

# **WEB APPLICATIONS**

## **LECTURE 3A**

### **HTML FORMS**

# FORMS

- Suppose we want to allow people viewing our web site to enter information (e.g. their names, their email addresses etc.).
- *Forms* make it possible to collect and process user input, and formulate personal replies (among other things).
- Lots of applications e.g.
  - ☐ Online Shopping
  - ☐ Web based surveys
  - ☐ etc.



The screenshot shows the Amazon.com sign-in interface. At the top left is the Amazon logo and 'amazon.com'. At the top right are links for 'Your Account' and 'Help'. The main heading is 'Sign In'. Below it is the question 'What is your e-mail address?'. There is a text input field preceded by 'My e-mail address is:'. Below that is the question 'Do you have an Amazon.com password?'. There are two radio button options: 'No, I am a new customer.' and 'Yes, I have a password:'. The 'Yes' option is selected. To the right of the 'Yes' option is another text input field. Below this is a link 'Forgot your password?'. At the bottom is a yellow button with the text 'Sign in using our secure server' and a play icon.

# FORMS

- **Two aspects of using forms**

- ☐ (a) collecting input
- ☐ (b) processing input

- **(a) is done with standard HTML but**

- **(b) is a bit trickier and is usually accomplished by writing programs which run on the server, ie PHP**

# HTML Forms

- Textfield
- Hidden field
- Password field
- Fieldset
- Label
- Submit / Reset Button
- Radio Button
- Checkboxes

- Dropdown menu
- Scrolling menu
- Listbox selection
- Optgroup
- Textarea
- File Uploading
- Tab Order



# EXAMPLE - INPUT FORM



The image shows a screenshot of a web browser window displaying a user registration form. The browser's address bar shows the URL `http://localhost:8080/register.html`. The form is titled "USER REGISTRATION" and includes a note that asterisks (\*) denote required fields. The form contains several input fields: "First Name\*", "Last Name\*", "E-Mail\*", "Zip Code\*", "User Name\*", "Password\*", and "Confirm Password\*". Below these fields, there is a section for music preferences with checkboxes for "Rock", "Pop", "Bluegrass", "Blues", "Jazz", and "Country". At the bottom, there is a question about receiving e-mail notifications with radio buttons for "Yes" and "No". The form ends with "Submit" and "Reset" buttons. The browser's status bar at the bottom indicates "Document Done".

**USER REGISTRATION**  
\* Required Fields

**First Name\***

**Last Name\***

**E-Mail\***

**Zip Code\***

**User Name\***

**Password\***

**Confirm Password\***

**What music are you interested in?**  
☐ Rock   ☐ Pop   ☐ Bluegrass  
☐ Blues   ☐ Jazz   ☐ Country

**Would you like to receive e-mail notifications on our special sales?**  
☒ Yes   ☐ No

# FORMS

- A form is an area that can contain form elements.
- Form elements allow the user to enter information (e.g. text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.)
- A form should have at least one control like a Submit button that activates the Action attribute of the form.
- Action attributes specifies the URL or JSP that will be called when the user clicks the Submit button
- A form is defined with the **<form> tag**.
- **<form action = ..... method = “post”>**
- **<input type = .....>**
- **<input type = .....>**
- **.....**
- **</form>**

# Form Template

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Form Template</title>
<meta http-equiv="content-type" content="text/html; charset=utf-8"/>
</head>
<body>
<form action = "formscript.php" method = "post">
Put your textboxes, drop down menus, radio buttons etc. in here.
</form>
</body>
</html>
```



- `<form action = ..... method = "post">`
- `<input type = .....>`
- `<input type = .....>`

# INPUT TAG

Most commonly used form tag is the **<input>** tag.  
Describes the input type

## Attributes of <input> tag

- **type** - kind of input field it will be.
  - **text, hidden, password, and submit or reset.**
- **name** - Assigns a name to the given field so that you can reference it later.
- **size** - Sets the horizontal width of the field. The unit of measurement is in blank spaces.
- **maxlength** - Max number of characters that can be entered.
- **value** – specifies value of data e.g. “ ”, or any default value e.g. xyz

# INPUT TAG - TEXT FIELDS

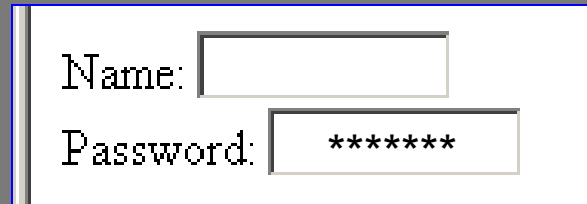
Text fields are used when you want the user to type letters, numbers, etc. in a form.

```
<form method="post" action="mailto:youremail@email.com">
```

```
Name: <input type="text" size="10" maxlength="40" name="name"
      value = " "> <br >
```

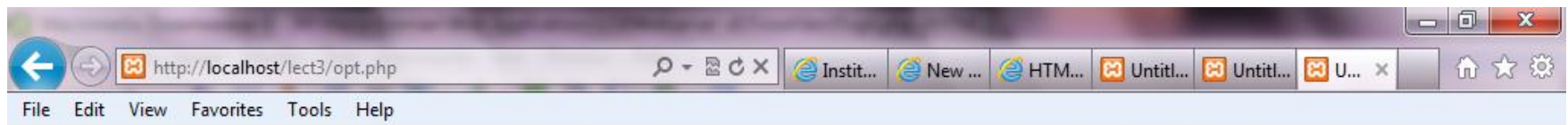
```
Password: <input type="password" size="10" maxlength = "10" name
="password"> <br>
```

