

# **Interactivity**

---

Web Design in a Nutshell, Third Edition

Chapters 15, 26, 27 and 35

---

# Interacting with your users

---

- Most basic form of interactivity is links to other pages, web sites and scripts.
  - HTML forms are the interface you create for users to enter data and send it to you, such as a search or a contact form.
  - Javascript is the scripting language for browsers that offers unlimited possibilities.
-

# Interacting with your users

---

- PDF files can have interactive form and javascript elements to allow users, for instance, to send you registration data.
  - Flash is a technology that has evolved from offering animated pretty pictures to being a platform for building applications and more.
  - Server-side technologies work with each of these to offer much more.
-

# Understanding How Forms Work

---

- Forms let you build interactive Web pages that collect information from a user and process it on the Web server.
  - Common examples of HTML forms are search, contact information, checkout forms at e-commerce sites, and login screens.
  - The data is usually processed by applications that reside on the Web server.
-

# A Basic Form

---

Please complete the form below. Mandatory fields marked \*

## Delivery Details

Name \*

Address \*

Town/City

County \*

Postcode \*

Is this address also your invoice address? \*

Yes

No

## Other Information

Date of Birth \*

Sex \*

Which of the following sports do you enjoy?

Football

# Basic Form Elements

---

Please complete the form below. Mandatory fields marked \*

## Delivery Details

Name \*

Address \*

Town/City

County \*

Postcode \*

Is this address also your invoice address? \*

Yes

No

Text Input Boxes

Radio Buttons

# Basic Form Elements

## Other Information

Date of Birth \*

Sex \*

Which of the following sports do you enjoy?

- Football
- Golf
- Rugby
- Tennis
- Basketball
- Boxing

Comments

Submit order

Select Lists

Checkboxes

Text area

Submit Button

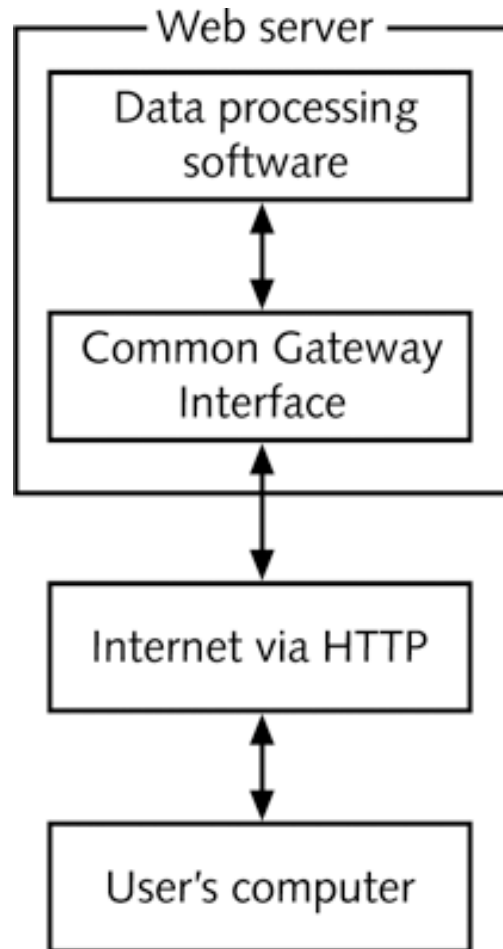
## Common Elements for Every Form

---

- Opening and closing form tags, including either an Action (script on the server) or javascript function (on submit) or both.
  - One or more user input elements — text, radio, checkbox, submit or other inputs, text areas, or select lists.
-

# Form Workflow

---



---

**Figure 10-2** Common Gateway Interface architecture

# Forms Syntax

---

- Five basic form elements are commonly supported by all web browsers:
  - `<form>`
    - `<input>`
    - `<select>`
    - `<option>`
    - `<textarea>`
-

# Using the `<form>` element

---

- The `<form>` element is the container for creating a form.
  - A variety of attributes describe how the form data will be handled.
-

# Form tag

---

**Table 10-1** Form attributes

Attribute	Description
action	The URL of the application that processes the form data; this URL usually points to a CGI script file
enctype	The content type used to submit the form to the server (when the value of the method is "post"); most forms do not need this attribute
method	Specifies the HTTP method used to submit the form data; the default value is "get" <ul style="list-style-type: none"><li>• get—The form data is appended to the URL specified in the action attribute</li><li>• post—The form data is sent to the server as a separate message</li></ul>
accept	A comma-separated list of content types that a server processing this form can handle correctly; most forms do not need this attribute
accept-charset	A list of allowed character sets for input data that is accepted by the server processing this form; most forms do not need this attribute

