中山大學
SUN YAT-SEN UNIVERSITY

Lecture 8
HTML Forms

SE-805 Web 2.0 Programming (supported by Google)

http://my.ss.sysu.edu.cn/courses/web2.0/

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

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

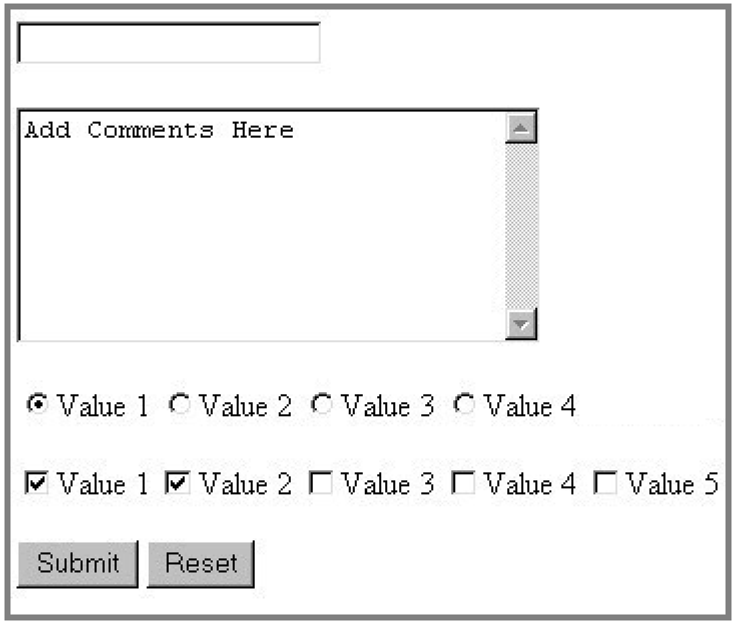
- $_REQUEST["*parameter name*"] returns a parameter's value as a string
- Test whether a given parameter was passed with isset

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**


Add Comments Here
Value 1 Value 2 Value 3 Value 4
Value 1 Value 2 Value 3 Value 4 Value 5
Submit
Reset

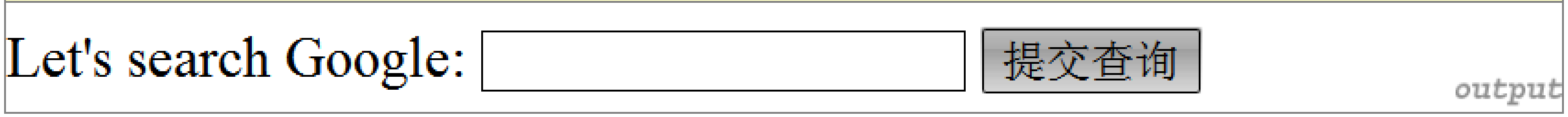
HTML Form: <form>

```
<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

```html
<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: 提交查询

output

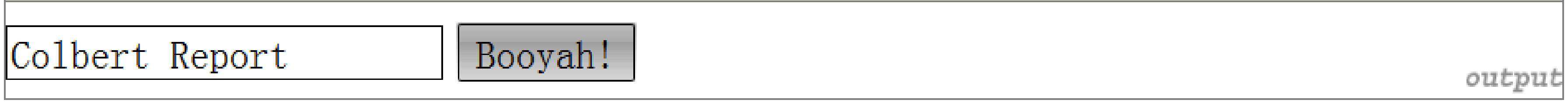
- Must wrap the form's controls in a block element such as div, fieldset, etc.