```
</form>
```



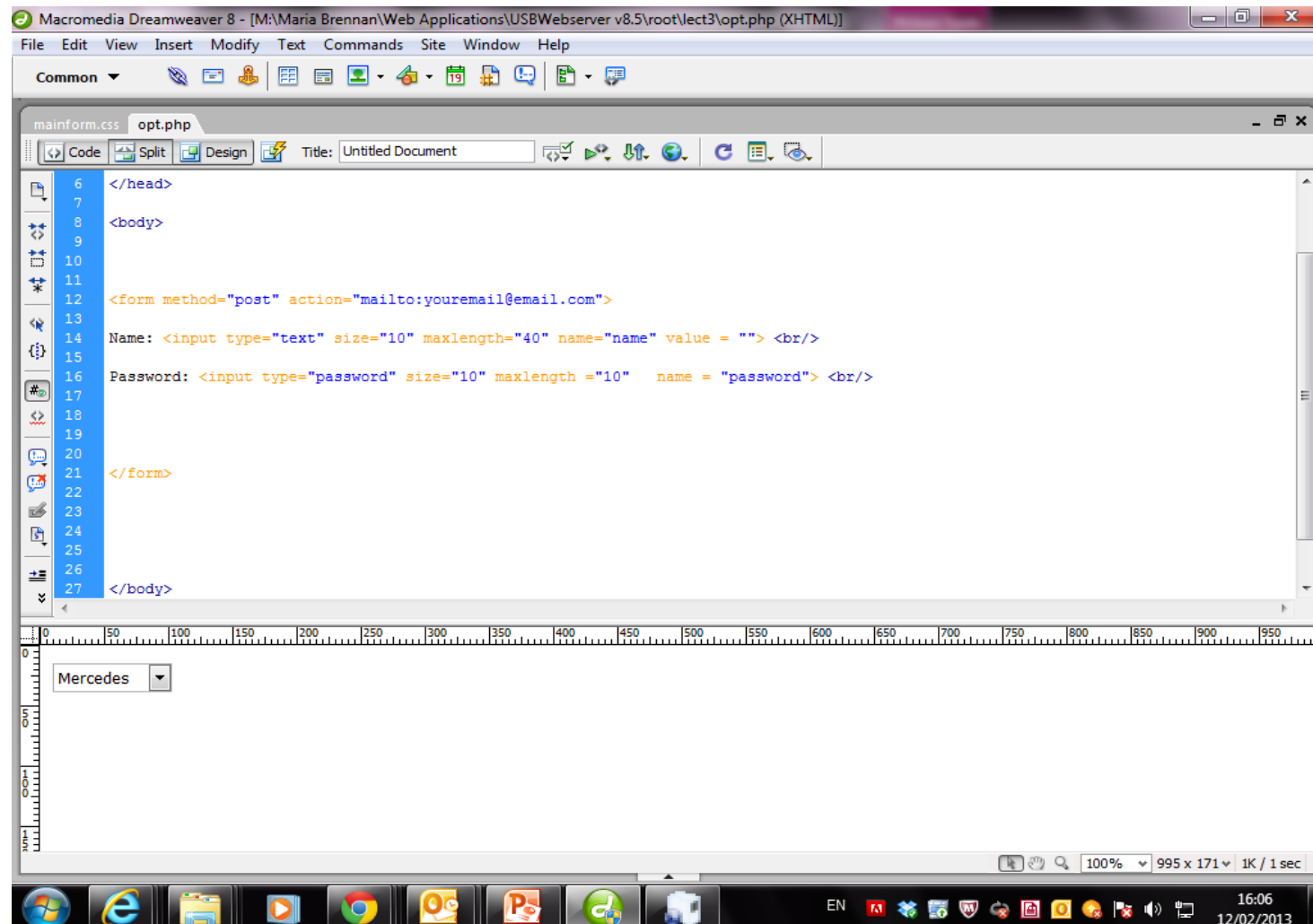
Name:

Password:



Name:

Password:



# INPUT TAG - TEXT FIELDS

- Every form should have at least a **SUBMIT** button
- May also have a **RESET** button or user named button also.
- Submit button should be the last item of your form and have its
- *name* attribute set to "**Send**" or "**Submit**".

# ATTRIBUTES OF BUTTONS

- **Type** – identifies the type of button
- **Submit** button – activates the Action attribute of the form when clicked
- **Reset** attribute resets all controls on the form to the default values when it's clicked
- Attribute of type **button** creates a Javascript button which can activate a Javascript OnClick() method

# INPUT TAG - BUTTONS

## Examples of buttons

**<input type = “submit” value = “Submit”>**

**<input type = “reset” value = “Reset”>**

**<input type = “button” value = “Enter data”>**

# INPUT TAG - SUBMIT BUTTON

Now add the **submit** functionality to your form.

Must also **add a destination for information on the form** and specify how you want it to travel to that place.

Add the following attributes to your `<form>` does this.

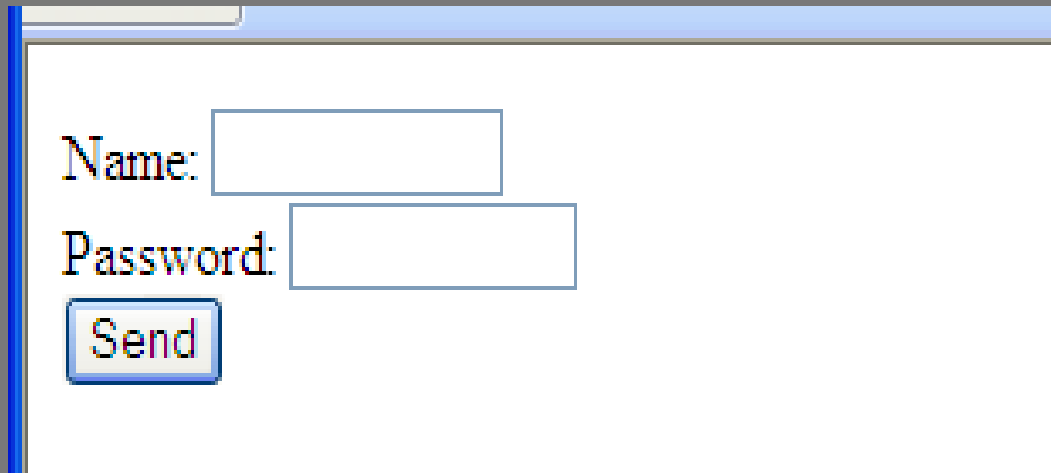
***action*** - Specifies the URL to send the data to ...a fake email address in this example

***method*** – use *GET* or *POST* methods

**POST** sends the data without displaying any of the information to the visitor.

**GET** sends the data and displays the information

# HTML FORM EMAIL CODE

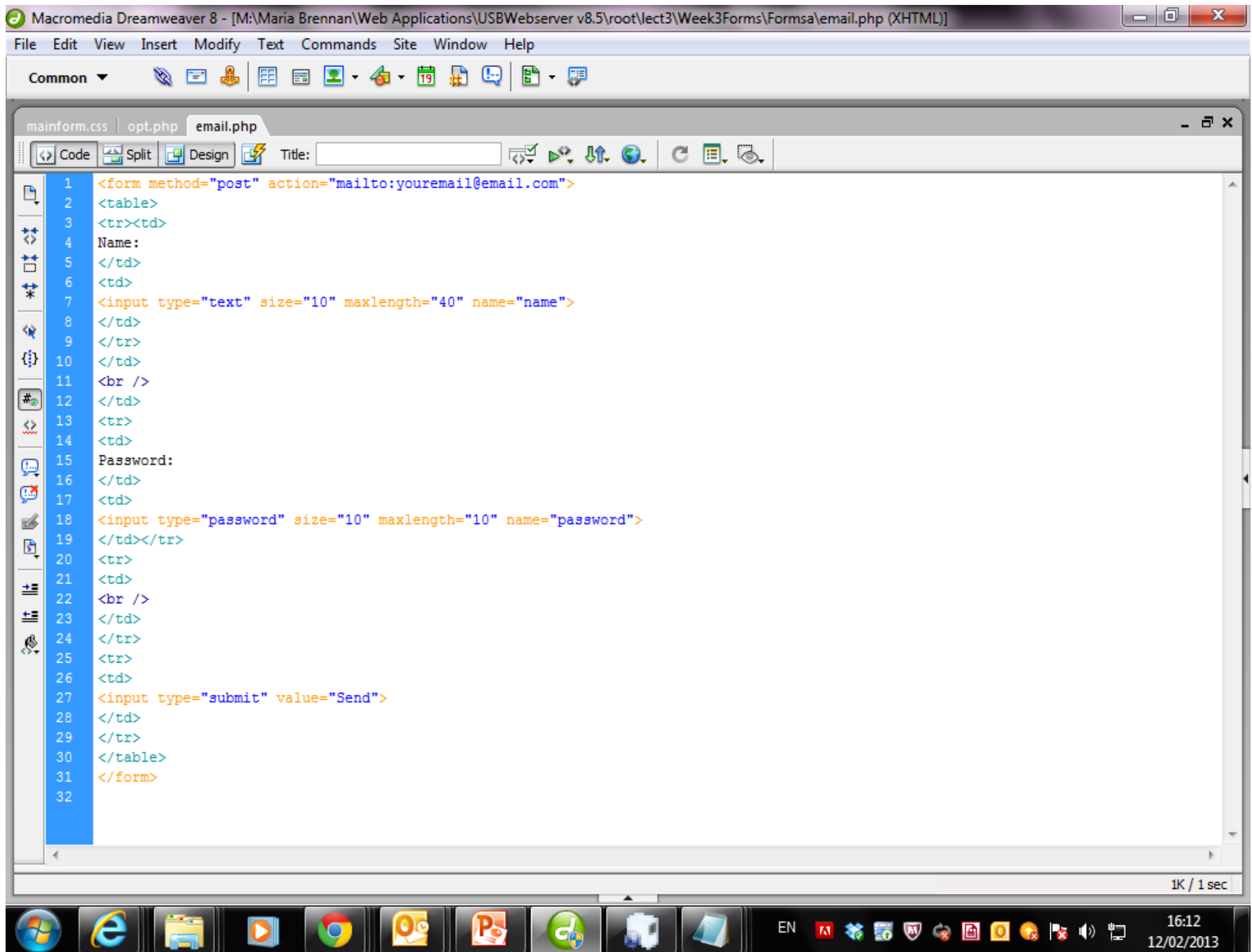


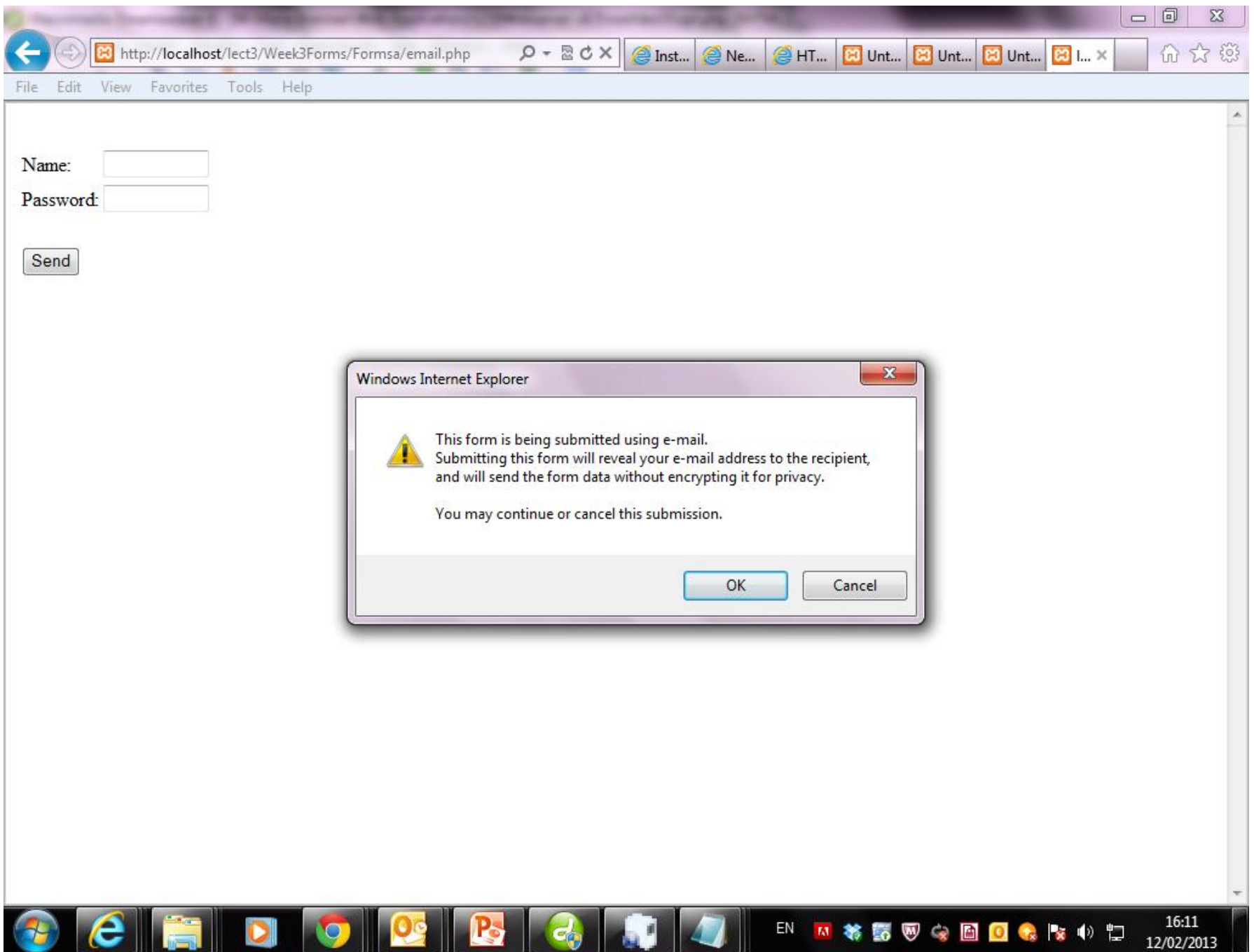
Name:

Password:

**Simply change the email address to your own and you will have set up your first functional form!**







# INPUT TAG - RADIO BUTTONS

Radio Buttons are used when you want the user to select one of a **limited number of choices**

Radio buttons are a popular form of interaction ...on **quizzes, questionnaires**, and other web sites that give the user a multiple choice.

## Attributes of the radio button.

- **type** – “radio” for radio button
- **name** - defines which **set** of radio buttons that it is a part of
- **value** - specifies what will be sent if the user chooses this radio button.
- Only one value will be sent for a given group of radio buttons (see *name* for more information).
- **checked** – selects a control

Example has 2 groups of radio buttons: **shade** and **size**

Note that only one option can be chosen.

# INPUT TAG - RADIO BUTTONS

What kind of shirt are you wearing?

Shade: ☐ Dark ☐ Light

Size: ☐ Small ☐ Medium ☐ Large

Email Myself

# Input Tag - Radio Buttons

