

HTML FORM STRUCTURE

HTML FORM

- ❖ A form structure is a practical method for collecting information from the user
- ❖ Forms on a web page are set up using a combination of HTML form tag elements and an associated script written in PHP, ASP, Perl, python, or other script language
- ❖ Scripts are commonly called CGI (Common Gateway Interface) and run on the web server

1

2

HOW DO THEY WORK?

- ❖ Front-end HTML markup is what users see – buttons, input fields, drop-downs, radio buttons (all known as FORM CONTROLS)
- ❖ Together with a server-side script we can.. store the data from the form, email it, verify it and return an appropriate response to users. The script is what makes the form's action attribute work!

[Form example](#)

3

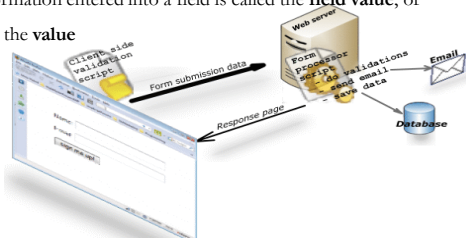
CONTROL TYPES (HTML FORM ELEMENTS)

- HTML v4 form elements (called controls) include
 - Input boxes for text and numeric values
 - Radio buttons, also called option buttons to select a single option from a predefined list
 - **Submit** button to submit the updated form data to the CGI script
 - **Reset** button to reset the form to its original state
 - **Check boxes** to specify an item as either being selected or not selected
 - ...Plus there are many new HTML5 input types
 - Selection lists for longer lists of options in a drop-down listbox
 - Text areas for longer amounts of text

5

HTML FORM

- ❖ Each form element in which the user can enter information is called a **field**
- ❖ Information entered into a field is called the **field value**, or simply the **value**



6

SIMPLE EXAMPLE

```

<form action="/my-handling-form-page"
method="post">
  <label for="name">Name:</label>
  <input type="text" id="name" />
  <label for="mail">E-mail:</label>
  <input type="email" id="mail" />
  <label for="msg">Message:</label>
  <textarea id="msg">by default this element is
  filled with this text</textarea> </div>
</form>
  
```

7

HTML FORM CONTROLS

❖ Controls (form elements) have attributes

- `name`, used to associate the form control's current data when it is sent to the server (variable name – needs to be simple and descriptive)
- `id`, used to uniquely identify a form control in the DOM and associate a label (optional)
- `value`, used to provide an initial value to a form control (optional)

```
<label for="fname">Enter name:</label>
```

```
<input type="text" name="firstname" id="fname"
value="Your first name">
```

8

FORM TAG

- The **<form>** element includes properties that control how the form is processed, information on which CGI script is used, how the data is transferred to the script

```
<form name = "myForm"
      action="script name"
      method="method name" > <!--
```

```
(post, get)-->
```

```
... form elements
```

```
</form>
```

9

FORM TAG

- A form structure may not be defined inside an existing form structure
- Multiple forms may be defined in a page but each form should have a unique *name* attribute value
- Form names should not contain spaces
- Omitting the form's *action* attribute will not break the form but will mean nothing happens when user clicks on the Submit
- Form controls must be defined inside the **<form>** element

10

TEXTAREA

- A field for entering multiple lines of information
- By default, the shape is a blank field of four lines and 40 characters wide each.
- *Name* attribute required
- *Maxlength* sets the limit of the number of characters that can be set into a field.
- To control how the text is wrapped in a text area, use the *wrap* attribute (soft is the default)
 - `wrap="off"` turns off text wrapping
 - `wrap="soft"` or `"virtual"` turns text wrapping on but does not send text wrapping data to web server
 - `wrap="hard"` or `"physical"` turns text wrapping on and also sends text wrapping data to web server

11

TEXTAREA FORM CONTROLS

- ❖ For more than one line of text use textarea instead of input.

```
<form name="myForm" ... >
  <textarea name="notes"
            rows="4"
            columns="20"
            placeholder="Some default text
appears in here." >
  </textarea>
</form>
```

- ❖ By default the text area in the form will be four lines and 40 characters wide.