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

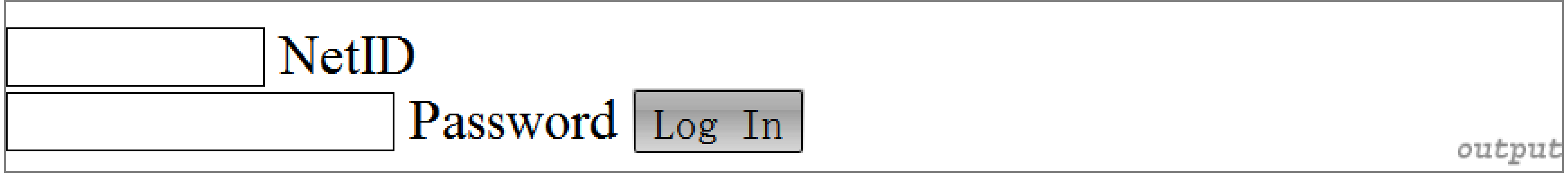


- input element is used to create many UI controls
  - an inline element that **MUST be self-closed**
- name attribute specifies name of query parameter to pass to server
- type can be button, checkbox, file, hidden, password, radio, reset, submit, text, ...
- value attribute specifies control's initial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

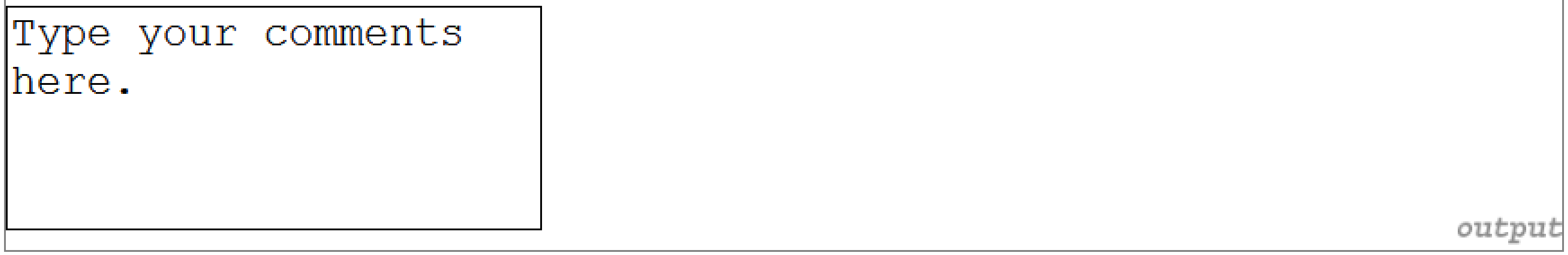


- input attributes: disabled, maxlength, readonly, size, value
- size attribute controls onscreen width of text field
- maxlength limits how many characters user is able to type into field

Text Boxes: <textarea>

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

HTML



```
Type your comments
here.
```

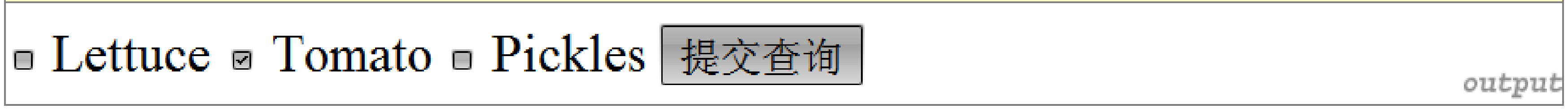
output

- 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 提交查询

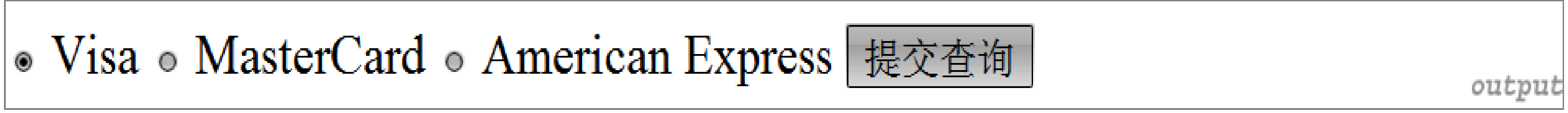
output

- 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)

```html
<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



- 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>

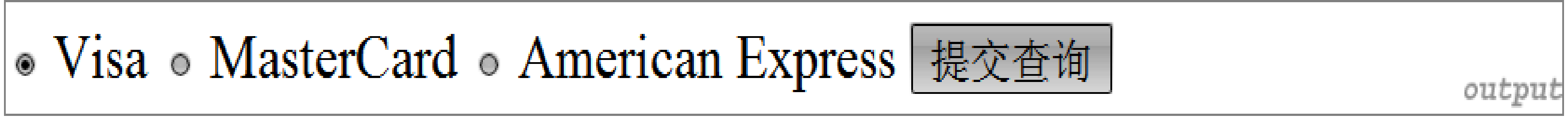
- So many types of input, why **NOT** use elements instead?
- <input type="text" … /> ➔ <text/> or <text></text>
- <input type="checkbox" … /> ➔ <checkbox …/>

- 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>