```
<form method="post" action="mailto:youremail@email.com">
```

What kind of shirt are you wearing?

```
<br />
```

Shade:

```
<input type="radio" name="shade" value="dark">
```

Dark

```
<input type="radio" name="shade" value="light">
```

Light <br />

Size:

```
<input type="radio" name="size" value="small">
```

Small

```
<input type="radio" name="size" value="medium">
```

Medium

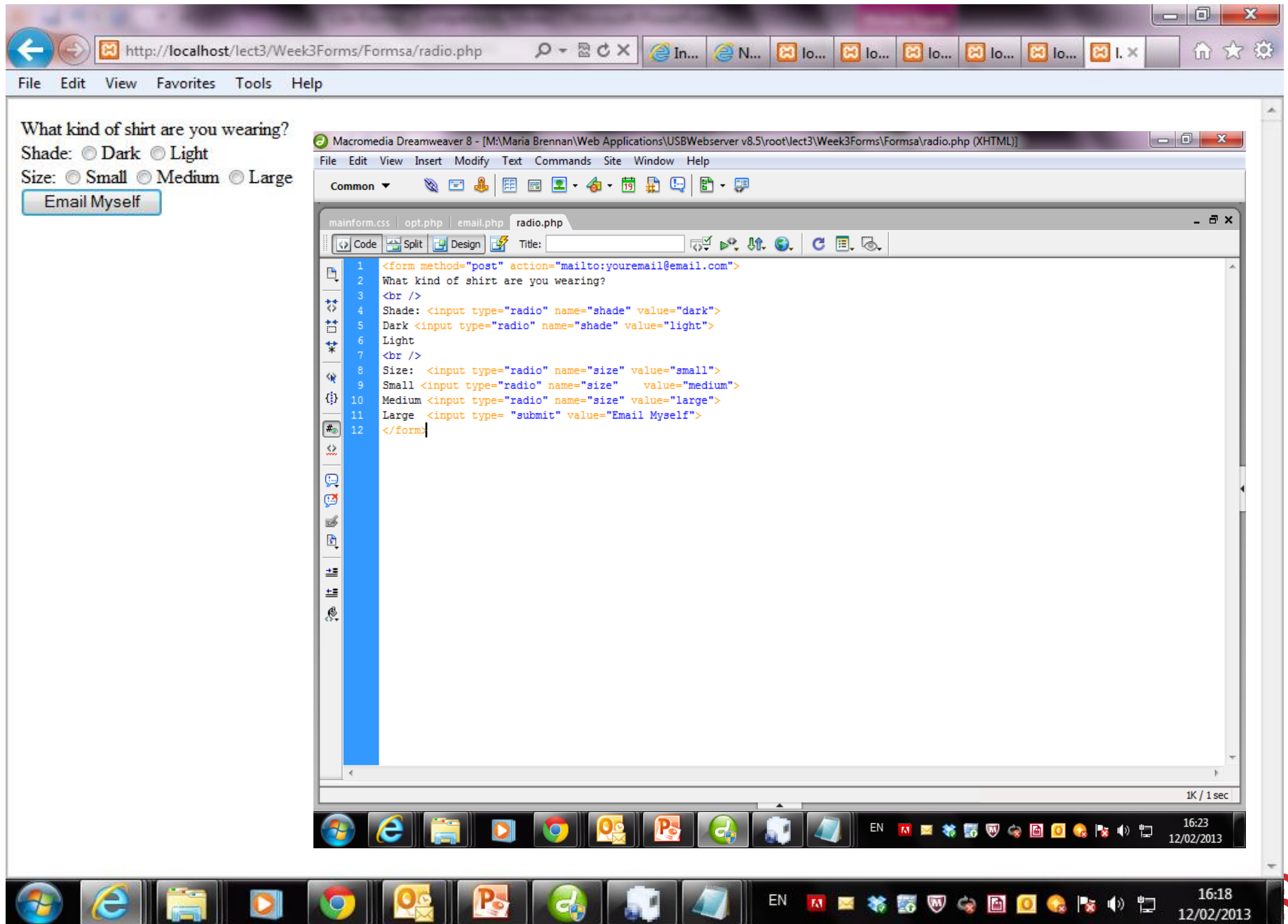
```
<input type="radio" name="size" value="large">
```

Large

