

1

# Form Basics

# Web Data

2

- 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

# Reading/writing an entire file

3

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

```
http://example.com/student_login.php?username=xenia&sid=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
- PHP code on the server can examine and utilize the value of parameters

# HTML forms

4

- **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 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". The "Value 1" radio button is selected.
- A row of five checkboxes labeled "Value 1", "Value 2", "Value 3", "Value 4", and "Value 5". The "Value 1" and "Value 2" checkboxes are checked.
- Two buttons at the bottom: "Submit" and "Reset".

# HTML form: `<form>`

5

```
<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

# Form example

6

```
<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:

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

7

# Form controls

# Form controls: `<input>`

8

```
<!-- '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



# Form controls: `<input>` (cont.)

9

```
<!-- '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!

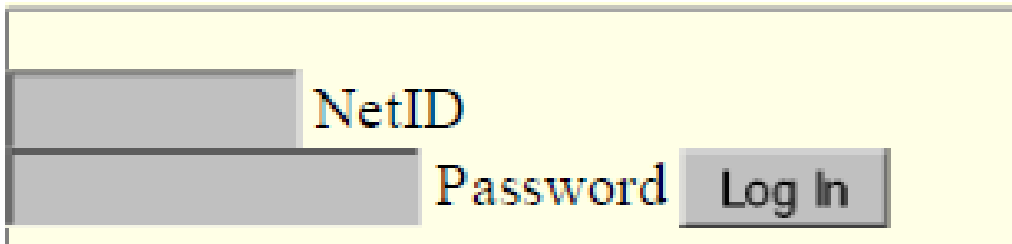
- `type` can be button, checkbox, file, hidden, password, radio, reset, submit, text, ...
- `value` attribute specifies control's initial text

# Text fields: `<input>`

10

```
<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>

11

```
<textarea rows="4" cols="20">
```

Type your comments here.

```
</textarea>
```

HTML

Type your comments  
here.

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

# Check boxes: `<input>`

12

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

HTML

- none, 1, or many checkboxes can be checked at same time

# Radio buttons: <input>

13

```
<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

# Text labels: <label>

14

```
<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

- 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 lists: `<select>`, `<option>`

15

```
<select name="favoritecharacter">
  <option>Frodo</option>
  <option>Bilbo</option>
  <option selected="selected">Gandalf</option>
  <option>Galadriel</option>
</select>
```

HTML

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

# Using: `<select>` for lists

16

```
<select name="favoritecharacter[]" size="3"
multiple="multiple">
  <option>Frodo</option>
  <option>Bilbo</option>
  <option>Gandalf</option>
  <option>Galandriel</option>
  <option selected="selected">Aragorn</option>
</select>
```

HTML

- 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>

17

```
<select name="favoritecharacter">
  <optgroup label="Major Characters">
    <option>Frodo</option>
    <option>Sam</option>
    <option>Gandalf</option>
    <option>Aragorn</option>
  </optgroup>
  <optgroup label="Minor Characters">
    <option>Galandriel</option>
    <option>Bilbo</option>
  </optgroup>
</select>
```

HTML

- What should we do if we don't like the bold italic?