```html
<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

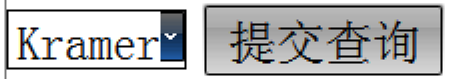
- Associates nearby text with control, so you can click text to activate control
- Can be used with checkboxes or radio buttons
- Either wrap the input elements or target input elements with id specified via the “for” attribute
- label element can be targeted by CSS style rules
- Reasons for preferring 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

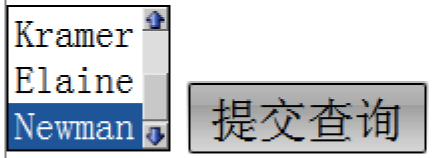


output

- 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



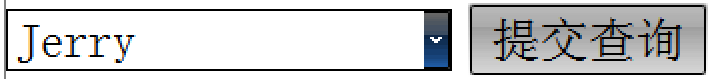
output

- 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



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

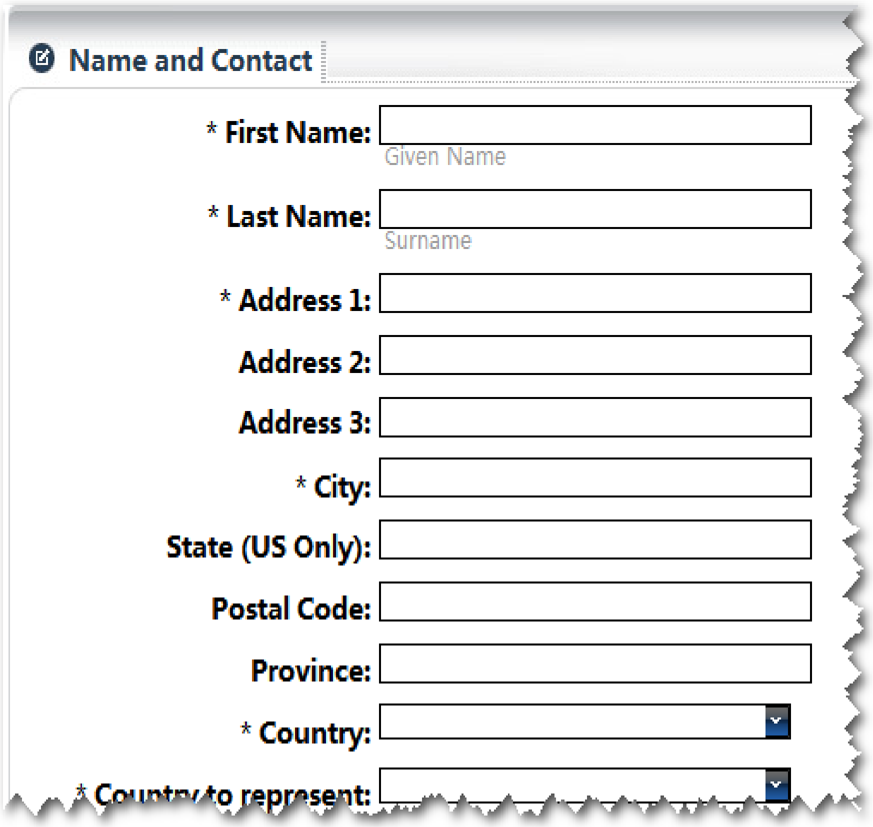
- These are special kinds of arrays called associative arrays.

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 on your Web server, which shows data you submitted at the top of this page



Name and Contact

* First Name:	Given Name
* Last Name:	Surname
* Address 1:	
Address 2:	
Address 3:	
* City:	
State (US Only):	
Postal Code:	
Province:	
* Country:	
* Country to represent:	

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!