---

# Using the <form> element

---

- The following code shows a typical <form> element:

```
<form action="http://www.website.com/  
cgi\_bin/script.cgi" method="post">
```

---

# Input Objects

---

- The `<input>` element defines many of the form input object types
  - The `type` attribute specifies the type of input object
-

**Table 10-2** <input> element types

Type Attribute Value	Description
text	Creates a text entry field that lets the user enter a single word or a line of text; this is the default object type
password	Creates the same type of text entry field created by the value "text", but the user entry is masked by asterisks
check box	Provides on/off toggles that the user selects. Check boxes are best used with multiple-choice questions. Multiple check boxes can contain the same name, letting you group them together so that users can select multiple values for the same property.
radio	Lets a user choose one value from a range of values; when radio buttons are grouped together with the same name, only one choice can be selected
submit	Sends the form data to the server using the transmission method specified in the <form> element; every form needs a submit button
reset	Clears the form of any user-entered data and returns it to its original state
hidden	Adds a control that is not displayed in the browser; the hidden type is useful for sending additional information with the form data that may be needed for processing
image	Adds a graphic button to the form, rather than the default button
button	Creates a button that has no default behavior. The button's function is usually defined by a script. When the user pushes the button, the script function is triggered.
file	Lets the user select a file that is submitted with the form

# Text Input Boxes

---

Please complete the form below. Mandatory fields marked \*

## Delivery Details

Name *	<input type="text"/>
Address *	<input type="text"/>
Town/City	<input type="text"/>
County *	<input type="text"/>
Postcode *	<input type="text"/>

## Attributes

Name

Value (Default)

Size

Max Size

```
<label for="postcode">Postcode <em>*</em></  
label> <input type="text" name="postcode" size="20"  
maxlength="35" value="Your Zip Code">
```

---

# Creating Input Text Boxes

---

- The text box is the most commonly used form element.

```
<input type="text" name="firstname" size="20"  
maxlength="35" value="First Name">
```

- The Size attribute is generally no longer used as you get more accurate results in using CSS width and height.
  - The value attribute is the default and can be overridden by the user who simply types into the field.
  - Maxlength is the maximum number of characters.
-

# Radio Buttons and Check Boxes

---

Please complete the form below. Mandatory fields marked \*

## Delivery Details

Is this address also your invoice address? \*

- Yes  
 No

## Other Information

Which of the following sports do you enjoy?

- Football  
 Golf  
 Rugby  
 Tennis  
 Basketball  
 Boxing

## Attributes

Name

Value (Default)

---

# Creating Check Boxes

---

- Check boxes are an on/off toggle that the user can select.

```
<input type="checkbox" name="species"  
value="smbass" /> Smallmouth Bass <input  
type="checkbox" name="species" value="smbass" />  
Largemouth Bass
```

- With Check boxes, a field can have multiple values; checked values are sent to the server while unchecked values are not.
-

# Creating Radio Buttons

---

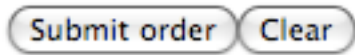
- Radio buttons are like checkboxes, but only one selection is allowed.

```
<input type="radio" name="list" value="yes"
checked="checked" /> Yes <input type="radio"
name="list" value="no" /> No
```

- With Radio buttons, a field can have only one value
  - In both Radio buttons and Checkboxes, the checked attribute determines whether the object is selected
-

# Submit and Reset Buttons

---



## Attributes

Name

Value (What shows up in the button)

---

# Creating Submit & Reset Buttons

---

- The submit and reset buttons let the user choose whether to send the form data or start over.

```
<input type="submit" value="Submit your answers" />
```

```
<input type="reset" value="Clear the form" />
```

- A submit button sends data to the server while a reset button simply clears all the field values within the current form
  - The value attribute is what is displayed in the browser.
-