12

HTML TEXTAREA EXAMPLE

A simple form that uses textarea

1. Story:

2. Comments:

3. Notes:

Some default text appears in here.

4. Confirm:

You cannot edit this text.

13

INPUT FORM ELEMENT

- Unlike `textarea` and `select` elements the input element has a single tag (it is self-closing, like `img`)
- Used for single line text, check boxes, radio buttons, passwords, and many other form element types
- Attributes:
 - `type`, tells the browser what kind of information is being input
 - `name`, the name of the data field
 - `size`, the length of the field in characters, default is 20
 - `maxlength`, max number of input characters to accept
 - `value`, usage depends on the type of input element

14

INPUT ELEMENT – TEXT TYPE

❖ Type attribute

Text, default value for type attribute, the text type is used in cases a type is specified unknown by the browser (e.g. HTML 5 has new input types like `tel`, `url`, `search` and `email`), this form element accepts text from the user.

```
<input name="firstname">
<input type="text"
       name="phone"
       size="15"
       maxlength="12">
<input name="country"
       value="Canada">
```

15

INPUT FORM ELEMENT

❖ `value` attribute, if the input element type is text or password, this is the default text to display; for checkbox or radio button, this is the value returned to the server; for submit and reset buttons this is the text to show inside the button

❖ `checked` attribute, sets a radio button or check box to "on" – usage: `checked="checked"`

❖ Try it out on the W3Schools

http://www.w3schools.com/html/html5_form_input_types.asp

16

HTML INPUT TEXT EXAMPLE

1. First name	<input type="text"/>
2. Last name	<input type="text"/>
3. Address	<input type="text" value="Enter your address here"/>
4. Reason	<input type="text" value="This is readonly text."/>

17

INPUT ELEMENT – PASSWORD TYPE

❖ `password`, a type of text field but the user's typed characters are shown onscreen as bullets to prevent echoing of characters – **this is not encrypting the data sent to the server**

```
<input type="password"
       name="password"
       size="30"
       maxlength="30" >
```

18

INPUT ELEMENT – CHECKBOX TYPE

❖ `checkbox`, a switch control is "on" when its checked attribute is set (e.g. `checked`)

❖ checkbox controls within the form may share the same name value

- Let's the user select zero, one, or many selectable options
- Can make them preselected with `checked="checked"`
- To make PHP scripts work easier place `[]` at the end of the name attribute
- If `value` attribute is not provided, then either 0 or 1 is used

19

CHECKBOX EXAMPLE

```
<input type="checkbox" name="os[]" value="unix"> Unix
<input type="checkbox" name="os[]" value="Win" > Windows XP
<input type="checkbox" name="os[]" value="Mac" checked=""> Mac OS
```

☐ Unix ☐ Windows XP ☒ Mac OS

20

INPUT ELEMENT – RADIO BUTTON TYPE

- ❖ Allows only one option selected
- ❖ All the radio elements must have same name attribute
- ❖ Can supply optional checked="checked" to preselect as the default

```
<input type="radio" name="user" value="home"> Home
<input type="radio" name="user" value="business" checked=""> Business
```

21

INPUT ELEMENT – HIDDEN TYPE

- ❖ Creates an undisplayed field in the form
- ❖ Used to send calculated or some default value information to the server when form data is submitted

```
<input type="hidden" name="IM" value="smmh@hotmail.com">
```

22

INPUT ELEMENT – RESET TYPE

- ❖ Displays a push button with the present function of clearing all the data in the form back to its original state
- ❖ Name and value attributes are optional unless there is more than one reset button
- ❖ Value attribute can be used to show text in the reset button

```
<input type="reset">
```

Reset

23

INPUT ELEMENT – SUBMIT TYPE

- ❖ Displays a push button with the preset function of sending the entered data in the form to the server for processing
- ❖ Value attribute defines what text to show in the button
- ❖ Name and value attributes optional unless there is more than one submit button

```
<input type="submit" value="Click here to download" name="download">
```

Click here to download

24

INPUT ELEMENT – FILE TYPE

- ❖ Used to upload a file to the web server

```
<form action="/client.php" method="post" enctype="multipart/form-data">
  <input type="file" name="classlist" size="28" />
</form>
```

- ❖ *Size* indicates the character width of the text input field in Firefox.