```
<br />
```

```
<input type="submit" value="Email Myself">
```

```
</form>
```





# QUIZ

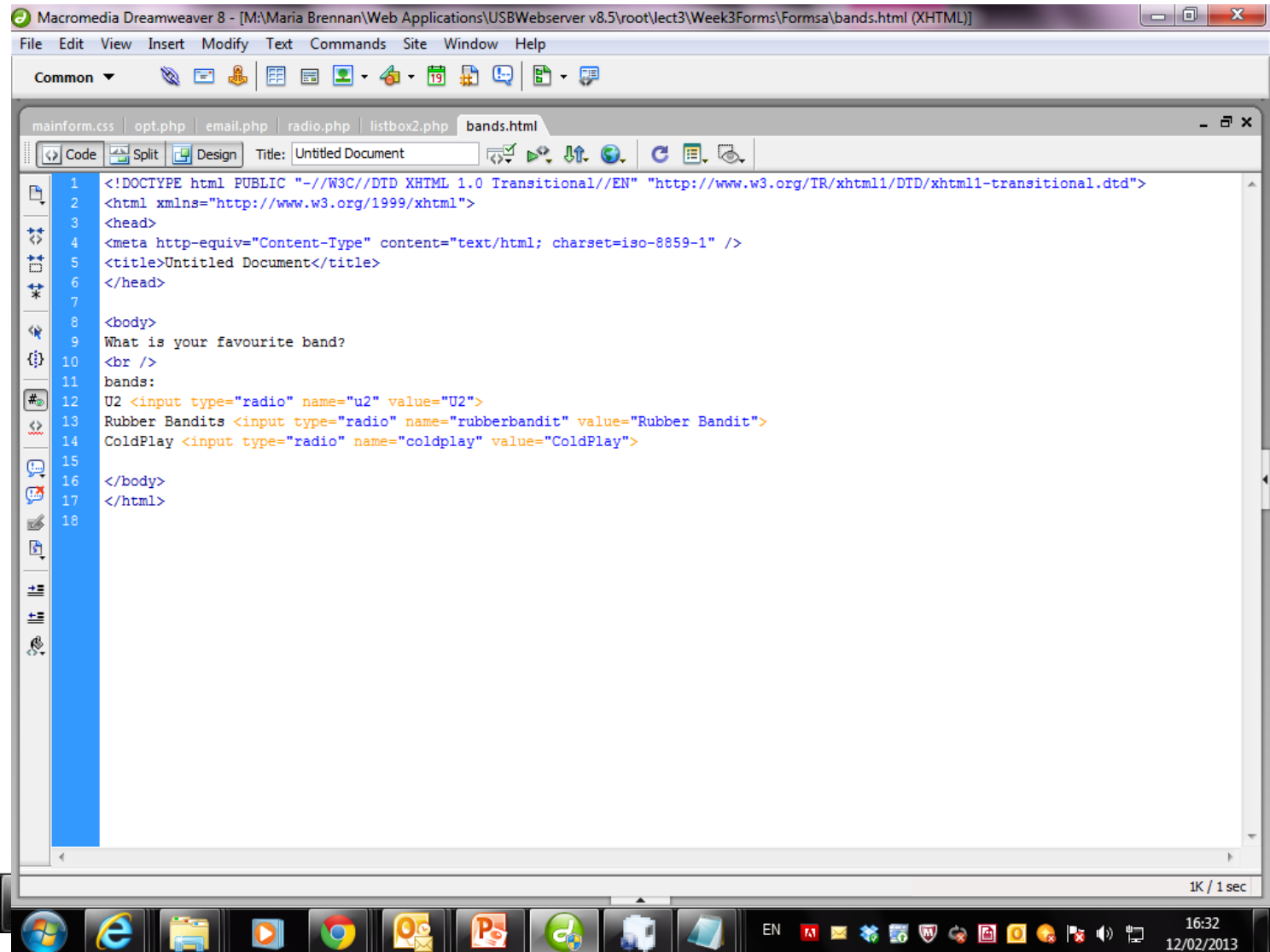
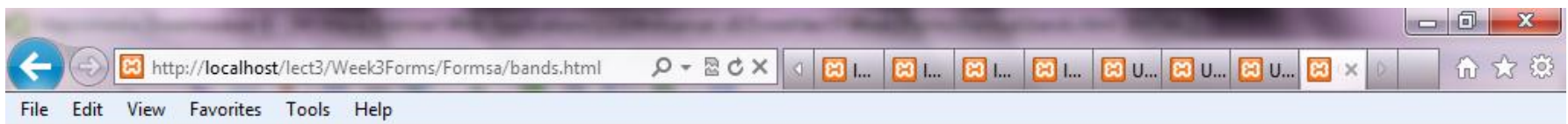
Create a form that contains radio buttons with the following details:

What is your favourite band?

Bands: ☒ U2 ☐ Rubberbandits ☐ Coldplay

Email Myself






# INPUT TAG - CHECKBOXES

Checkboxes are used when you want the user to select one or more options of a limited number of choices

The check box's **name** and **value** and **checked** attributes behave the same as a radio button.

Note: [ ] with checkboxes

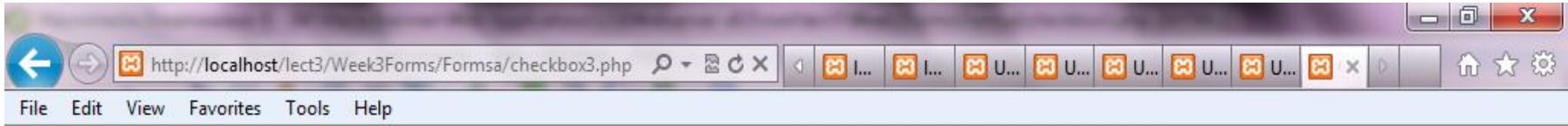


```
I have a bike: ☐  
I have a car: ☐  
I have an airplane: ☒
```

# QUIZ

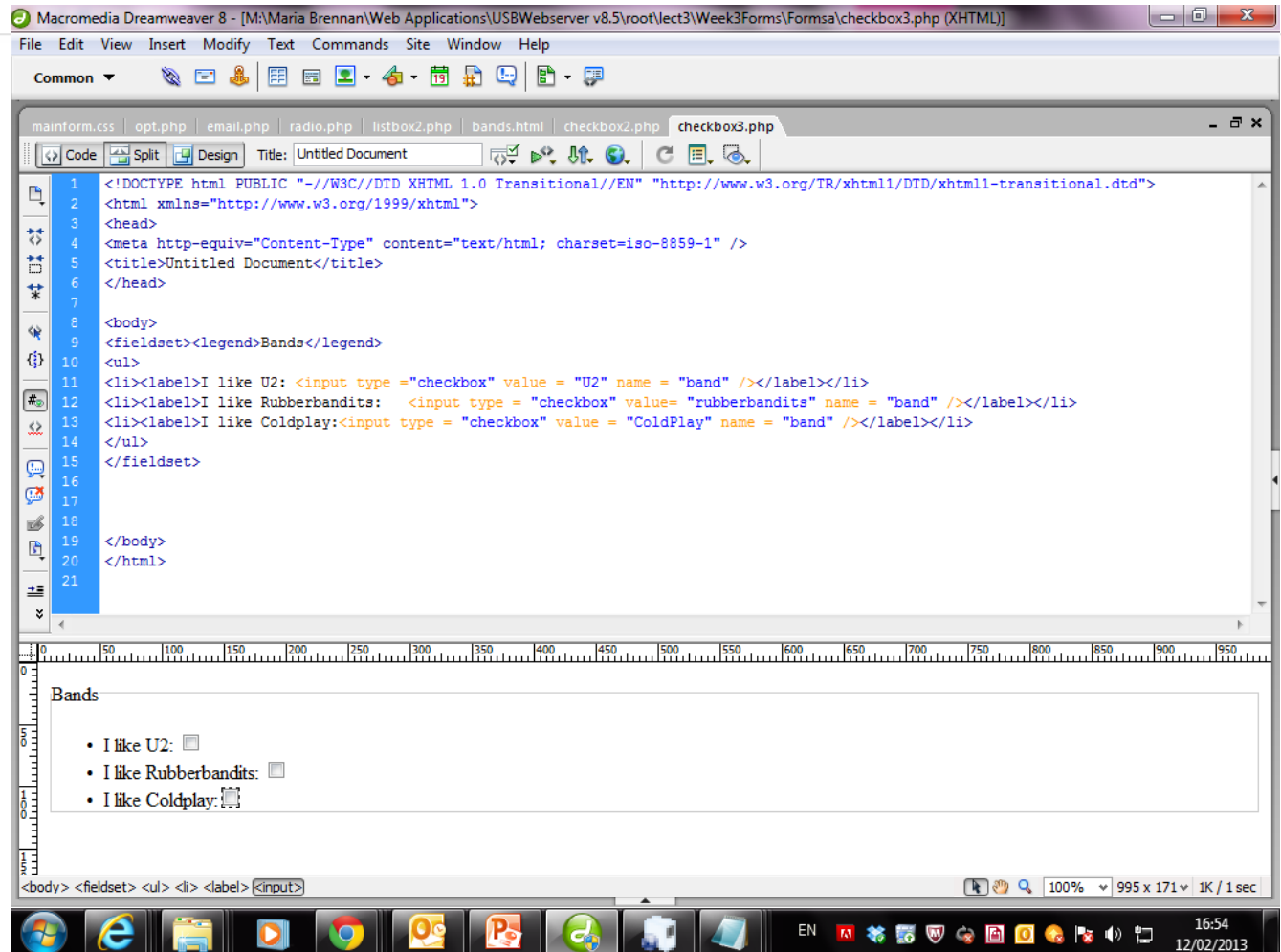
- Create a form that contains the following checkboxes:

- I like U2: ☐
- I like Rubberbandits: ☐
- I like Coldplay: ☐



## Bands

- I like U2: ☐
- I like Rubberbandits: ☐
- I like Coldplay: ☐



# DROPDOWN MENU

A drop-down box is a selectable list.

Drop down menus are created with the **<select>** and **<option>** tags.

**<select>** is the **list itself** and each **<option>** is an available choice for the user.