# Submit Button with an Image

---



**Figure 10-8** Using an image for the submit button

---

# Creating an Image for the Submit Button

---

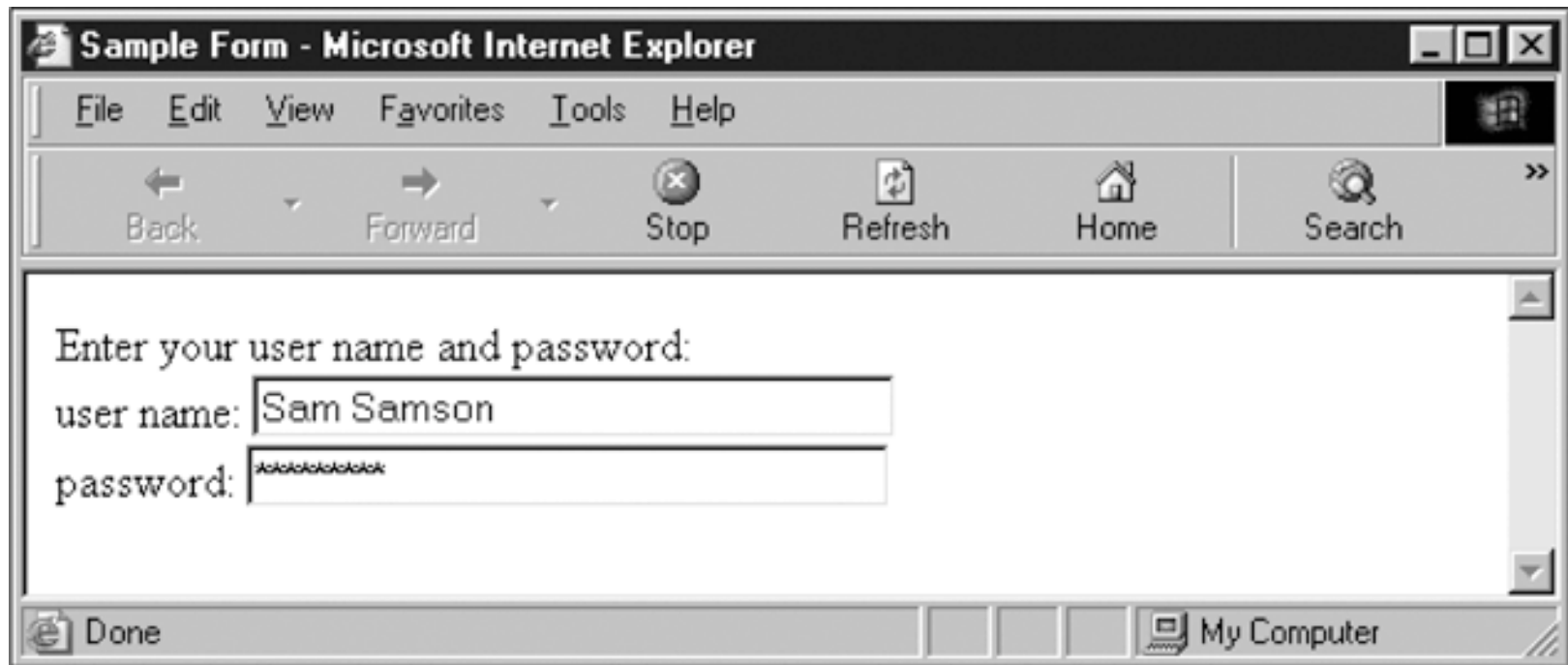
- You can choose an image file and use it instead of the default submit button.

```
<input type="image" src="submit.gif" alt="submit button" />
```

- The src attribute is the URL for the image to be displayed
-

# Password Input

---



**Figure 10-10** Password type input

---

# Creating a Password Entry Field

---

- The password input box works like the text input, but the entered text is hidden by bullets or asterisks.

password: `<input type="password" size="30" />`

- Although someone looking over your shoulder can't see what you type, hackers monitoring your ethernet connection can perhaps determine what you are sending to the server.
  - To protect passwords and login info, your server should use https or Secure Sockets Layer encryption.
-

# Hidden and File input types

---

- The hidden input type allows you to insert a value to be passed to the server that won't be displayed on a page.

```
<input type="hidden" value="You won't see this" />
```

- The file input type allows you to upload a file when you submit the form. It is displayed with a browse button allowing you to choose a file for uploading.

```
<input type="file" size="28" />
```

---

# Select Lists

---

**Other Information**

Date of Birth \*    1    Jan    1979

Sex \*    Female  
Male

Elements

Select

Option

Attributes

Select (name)

Value (option)

---

# Using the <select> Element

---

- The <select> element lets you create a list box or scrollable list of selectable options. In most browsers, this appears as a popup menu.

```
<select name="boats">  
<option>Canoe</option>  
<option>Jon Boat</option>  
<option>Kayak</option>  
<option>Bass Boat</option>  
<option>Family Boat</option>  
</select>
```

---

# Using the `<select>` Element

---

- You can choose to let the user pick multiple values from the list by adding the `multiple` attribute.

```
<select name="snacks" multiple="multiple" size="6">  
<option>Potato Chips</option>  
<option>Popcorn</option>  
<option>Peanuts</option>  
<option>Pretzels</option>  
<option>Nachos</option>  
<option>Pizza</option>  
<option>Fries</option>  
</select>
```

