

CSE 154

LECTURE 9: FORMS



“This is interesting, 70% of the respondents to our survey said they don't respond to surveys.”

Web data

- most interesting web pages revolve around data
 - examples: Google, IMDB, Digg, Facebook, YouTube, Rotten Tomatoes
 - 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=Romney
```

```
http://example.com/student_login.php?username=obourn&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 obourn, and sid 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: \$_GET, \$_POST

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

PHP

- `$_GET["parameter name"]` or `$_POST["parameter name"]` returns a GET/POST parameter's value as a string
- parameters specified as `http://....?name=value&name=value` are GET parameters
- test whether a given parameter was passed with `isset`

Example: Exponents

```
$base = $_GET["base"];  
$exp = $_GET["exponent"];  
$result = pow($base, $exp);  
print "$base ^ $exp = $result";
```

PHP

exponent.php?base=3&exponent=4

3 ^ 4 = 81

output

Example: Print all parameters

```
<?php foreach ($_GET as $param => $value) { ?>
  <p>Parameter <?= $param ?> has value <?= $value ?></p>
<?php } ?>
```

PHP

```
print_params.php?name=Allison+Obourn&sid=1234567
```

Parameter name has value Allison Obourn

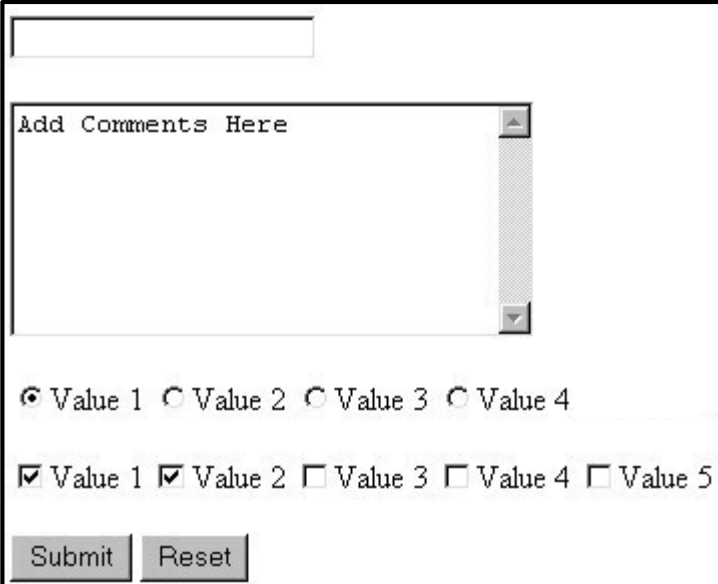
Parameter sid has value 1234567

output

- or call `print_r` or `var_dump` on `$_GET` for debugging

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**
- JavaScript can be used to create interactive controls (seen later)



The image shows a screenshot of a web form. At the top is a single-line text input field. Below it is a text area with the placeholder text "Add Comments Here" and a vertical scrollbar on the right side. Underneath the text area are four radio buttons labeled "Value 1", "Value 2", "Value 3", and "Value 4", with "Value 1" selected. Below the radio buttons are five checkboxes labeled "Value 1", "Value 2", "Value 3", "Value 4", and "Value 5", with "Value 1" and "Value 2" checked. At the bottom of the form are two buttons: "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 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

Colbert Report

Booyah!

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>

a multi-line text input area (inline)

```
<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" checked="checked" /> 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://webster.cs.washington.edu/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

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

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
- label element can be targeted by CSS style rules

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 ▾ Submit Query

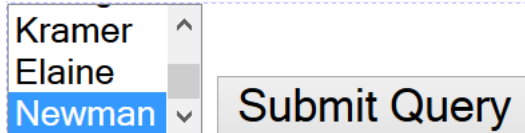
output

- 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



Kramer
Elaine
Newman

Submit Query

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



Submit Query

output

- What should we do if we don't like the bold appearance of the optgroups?

Reset buttons

```
Name: <input type="text" name="name" /> <br />
Food: <input type="text" name="meal" value="pizza" /> <br />
<label>Meat? <input type="checkbox" name="meat" /></label> <br />
<input type="reset" /> HTML
```

Name:

Food:

Meat?

output

- when clicked, returns all form controls to their initial values
- specify custom text on the button by setting its value attribute

Hidden input parameters

```
<input type="text" name="username" /> Name <br />  
<input type="text" name="sid" /> SID <br />  
<input type="hidden" name="school" value="UW" />  
<input type="hidden" name="year" value="2048" />
```

HTML

<input type="text"/>	Name
<input type="text"/>	SID
<input type="submit" value="Submit Query"/>	

output

- an invisible parameter that is still passed to the server when form is submitted
- useful for passing on additional state that isn't modified by the user

Grouping input: <fieldset>, <legend>

groups of input fields with optional caption (block)

```
<fieldset>
  <legend>Credit cards:</legend>
  <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
</fieldset>
```

HTML

Credit cards:

Visa MasterCard American Express

Submit Query

output

- fieldset groups related input fields, adds a border; legend supplies a caption

Styling form controls

```
element [attribute="value"] {  
  property : value;  
  property : value;  
  ...  
  property : value;  
}
```

CSS

```
input [type="text"] {  
  background-color: yellow;  
  font-weight: bold;  
}
```

CSS

Borat

output

- attribute selector: matches only elements that have a particular attribute value
- useful for controls because many share the same element (input)