```
<form action="">
```

```
<select name="cars">
```

```
    <option value="volvo"> Volvo    </option>
```

```
    <option value="saab">  Saab    </option>
```

```
    <option value="fiat">   Fiat    </option>
```

```
    <option value="audi">  Audi     </option>
```

```
</select>
```

```
</form>
```

# DROPDOWN MENU



# DROPDOWN MENU

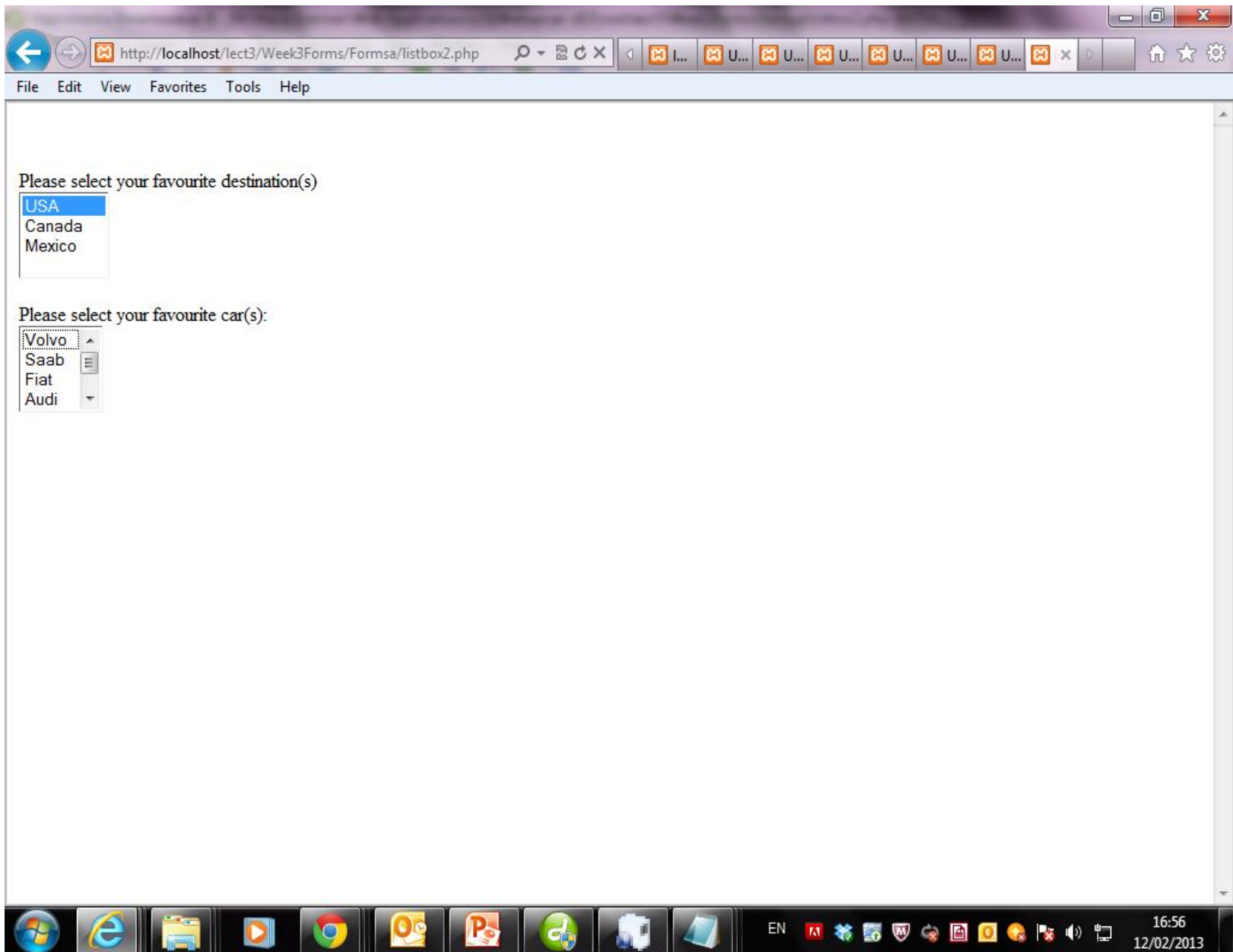
Another type of form, a **highlighted selection list**.

**size** attribute selects how many options will be shown at once before needing to scroll

**selected** option tells the browser which choice to select by default.

```
<form action="">
<select name="cars" size = "2">
    <option value="volvo">Volvo</option>
    <option value="saab" selected> Saab</option>
    <option value="fiat"> Fiat</option>
    <option value="audi">Audi</option>
</select>
</form>
```







# LIST BOX SELECTION

Convert a drop down menu to a list box by adding the **multiple** attribute

```
<form action="">
<select name="cars" multiple>
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
    <option value="fiat">Fiat</option>
    <option value="audi">Audi</option>

</select>
</form>
```

# LIST BOX SELECTION

If the user is allowed to select more than one item for the list then an array should be used:

```
<form action="">  
<select name="cars[]" multiple>  
<option value="volvo">Volvo</option>  
<option value="saab">Saab</option>  
<option value="fiat">Fiat</option>  
<option value="audi">Audi</option>  
</select>  
</form>
```

# QUIZ

- Create a form that contains the following dropdown menu:

What is your favourite band?

U2 ▼

U2

Rubberbandits

Coldplay

# TEXTAREA

A user can enter text in a text area.

Attributes;

**Rows** – number of visible lines in text are. A scroll bar appears if the number of lines in the text box exceeds this setting.

**Cols** – specifies width of text area

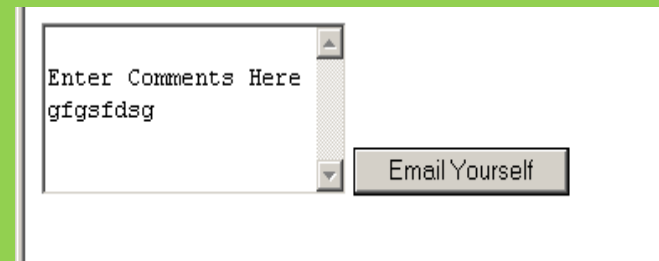
```
<form method="post" action="mailto:youremail@email.com">  
<textarea rows="5" cols="20" wrap="physical" name="comments">
```

Enter Comments Here

```
</textarea>
```

```
<input type="submit" value="Email Yourself">
```

```
</form>
```



http://localhost/lect3/Week3Forms/Forms/textArea.php

File Edit View Favorites Tools Help

• Comments

Enter Comments Here

Email Yourself

Macromedia Dreamweaver 8 - [M:\Maria Brennan\Web Applications\USBWebserver v8.5\root\lect3\Week3Forms\Forms\textArea.php (XHTML)]

File Edit View Insert Modify Text Commands Site Window Help

Common

mainform.css | opt.php | email.php | radio.php | listBox2.php | bands.html | checkbox2.php | checkbox3.php | textArea.php

Code Split Design Title:

```
1 <form method="post" action="process.php">
2
3 <li>
4 <label for="form-textarea">Comments</label><br>
5 <textarea rows="5" cols="20" wrap="physical" name="comments" id="form-textarea">
6 Enter Comments Here
7 </textarea></li>
8 <br><br>
9 <input type="submit" value="Email Yourself">
10
11 </form>
12
```

WRAP describes how the text in the text area should wrap at the end of lines. Until this attribute came along, browsers generally did not do word wrapping. If you typed a line that was longer than the display area, the line just kept going, hopefully with the display area scrolling along. This was not the way people are used to entering text, so Netscape added the WRAP attribute.

1K / 1 sec

17:03 12/02/2013

16:17 12/02/2013

# FIELDSET AROUND DATA

**This example demonstrates how to draw a border with a caption around your data.**

```
<fieldset>
```

```
<legend>
```

Health information:

```
</legend>
```

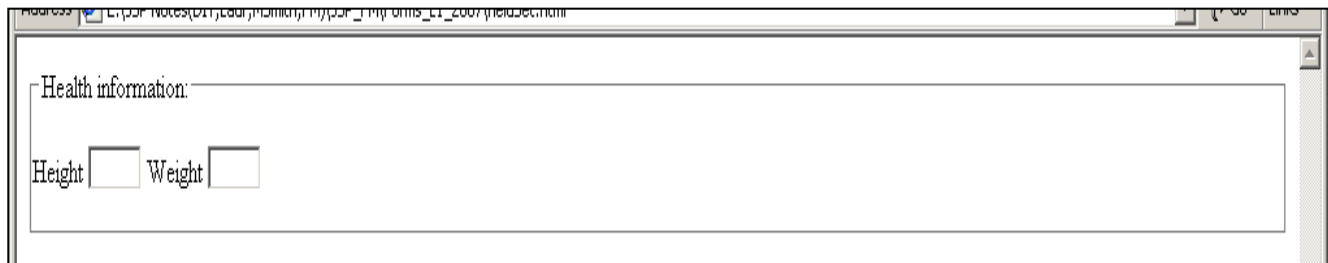
```
<form action="">
```

```
Height <input type="text" size="3">
```

```
Weight <input type="text" size="3">
```

```
</form>
```

```
</fieldset>
```



The screenshot shows a web browser window with the address bar displaying a local file path. The main content area shows a form with a legend titled "Health information:". Below the legend, there are two text input fields, one labeled "Height" and one labeled "Weight". The form is enclosed in a border, which is the visual representation of the <fieldset> and <legend> tags used in the code above.

# QUIZ

- Create a form that contains the following fieldset details:

Customer Information	
1. Full name:	<input type="text"/>
2. Password:	<input type="password"/>
3. Email:	<input type="text"/>
Health information:	
Height	<input type="text"/>
Weight	<input type="text"/>

