ADVANCED HTML (2)

Data entry with HTML Forms

Alex Sánchez

Forms

- ✓ What are they used for?
- ✓ Form components
  - ✓ Text fields
  - ✓ Check boxes
  - ✓ Radio buttons
  - ✓ Menus
  - ✓ Buttons
  - ✓ Hidden fields
  - ✓ Loading files fields
- ✓ ¿How is information sent/processed?
- ✓ Field validation

What are forms?

`<form>` is just another kind of HTML tag

HTML forms are used to create GUIs on Web pages

> Usually the purpose is to ask the user for information

> The information is then sent back to the server

A form is an area that can contain form elements

> The syntax is: `<form parameters> ...form elements... </form>`

> Form elements include: buttons, checkboxes, text fields, radio buttons, drop-down menus, etc

> Other kinds of HTML tags can be mixed in with the form elements

> A form usually contains a Submit button to send the information in the form elements to the server

The <form> tag

The <form *arguments*> … </form> tag encloses form elements (and probably other HTML as well)

The arguments to form tell what to do with the user input

action="*url*"   (required)

> Specifies where to send the data when the Submit button is clicked

method="get" (default)

> Form data is sent as a URL with ?form_data info appended to the end

> Can be used *only* if data is all ASCII and not more than 100 characters

method="post"

> Form data is sent in the body of the URL request

> Cannot be bookmarked by most browsers

target="*target*"

> Tells where to open the page sent as a result of the request

> *target*= _blank means open in a new window

> *target*= _top means use the same window

The <input> tag

Most, but not all, form elements use the input tag, with a type="…" argument to tell which kind of element it is

> type can be text, checkbox, radio, password, hidden, submit, reset, button, file, or image

Other common input tag arguments include:

> name: the name of the element
>
> value: the "value" of the element; used in different ways for different values of type
>
> readonly: the value cannot be changed
>
> disabled: the user can't do anything with this element

Other arguments are defined for the input tag but have meaning only for certain values of type

5

Text input

A text field:
    `<input type="text" name="textfield" value="with an initial value">`

A text field: `with an initial value`

A multi-line text field
    `<textarea name="textarea" cols="24" rows="2">Hello</textarea>`

A multi-line text field `Hello`

A password field:
    `<input type="password" name="textfield3" value="secret">`

A password field: `•••••••`

input   textarea

6

Buttons

A submit button:
      `<input type="submit" name="Submit" value="Submit">`

A reset button:
      `<input type="reset" name="Submit2" value="Reset">`

A plain button:
      `<input type="button" name="Submit3" value="Push Me">`

A submit button: **Submit**

A reset button: **Reset**

A plain button: **Push Me**

submit: send data

reset: restore all form elements to their initial state

button: take some action as specified by JavaScript

7

Checkboxes

A checkbox:

    `<input type="checkbox" name="checkbox"`
        `value="checkbox" checked>`

A checkbox: ☑

type: "checkbox"

name: used to reference this form element

value: value to be returned when element is checked

Note that there is *no text* associated with the checkbox—you have to supply text in the surrounding HTML

Radio buttons

Radio buttons:
    <input type="radio" name="radiobutton" value="myValue1">
    male
    <input type="radio" name="radiobutton" value="myValue2" checked>
    female

Radio buttons:
  ○ male
  ◉ female

If two or more radio buttons have the same name, the user can only select one of them at a time

This is how you make a radio button "group"

If you ask for the value of that name, you will get the value specified for the selected radio button

As with checkboxes, radio buttons do not contain any text

Drop-down menu or list

A menu or list:
```
<select name="select">
    <option value="red">red</option>
    <option value="green">green</option>
    <option value="BLUE">blue</option>
</select>
```

A menu or list: red ⬍

Additional arguments:

size: the number of items visible in the list (default is "1")

multiple: if set to "true", any number of items may be selected (default is "false")

Hidden fields

`<input type="hidden" name="hiddenField" value="nyah">`
`<-- right there, don't you see it?`

A hidden field: <-- right there, don't you see it?

What good is this?

All input fields are sent back to the server, including hidden fields

This is a way to include information that the user doesn't need to see (or that you don't want her to see)

The value of a hidden field can be set programmatically (by JavaScript) before the form is submitted

A complete example

```html
<html>
<head>
<title>Get Identity</title>
<meta http-equiv="Content-Type" content="text/html;
        charset=iso-8859-1">
</head>
<body>
<p><b>Who are you?</b></p>
<form method="post" action="">
  <p>Name:
    <input type="text" name="textfield">
  </p>
  <p>Gender:
    <input type="radio" name="gender" value="m">Male
    <input type="radio" name="gender" value="f">Female</p>
  </form>
</body>
</html>
```

Who are you?

Name: [   ]

Gender: ◯ Male ◯ Female

Exercise

- Create a standalone html page that contains.
  - A short explanation on processing some information
    - Statistical calculation (e.g. sample size)
    - Simple currency converter
  - Give it a quick, nice aspect using bootstrap.css
  - Add a form to collect the data to be processed
    - The form should include the basic elements
      - Text input, Check and radio buttons, Submit buttons