Choose File classlist.txt

25

SELECT FORM ELEMENT

- ❖ Has two parts: `<select>` tag and `<option>` tag
- `<option>` tags are nested within the `<select>` tag.
- ❖ Shows a list of options as a drop-down (or pull-down) menu.
- ❖ Attributes of `<select>`
 - Number of options to show a user use `size="n"`
 - Multiple selections use **multiple** using ^CTRL the user can select more than one option.
- ❖ Attributes of `<option>`
 - Preselect options with **selected** in the option tag
 - **Value** provides an overriding value to the server

27

SELECT

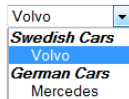
```
<select name="payment">
  <option selected
    value="credit">Credit Card
</option>
  <option value="debit">Bank Debit
</option>
  <option value="cheque">Cheque
</option>
</select>
```

28

SELECT OPTGROUP

- ❖ Items can be grouped into a separate submenu

```
<select>
  <optgroup label="Swedish Cars">
    <option value="volvo">Volvo</option>
  </optgroup>
  <optgroup label="German Cars">
    <option value="mercedes">Mercedes</option>
  </optgroup>
</select>
```



29

HTML SELECT EXAMPLE

A simple form that uses select

1. Favourite city:

2. Garden flowers: (pick multiple)

3. Cars Owned:

4. Age Group (set a default):

5. Favourite Movies:

[Page sample](#)

30

FIND THE ERRORS

Each of these markup examples contains an error. Write the proper syntax for these form controls.

1. `<input name="country" value="Your Country here">`
2. `<checkbox name="color" value="teal" >`
3. `<select name="popsicle" >`
`<option value="orange">`
`<option value="grape">`
`<option value="cherry">`
`</select>`
4. `<input type="password">`
5. `<textarea name="essay" height="6" width="100">Your`
`story.</textarea>`

31

DID YOU FIND ALL THE ERRORS

Each of these markup examples contains an error. Write the proper syntax for these form controls.

1. `<input name="country" value="Your Country here" type="text">`
2. `<checkbox name="color" value="teal" > This is not an element.`
 Should be `<input type="checkbox" value="teal">`
3. `<select name="popsicle" > Option is not an attribute!`
`<option value="orange">orange</option>`
`<option value="grape">grape</option>`
`<option value="cherry">cherry</option>`
`</select>`
4. `<input type="password" name="psswd">`
5. `<textarea name="essay" rows="6" cols="100">Your story.</textarea>`

32

DECIDING WHICH CONTROL TO USE?

If the answer is **input** don't forget to include the type attribute.

- Choose your astrological sign (there are 12) `<select>`
- Indicate whether you have a history of heart disease (yes or no) `<input type="radio">`
- Write-up a book review `<textarea>`
- Select your favourite ice cream flavours from a list of 8 `<input type="checkbox">`

33

FORM ACTION ATTRIBUTE

- ❖ The `<form>` tag uses an `action` attribute
 - Specifies the name of the script to run when the form's submit button is clicked
 - If no action attribute is provided, or the script does not exist, nothing will happen when submit is clicked
 - CGI scripts are usually organized into their own folder sometimes called script or cgi-bin
- ```
<form name="myform" action="script/runform.php">
```

34

## FORM METHOD ATTRIBUTE

- Method attribute controls how your browser sends data to the web server

– **Get** method is default. This sends the form data as a complete text appended to the URL. All the form data are shown in the URL separated by ampersands

<http://www.myscript.com/...cgi?name=John+Smith&email=jsmith@mail.com>

– **Post** method is **preferable**. This sends the form data in a separate data stream and is more flexible as most web servers truncate the amount of data they receive from the GET method