```
1 <html>
2 <body>
3 <form action="process.php" method="post" >
4 <fieldset>
5 <legend>Customer Information</legend>
6 <ol>
7     <li><label>Full name:<input type = "text" name = "name"/></label></li>
8     <li><label>Password: <input type = "password" name = "password"/></label></li>
9     <li><label>Email:    <input type = "text" name = "eamil"/></label></li>
10 </ol>
11 </fieldset>
12 <fieldset>
13 <legend>Health information:</legend>
14     Height <label><input type="text" size="3"></label>
15     Weight <label><input type="text" size="3"></label>
16 </fieldset>
17 </form>
18 </body>
19 </html>
20
```



http://localhost/lect3/Week3Forms/Formsa/fieldSet.php

Institute...

New Tab

localhost

localhost

local... x

FileEditViewFavoritesToolsHelp

Customer Information

1. Full name:

2. Password:

3. Email:

Health information:

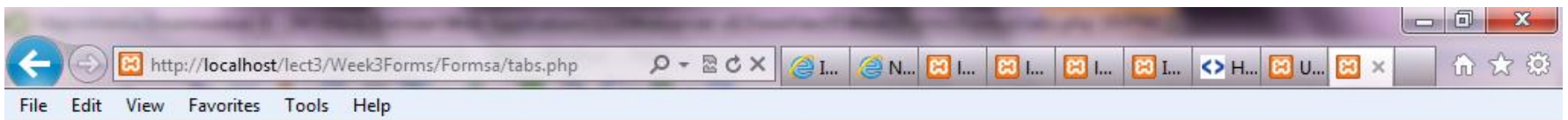
Height

Weight

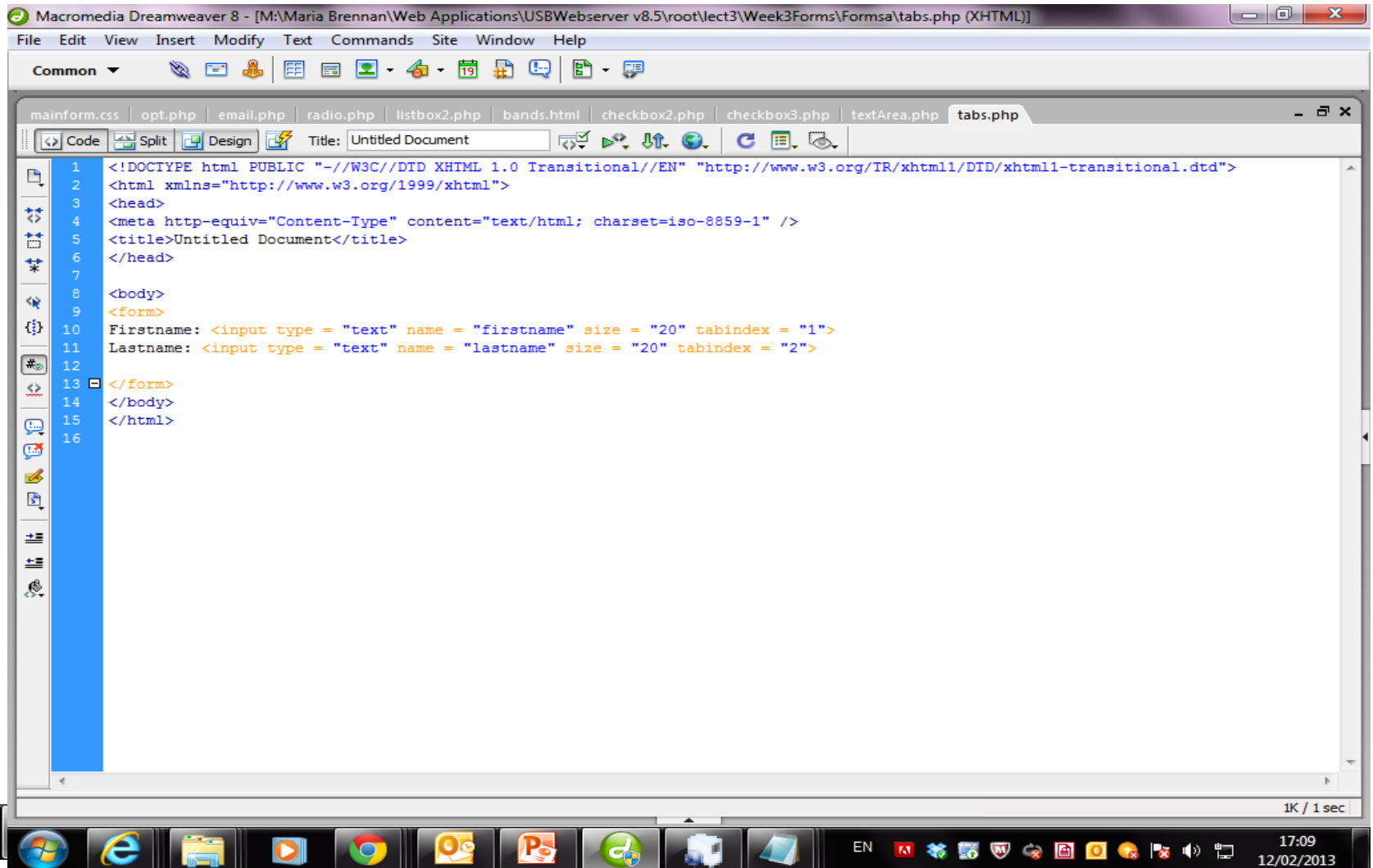
# SET TAB ORDER OF CONTROLS

- **Tab order** – determines the **sequence** in which the controls on a form receive the focus from the tab key when the Tab key is pressed
- By default, tab order is same as the sequence in which the controls are coded in HTML document
- To modify tab order use the **tabindex** attribute

```
<input type = "text" name = "firstname" size = "20" tabindex = "1">  
<input type = "text" name = "lastname" size = "20" tabindex = "2">
```



Firstname:  Lastname:



Macromedia Dreamweaver 8 - [tabs.php (XHTML)]

File Edit View Insert Modify Text Commands Site Window Help

Common

mainform.css | opt.php | email.php | radio.php | listBox2.php | bands.html | checkbox2.php | checkbox3.php | textArea.php | tabs.php

Code Split Design Title: Untitled Document

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
5 <title>Untitled Document</title>
6 </head>
7
8 <body>
9 <form>
10 Firstname: <input type = "text" name = "firstname" size = "20" tabindex = "1">
11 Lastname: <input type = "text" name = "lastname" size = "20" tabindex = "2">
12 Address: <input type = "text" name = "address" size = "20" tabindex = "4">
13 Phone: <input type = "text" name = "phone" size = "20" tabindex = "3">
14
15
16 </form>
17 </body>
18 </html>
19
```

http://localhost/lect3/Week3Forms/Forms/tabs.php

File Edit View Favorites Tools Help

Firstname: Lastname: Address: Phone:

1K / 1 sec

17:14 12/02/2013

# File Uploading

- You can also provide a facility to upload files using the following code:
- `enctype="multipart/form-data"` (must be included)
- `<input type = "file" ...>`

```
<form enctype="multipart/form-data" action="upload.php" method="post" >  
<h2>What files are you sending?</h2>  
<p><input type= "file" name="uploadfile" size="30" />  
</form>
```

**What files are you sending?**

Macromedia Dreamweaver 8 - [M:\Maria Brennan\Web Applications\USBWebserver v8.5\root\lect3\Week3Forms\Forms\uploading.php (XHTML)]

File Edit View Insert Modify Text Commands Site Window Help

Common

mainform.css | opt.php | email.php | radio.php | listBox2.php | bands.html | checkbox2.php | checkbox3.php | textArea.php | tabs.php | uploading.php

Code Split Design Title: Uploading Files

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4 <head>
5   <title>Uploading Files</title>
6 </head>
7 <body>
8
9
10 <form method="post" enctype="multipart/form-data" action="upload.php">
11
12
13 <h2>What files are you sending?</h2>
14
15 <p><input type="file" name="uploadfile" size="30" />
16
17 </form>
18 </body>
19 </html>
20
21 </body>
22 </html>
23
```

The enctype attribute specifies how the form-data should be encoded when submitting it to the server.

**Note:** The enctype attribute can be used only if method="post".

<body> <form>

1K / 1 sec

EN

17:17  
12/02/2013



## What files are you sending?



# Forms using Tables

## — Personal Information —

First Name:	<input type="text"/>
Last Name:	<input type="text"/>
Address:	<input type="text"/> <input type="text"/>
City:	<input type="text"/>
State:	<input type="text" value="Select a State/Province"/> ▼
Zip/Postal Code:	<input type="text"/>
Country:	<input type="text" value="US - United States"/> ▼



### Personal Information

Please enter your name and address as they are listed for your debit card, credit card, or bank account.

## — Contact Information —

	<div>How to Contact You?</div> <div><input type="radio"/> Phone <input checked="" type="radio"/> Email</div>
Email:	<input type="text"/> <small>We will never sell or disclose your email address to anyone. Once your account is setup, you may add additional email addresses.</small>
Re-enter Email:	<input type="text"/> <small>Must match the email address you just entered above.</small>



### Contact Information

Please enter your full email address, for example, **name@domain.com**

It is important that you provide a valid, working email address that you have access to as it must be verified before you can use your account.

Please enter a land line number, not a mobile phone number.



# FORMS AND TABLES

- Tables are a great way to align form details in a neat orderly fashion.
- The following examples show with and without table formats:

Name:	<input type="text"/>
Password:	<input type="password"/>
<input type="button" value="Send"/>	

Without  
Tables

Name:	<input type="text"/>
Password:	<input type="password"/>
<input type="button" value="Send"/>	

With Tables

# FORMS AND TABLES

## Example without using a Table

```
<form method="post" action="mailto:youremail@email.com">  
Name:  
<input type="text" size="10" maxlength="40" name="name">  
</br>  
Password:  
<input type="password" size="10" maxlength="10" name="password">  
</br>  
<input type="submit" value="Send">  
</form>
```

Name:

Password:

# FORMS AND TABLES

## Example using a Table

