Superglobal arrays: **\$_GET**, **\$_POST**, ...

Exercises

- write a php page to mimic the registration page of Topcoder
- get initial html from <https://www.topcoder.com/reg/>
- convert it to a php page, which shows data you submitted at the top of this page



The screenshot shows a registration form titled "Name and Contact" with a small icon of a notepad and pencil. The form contains the following fields:

- * First Name:** Text input field with the placeholder text "Given Name".
- * Last Name:** Text input field with the placeholder text "Surname".
- * Address 1:** Text input field.
- Address 2:** Text input field.
- Address 3:** Text input field.
- * City:** Text input field.
- State (US Only):** Text input field.
- Postal Code:** Text input field.
- Province:** Text input field.
- * Country:** Dropdown menu.
- * Country to represent:** Dropdown menu.

Further Readings

- PHP home page: <http://www.php.net/>
- W3Schools PHP tutorial: <http://www.w3schools.com/PHP/>
- Practical PHP Programming: <http://hudzilla.org/phpwiki/>
- PHP Cookbook:
http://commons.oreilly.com/wiki/index.php/PHP_Cookbook

Thank you!