35

## FORM ENCTYPE ATTRIBUTE

- ❖ Specifies the format of the data when it is transferred from the web page to the script
- ❖ Default value is either 'multipart/form-data' (for type="file") or 'application/x-www-form-urlencoded' (all other types)
- ❖ Tells the server what type of data it should be expecting from the browser

36

## ORGANIZING FORM ELEMENTS

- You can group different form components into a related group called a *fieldset*
- Makes it easier for visitors to the page to understand the form and fill it out properly

- The legend tag is descriptive text

```
<fieldset>
```

```
<legend>Personal Information</legend>
```

```
... Rest of form elements
```

```
</fieldset>
```

37

## FORM LABEL

- Label tag enables CSS formatting to the form elements

```
<label for="fname">First Name: </label>
```

```
<input type="text" name="firstname"
size="15" id="fname">
```

38

## LAB 5 QUESTIONS?

- ❖ Following the correct process of the Lab 5 is important to ensure you are ready for the JavaScript assignment in Lab 6.
- ❖ Please email me if you have any questions for the lab.

39

## TITLE ATTRIBUTE

- ❖ Allows for tool-tips pop-up
- ❖ Provides meaningful explanation or a short description of an element

```
<input type="text" name="dept" title="Department Category" />
```

44

## BUTTON TAG

- ❖ User defined image buttons
- ❖ Define an **onclick** attribute and an image

```
<button onclick="script/mybutton.php">

 Click this blue button.
</button>
```

45

## HTML5

- ❖ New form elements and attributes in HTML5
- ❖ Not supported by all browsers, however the features degrade nicely (e.g. they become regular text boxes)

```
<input type="tel"> for phone numbers
<input type="number"> numbers only
<input type="search"> search box (like text)
<input type="email"> email address
<input type="url"> for url addresses
<input type="date"> for date information
<input type="color"> for colour selection
```

46

## HTML5

- ❖ New form elements
  - `<progress>` shows progression along a set of tasks
  - `<meter>` a scalar measurement within a range
  - `<datalist>` for making options within a dropdown
  - `<keygen>` a control for key-pair generation
  - `<output>` shows the output of a calculation

47

## HTML5 - PROGRESS

- ❖ Show a progress bar to indicate progression along a sequence of tasks
- ❖ Use the **orient** property to specify horizontal (default) or vertical display
- ❖ Use JavaScript to dynamically update
- ❖ `<progress value="70" max="100">`

```
70 %</progress>
```

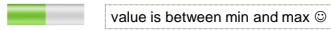


48

## HTML5 - METER

❖ The meter element is similar to progress but meter shows the status of a value within a range (ideally between a min and max)

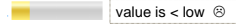
❖ `<meter min="200" max="500" value="350"></meter>`



❖ `<meter low="69" high="80" max="100" value="84"></meter>`



❖ `<meter low="69" high="80" min="50" max="100" value="60"></meter>`



49

## HTML5 - DATALIST

❖ Use datalist to define a predefined set of valid entries to a text box (not supported by Safari)

`<datalist id="soup">`

`<option value="chicken">`

`<option value="minestrone">`

`<option value="tomato">`

`<option value="miso">`

`</datalist>`

Enter your soup:

- minestrone
- miso

Enter your soup: `<input name="sel" list="soup">`

50

## HTML5 - KEYGEN

❖ The keygen form element creates a public key as part of HTML form to be used for web-based certificate management systems (encrypt the form information)

❖ Microsoft does not support this element

<http://support.microsoft.com/kb/190282>

❖ PHP script can access the local private key stored within the browser certificate area