```
<form method="post" action="mailto:youremail@email.com">
<table>
<tr>
<td>Name:</td>
<td><input type="text" size="10" maxlength="40" name="name"></td>
<td></br></td></tr>
<tr><td>Password:</td>
<td><input type="password" size="10" maxlength="10" name="password"></td>
<td></br></td></tr>
<tr><td><input type="submit" value="Send">
</td></tr>
</table>
</form>
```

Name:	<input type="text"/>
Password:	<input type="password"/>
<input type="submit" value="Send"/>	

# What form elements are on this form?

## Survey

If you have a moment, we'd appreciate it if you would fill out this survey.

### Your information:

First Name

Last Name

Email Address

### How did you hear about us?

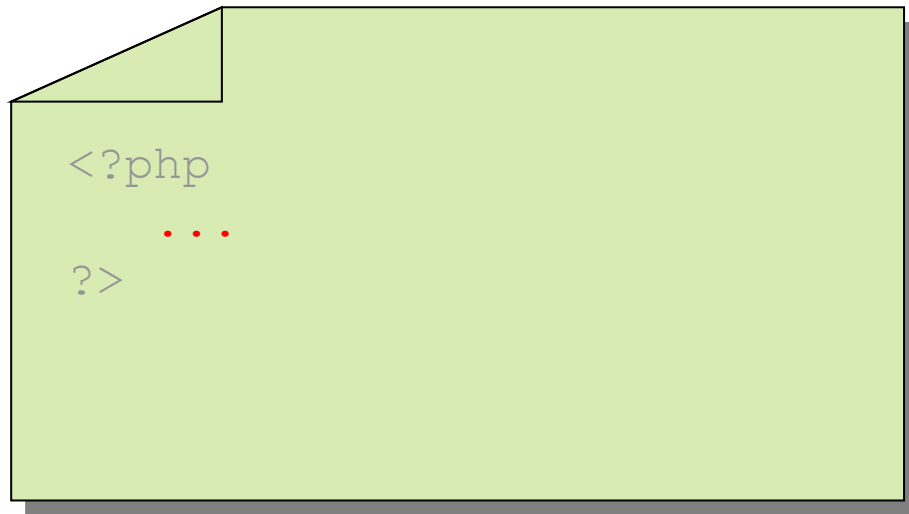
☐ Search engine ☐ Word of mouth ☐ Other

### Would you like to receive announcements about new CDs and special offers?

☒ YES, I'd like to receive information on new CDs and special offers.

Please contact me by

# Simple Form Processing



# FORMS: RECAP

- HTML form input fields send data as **key-value** pairs where the **input is the key**, and the input **field content is the value**.
- Form **action** attribute specifies a form that will process the data when the data is submitted.
- Form **method** attribute can be “**get**” or “**post**”

# GETTING FORM VALUES --- GET

Using 'GET'

fav.html

```
<html><head><title>Your Favorites</title></head><body>  
<form action="fav.php" method="get">  
  <b>Please enter your first name:</b>  
  <input type="text" size="45" name="username"> <br>  
  <b>Please select your favorite color wine:</b> <br>  
  <input type="radio" name="color" value="white"> White  
  <input type="radio" name="color" value="rosé"> Rosé  
  <input type="radio" name="color" value="red"> Red <br>  
  <b>Please enter your favorite dish:</b>  
  <input type="text" size="45" name="dish"><br><br>  
  <input type="submit" name = "button" value="Submit This Form">  
</form></body></html>
```

Please enter your first name:

Please select your favorite color wine:

☐ White ☐ Rosé ☐ Red

Please enter your favorite dish:

# Getting Form Values --- post

fav.html

Using POST

```
<html><head><title>Your Favorites</title></head><body>  
<form action="fav.php" method="post">  
  <b>Please enter your first name:</b>  
  <input type="text" size="45" name="username"> <br>  
  <b>Please select your favorite color wine:</b> <br>  
  <input type="radio" name="color" value="white"> White  
  <input type="radio" name="color" value="rosé"> Rosé  
  <input type="radio" name="color" value="red"> Red  
  <br>  
  <b>Please enter your favorite dish:</b>  
  <input type="text" size="45" name="dish"><br><br>  
  <input type="submit" name="button" value="Submit This  
  Form">  
</form></body></html>
```



GET

Your submission - Windows Internet Explorer

http://localhost/php5code/fav.php?username=Sean+&dish=Chicken+wings

Please enter your first name: Sean

Please select your favorite color wine:  
☐ White ☐ Rosé ☐ Red

Please enter your favorite dish: Chicken wings

Submit This Form

POST

http://localhost/php5code/fav.html

File Edit View Favorites Tools Help

McAfee SiteAdvisor

Your Favorites

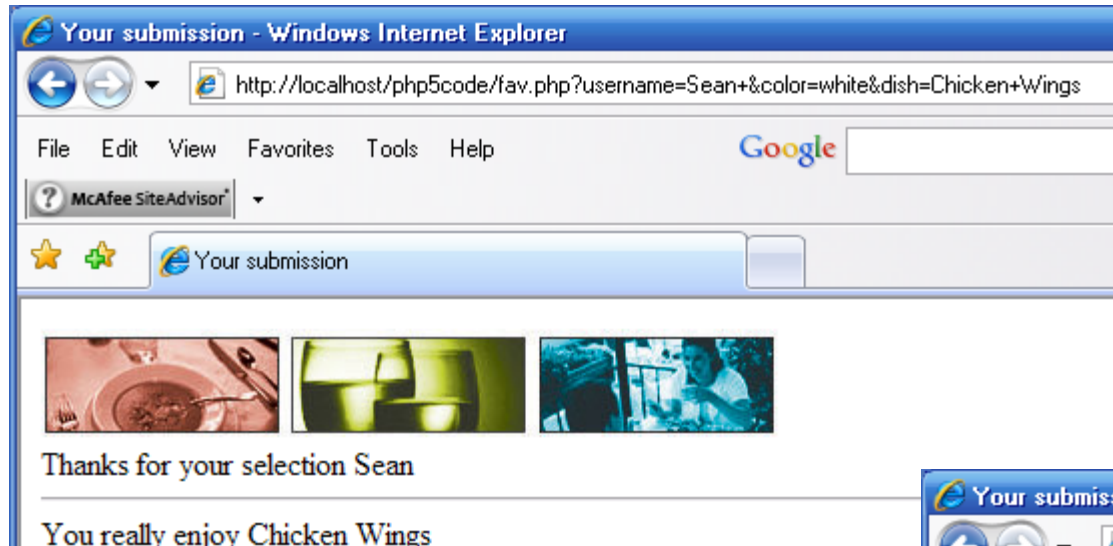
Please enter your first name: Paul

Please select your favorite color wine:  
☒ White ☐ Rosé ☐ Red

Please enter your favorite dish: meatballs

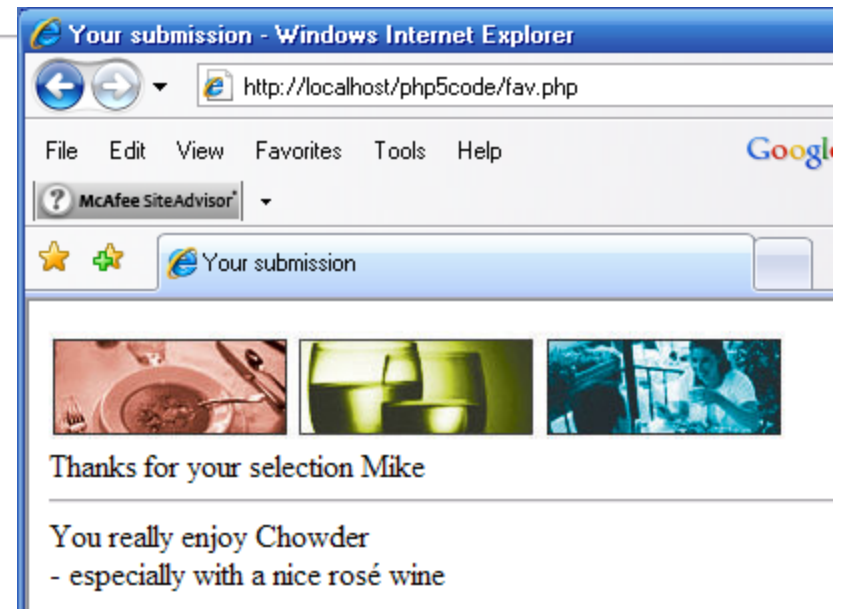
Submit This Form

# Displaying the contents



**“get” used**

**“post”**



# Forms: Get V Post methods

- There are **limitations to the GET** method:
  - long strings may exceed browser limitations
  - Name value pairs are visible to anyone
  - Not secure
  - However, URL's can be bookmarked

# Forms: Post method

- **POST** method doesn't have these limitations.
- Forms can specify either GET or POST
- In either case, the name value pairs are always available in the `$_REQUEST` array
- `$_REQUEST` used in most code examples
- These are super global variables.....

## WHAT ARE SUPERGLOBALS??

- **SEVERAL PREDEFINED VARIABLES IN PHP ARE "SUPERGLOBALS", WHICH MEANS THEY ARE AVAILABLE IN ALL SCOPES THROUGHOUT A SCRIPT. THERE IS NO NEED TO DO GLOBAL \$VARIABLE; TO ACCESS THEM WITHIN FUNCTIONS OR METHODS. THESE SUPERGLOBAL VARIABLES ARE:**

*\$GLOBALS*

*\$ \_SERVER*

*\$ \_GET*

*\$ \_POST*

*\$ \_FILES*

*\$ \_COOKIE*

*\$ \_SESSION*

*\$ \_REQUEST*

*\$ \_ENV*

We will get to test most of these thus semester.

# Form processing

- When you submit the form the **name-value** pairs are sent to the server-side script
- In the case of PHP these values are available to the script in the arrays `$_GET`, `$_POST`, `$_REQUEST`
  - `$_GET` contains any **name-value pairs** sent using the GET method.
  - `$_POST` contains any **name-value pairs** sent using the POST method.
  - `$_REQUEST` contains **all name-value pairs**.

# DISPLAYING SUBMITTED VALUES – VERSION 1

## fav.php