---

# Text Area

---

Other Information

Comments

Submit order

Attributes

Name

Rows

---

# Using the `<textarea>` Element

---

- The `<textarea>` element lets you create a larger text area with more than a single line for user input.

```
<p>Briefly tell us your favorite fish story:</p><textarea  
name="fishstory" rows="5" cols="30">Enter your story  
here...</textarea>
```

- Rows are the number of lines that are displayed
  - Cols are the number of characters wide that are displayed
  - CSS provides many more accurate options, including height and width
-

# Fieldsets and Legends

---

Please complete the form below. Mandatory fields marked \*

## Delivery Details

1. Name \*

2. Address \*

3. Town/City

4. County \*

5. Postcode \*

6. Is this address also your invoice address? \*

Yes  No

Submit order

---

# Accessibility for Forms — Input Groups

---

- The <fieldset> element is used to create a logical group of form controls.
- The <legend> element is a description of the enclosed fields that may be useful for non-visual browsers

```
<fieldset><legend><b>Select the species you prefer to fish:</b></legend><input type="checkbox" name="species" value="smbass"> Smallmouth Bass<input type="checkbox" name="species" value="lgbass"> Largemouth Bass <br /><input type="checkbox" name="species" value="pike"> Pike</fieldset>
```

---

# Labels and IDs

---

Please complete the form below. Mandatory fields marked \*

## Delivery Details

1. Name \*

2. Address \*

3. Town/City

4. County \*

5. Postcode \*

6. Is this address also your invoice address? \*

Yes  No

Submit order

---

# Accessibility for Forms – Labels and IDs

---

- The `<label>` element is used to associate descriptive text with a form field, providing context for users accessing the form with a speech-based browser.

```
<label>Login: <input type="text" name="login" /></label>
```

- An `id` value specified in the input form allows you to associate which form element the current label is for.

```
<label for="log">Login account:</label> <input type="text" name="login" id="log" />
```

- Labels and IDs provide more CSS selectors for making work better for users
-

# Accessibility — accesskey, tabindex and title

---

- The **accesskey** attribute specifies a character to be used as a keyboard shortcut to an element.
  - The **tabindex** attribute allows you to rearrange the tab order of forms. By default when you tab between fields, they go in the order they were written.
  - The **title** attribute, similar to how it works with links or img elements, allows you to offer descriptive text about a form object. Speech browsers may speak the title when the form field is brought into focus; Visual browsers may render titles as "tool tips" that appear when the pointer pauses over the control.
-

# Considerations

---

- Generally, You will need to work with some type of server-based software program to process the data from your form
  - These software programs, commonly called CGI scripts, should include code for handling each object of a form, i.e. every field.
  - For example, if you want to save someone's name and address into a database, you need fields that the user can type those things into, and a script that takes that user input and inserts it into the database.
-

# Using Server Scripts

---

- Using server-side scripts, CGIs can collect data sent by a user's browser via HTTP and transfer it to a variety of programs including spreadsheets, databases, e-mail programs, and word processors.
  - Common CGI languages are:
    - .php — Personal Home Page Tools (PHP Tools)
    - .asp — Active Server Pages (Windows)
    - .jsp — Java Server Pages (Unix)
    - .pl — Perl (Unix)
-

# Common Concepts

---

- Most server-side technologies use a database, such as MySQL, Oracle or Microsoft SQL Server to store data.
  - All support the concept of variables. A variable can be a value submitted by a user via a form, a value that's part of the request, such as the date, or just something the developer made up like a random number.
  - All support functions, reusable bits of code that can be applied to any variable or set of variables, returning a result, such as one that formats a date a certain way.
-

# HTTP Cookies

---

- Snippets of text sent by the server to the user's browser and then returned each time by the browser each time a user visits the site until the cookie expires.
- They are used for authenticating, tracking and maintaining specific info about users, such as your login info on your Wordpress sites, items in your Amazon.com shopping cart, and your preferences at Google and Yahoo!
- This is my USAToday cookie:

`.usatoday.com TRUE / FALSE 1767160800 zagCookie`

---

# Javascript

---

- Originally created by Brendan Eich for Netscape 2.0 in 1995
  - It is a client-side scripting language for offering a rich experience for users.
  - While all modern browsers support it to one extent or another, it requires diligence to make sure it works with the specific browsers you are targeting.
  - Users can turn off Javascript so you should make sure that your site works with or without it.
-

# Ajax and Web 2.0

---

- AJAX stands for Asynchronous JavaScript and XML.  
The term was first used in February 2005
  - It's a technique that uses Javascript to get data by loading sections of a page rather than reloading a whole page, giving the illusion that things are faster and more interactive.
  - Web 2.0 is a concept involving the mashup of social networking sites like Facebook with other sites, such as Google Maps, to create a third, different site.
-