Introduction to Web Programming

Lecture 9: Forms

9.1: Form Basics

- **9.1: Form Basics**
- 9.2: Form Controls

Web data

- most interesting web pages revolve around data
  - examples: Baidu, Google, IMDB, Digg, Weibo, Facebook, YouKu, 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=Marc
http://example.com/student_login.php?username=marc&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 `marc`, 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"];
$eats_meat = FALSE;
if (isset($_GET["meat"])) {
  $eats_meat = TRUE;
}
```

- `$_GET["`*parameter name*`"]` or `$_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";
```

```
exponent.php?base=3&exponent=4
```

```
3 ^ 4 = 81
```

Example: Print all parameters

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

```
print_params.php?name=Marc+Gaetano&sid=1234567
```

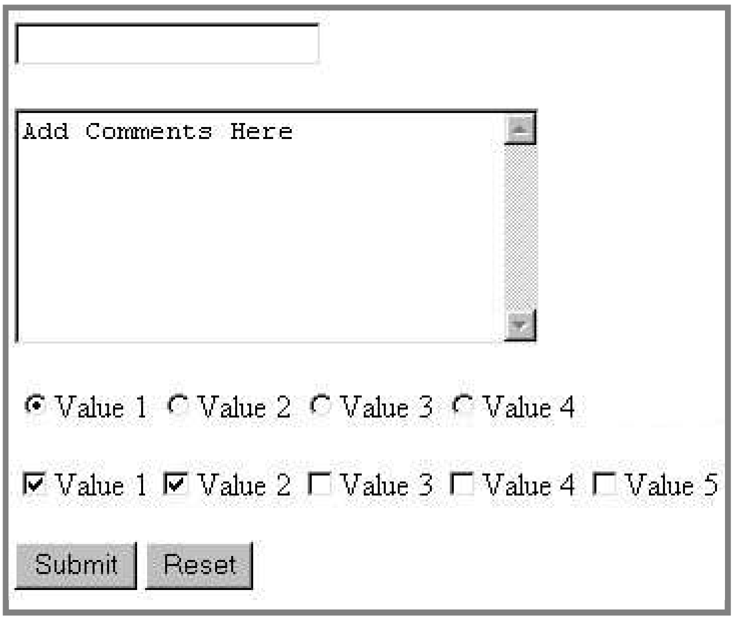
```
Parameter name has value Marc Gaetano

Parameter sid has value 1234567
```

- 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)



Add Comments Here

● Value 1 ○ Value 2 ○ Value 3 ○ Value 4

☑ Value 1 ☑ Value 2 □ Value 3 □ Value 4 □ Value 5

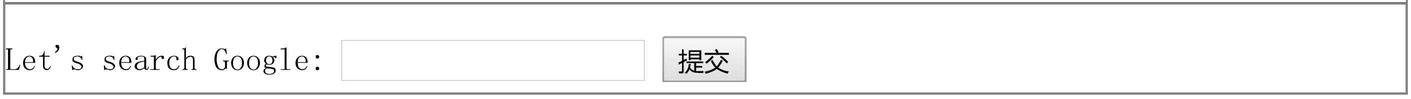
HTML form: `<form>`

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

- 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>
```



Let’s search Google:

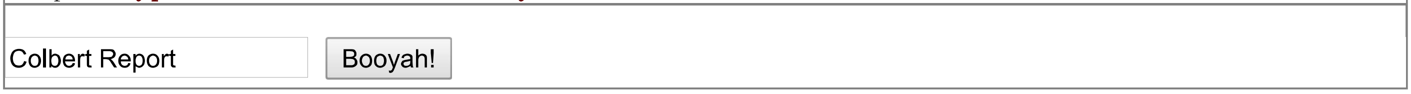
- must wrap the form's controls in a block element such as `div`

9.2: Form Controls

- 9.1: Form Basics
- **9.2: Form Controls**

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!" />
```

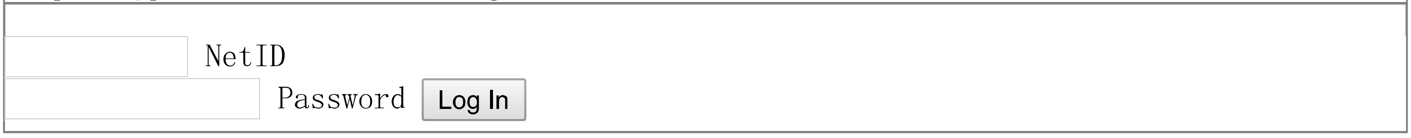


Colbert Report

- 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″ />
```

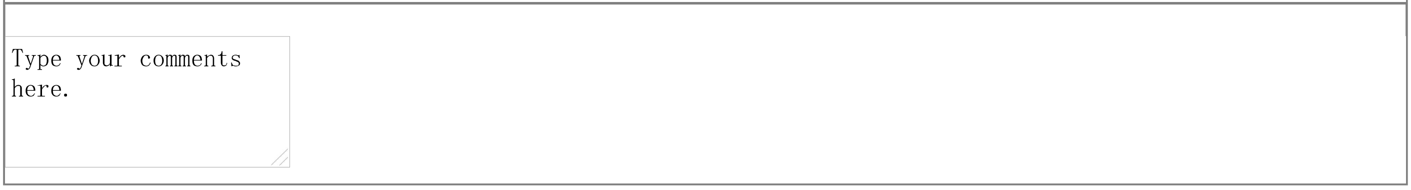


- `input` attributes: `disabled`, `maxlength`, `readonly`, `size`, `value`
- `size` attribute controls onscreen width of text field
- `maxlength` limits how many characters user is able to type into field

Text boxes: `<textarea>`

a multi-line text input area (inline)

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



Type your comments here.

- 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
```

☐ Lettuce ☑ Tomato ☑ Pickles 提交

- 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://www.polytech.unice.fr/~gaetano/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
```

● Visa ○ MasterCard ○ American Express 提交

- 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>
```

● Visa ○ MasterCard ○ American Express 提交

- 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

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" />
```

Name:

Food: pizza

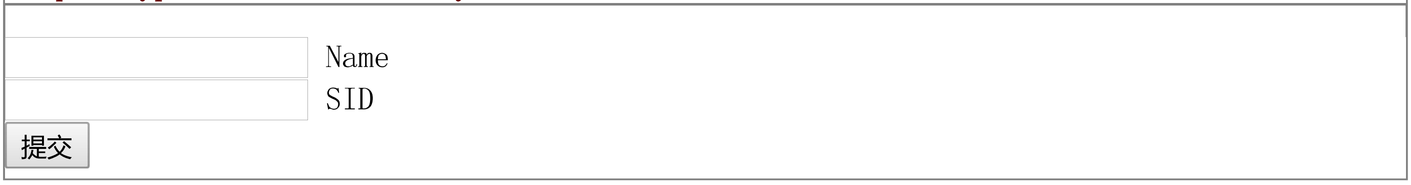
Meat? □

重置 提交

- 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" />
```



- 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

Styling form controls

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

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

Borat

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