51

## HTML5 - OUTPUT

❖ The output element represents the result of a calculation

❖ `<form oninput=`

`"result.value=parseInt(a.value) + parseInt(b.value)">0`

`<input type="range" name="b" value="50">100`

`+`

`<input type="number" name="a" value="10">`

`= <output name="result"></output>`

`</form>`

0  100 +  = 34

52

## HTML5

❖ Attributes for the `<input>` element

- placeholder – shows a light coloured hint of input
- autofocus – focuses on element when page is loaded
- required – if present, the element cannot be left blank by user when submit is clicked
- pattern – the input data must match the provided regular expression
- autocomplete – default is on, set it to off for elements you don't want autofilled (credit card)

53

## HTML5

❖ The pattern attribute specifies a regular expression:

- `\d` indicates a single number 0 to 9 same as `[0-9]`
- `\d{3}` indicates exactly three digits
- `\d{4,7}` indicates min 4 max 7 digits are needed

❖ Use **title** attribute to let users know input

❖ Special characters like hyphen, (, and ) are enclosed by square brackets preceded by \

- `[\-]` or `[\(]` or `[\)]`

54



## HTML5

❖ `<input type="text" placeholder="first name" required>`

❖ `<input type="number" value="1" min="1" max="10">`

❖ `<input type="tel" pattern="\d{7}">`

55

## HTML5

❖ `<input type="date">` works in Chrome, Opera, Firefox but not yet in IE

56

## HTML5

❖ `<input type="range" name="skill" min="1" max="10" value="1">`

What skill level?

```
<form oninput="amount.value=rangeInput.value">
 <input type="range" id="rangeInput"
 name="rangeInput" min="0" max="100">
 <output name="amount" for="rangeInput">0
</output>
</form>
```

57

## HTML5

❖ Other date and time variations ... implemented in Opera browser only

❖ `<input type="datetime">`

❖ `<input type="month">`

❖ `<input type="week">`

❖ `<input type="time">`

❖ `<input type="datetime-local">`

58

## HTML5

❖ `<input type="email">`

- If attribute **multiple** is specified, two or more email addresses can be entered separated by commas
- User must enter an address having an **@** symbol

59

## HTML5

❖ `<input type="color">`

❖ `<input type="range" min="-100" max="100" value="0" step="10" name="power" list="powers">`  
`<datalist id="powers">`  
`<option value="0">` `<option value="-30">`  
`<option value="30">` `<option value="+50">`  
`</datalist>`

60

## SPELL CHECKING

❖ Text fields can be spell checked using HTML 5

spellcheck attribute

❖ Based on the `lang` attribute in `<html>`

`<html lang="en">` means use English spelling

`<textarea spellcheck="true"> </textarea>`

`<input type="text" spellcheck="true">`

61

## CSS3

❖ pseudoclasses `:invalid` and `:required` can be styled in CSS

```
:invalid {
 box-shadow: 0 0 5px rgba(255, 0, 0, .8);
 border-color: red;
}
```

```
:required {
 box-shadow: 0 0 5px rgba(0, 0, 255, .5);
 border-color: blue;
}
```

62

## FORM DESIGN GUIDELINES

- Form field descriptions on the left, alignment
- Label input boxes with clear instructions
- Group related form items into a fieldset
- Control user's entries with radio buttons, checkboxes and selection lists whenever possible
- Let users know the correct format to enter date fields (yyyy/mm/dd)
- Use selection lists for many possible options
- Use radio buttons for five or fewer options

63

## FORM ANOMALIES

❖ Can a form prevent users hitting submit twice?

- Can't -- unless you use cookies or use a unique ID in the form

❖ Some form elements can be easily styled with CSS

(form, fieldset, label, input, output) – other form elements not so easily (date, select, option, progress, meter, legend)

64

## GET INTO GROUPS SUDO CODE TIME

1. Blog posting that requires at least your name (two fields), email, comment and rating.

2. Sign-up for an information session requires you name, email, phone number and for you to select a session from 6 (time and date).

3. Booking a facility rental, name (two fields), email, phone number, purpose, number of guests and the type of room (choose from 6 options)

4. Buying something online. Define the fields you would need.

5. Submitting an application for a job. Define the fields you would need.

65

## HTML FORMS

❖ Mozilla Developer guide HTML forms

<https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Forms>

66