```
<html> <head><title>Your submission</title> </head> <body>
 <br>
<?php
    $username = $_REQUEST['username'];
    $color =   $_REQUEST['color'];
    $dish =   $_REQUEST['dish'];
    if( $username != null ) {
        echo "Thanks for your selection $username <hr/>" ;
    }
    if( ( $color != null ) && ( $dish != null ) ) {
        $msg = "You really enjoy $dish <br/>";
        $msg .= "- especially with a nice $color wine";
        echo( $msg );
    }
?>
</body></html>
```



# RECAP GETTING NAME-VALUE PAIRS

**Getting name-value pairs from form:**

**If get used:**

- `$username = $_GET['username'];`

**If post used:**

- `$username = $_POST['username'];`

**If not sure which has been used, or for flexibility use:**

- `$username = $_REQUEST['username'];`



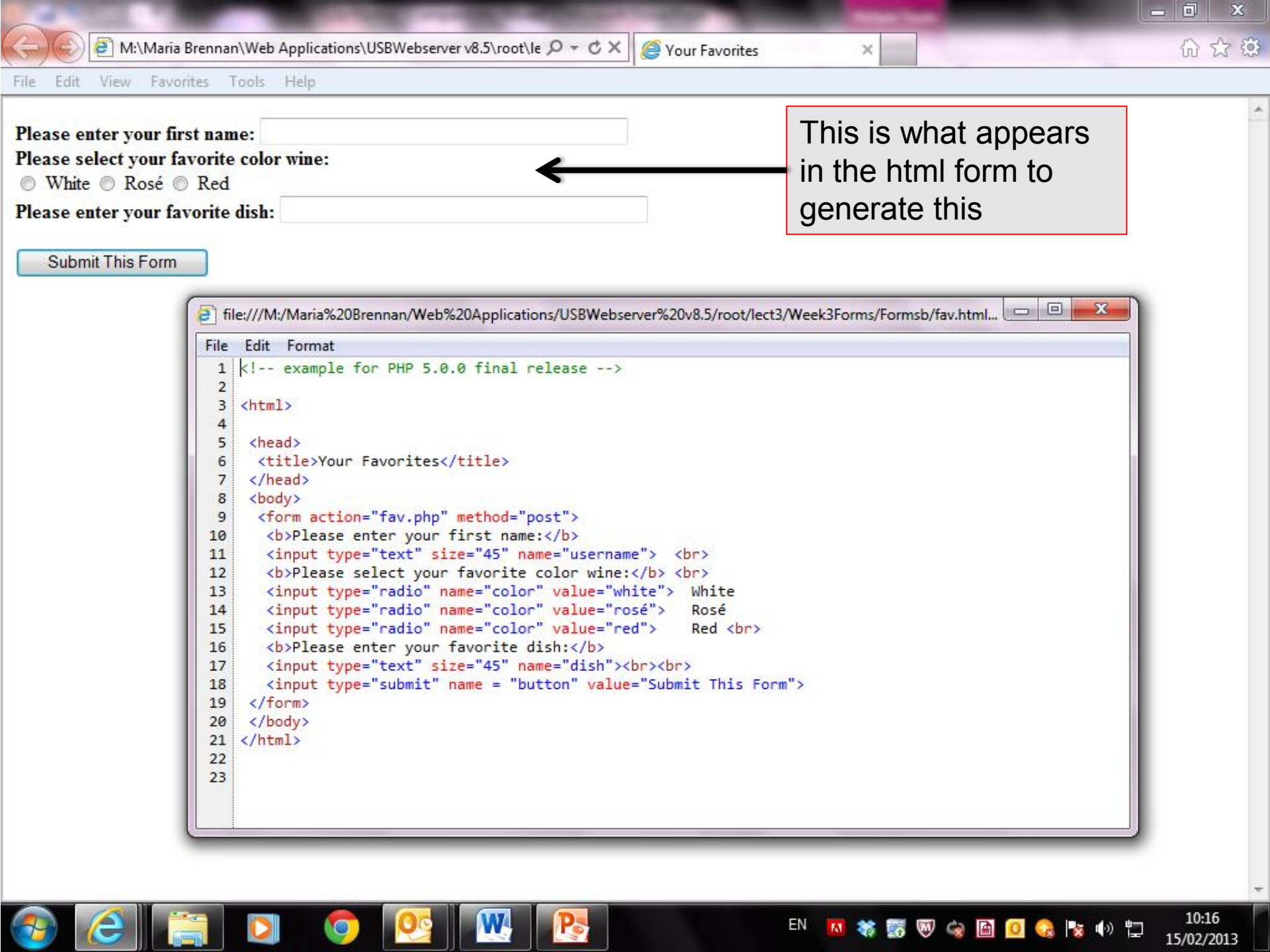
# RECAP GETTING NAME-VALUE PAIRS

- Use `$_GET` when someone is requesting data from your application
- Use `$_POST` when someone is pushing data to your application
- `$_REQUEST/$_GET` leave you open to attack as they store data in URL
- Speed difference is minimal

# THE ALL-IN-ONE FORM & RESPONSE

## Several common patterns in processing forms

1. Put the form in one file & the PHP to generate the response in another form.
2. Specify the second form as the action of the first  
Example: fav.html, fav.php
3. Form & response are put in the same PHP file.  
favForm.php



Now lets look at what goes in the [fav.php](#) file.....

# THE ALL-IN-ONE FORM & RESPONSE

- A test is made to see if the main form variable is set i.e. Submit button pressed
  - If it isn't, the HTML form is generated as output
  - If it is, the form is processed and a response is generated
- `if (isset($_REQUEST["buttonName"])) {`
    - `// do the form processing & generate the response here`
    - `}`
    - `else {`
      - `// generate the form here`
      - `}`
- Idea is to put as much HTML code outside main block of PHP code

# THE ALL-IN-ONE FORM & RESPONSE

**isset()** function checks for the existence of

..... is it set?

Use **\$\_SERVER['PHP\_SELF']** as the **action** in the form

**\$\_SERVER['PHP\_SELF']** evaluates to filename of the PHP script currently running.

It gives the **script location** relative to the document root.

**\$self = \$\_SERVER['PHP\_SELF'];** puts the file path into a variable called \$self

# THE ALL-IN-ONE FORM & RESPONSE

```
<html><head><title>Your Favorites</title></head><body>  
<form action= "<?php  
    echo $_SERVER['PHP_SELF'];  
?>"      method="post">  
  
...  
<input type="submit" name = "button" value="Submit">  
</form>...  
  
<?php  
?>
```

# DISPLAYING SUBMITTED VALUES – VERSION 2

**What if the user tries to get the form processed without entering any form data?**

**Check whether a field e.g. username has been filled in or not**

- Use `empty()` or `isset()` functions

**If not, re-direct the user to the form**

- Use the `header()` function

```
if (empty($_REQUEST['username']))  
{  
    header("Location: url or filename.php");  
    exit;  
}
```

# DISPLAYING SUBMITTED VALUES – VERSION 2

**header()** function redirects the user to a location – a url or the form itself

**header()** function must go at the top of the page – it has to happen before any HTML body is sent, or even whitespace.



# THE ALL-IN-ONE FORM & RESPONSE

```
<?php
if (isset($_REQUEST['button'])) //if the submit button is pressed
{
    If the submit button is pressed do this
    $self = $_SERVER['PHP_SELF']; //assign the function to a variable
    if(empty($_REQUEST['username'])) //if the username field is empty
    {
        header("Location: $self");
        exit; //call the program that is running using the header command
    }

    // process the form (using PHP) & display the form response
} //end of if block

Else ////display the html form
{
    Else generate the html again
    <html> create a HTML form here, using plain HTML </html>
    } // end of else-block
?>
```

# ALL-IN-ONE FORM LOGIC

favForm.php

## Pseudocode

### If Submit button pressed

Check if any fields in the form are blank  
If any fields are blank, display the HTML  
form again & exit  
If not blank -Get values sent in by user &  
process form & display results

### Else

Display the HTML form

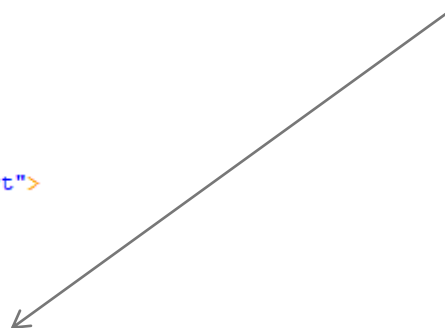
# THE ALL-IN-ONE FORM & RESPONSE

## Favform.php

```
<?php
if (isset($_REQUEST["button"]))
{
    $self = $_SERVER['PHP_SELF'];
    if ((empty($_REQUEST["username"])) || (empty($_REQUEST["dish"])) || (empty($_REQUEST["color"])))
    {
        header("Location:$self");
        exit;
    }
    echo "<img src='foodbanner.jpg' width='368' height='54' />";
    $username = $_REQUEST['username'];
    $color = $_REQUEST['color'];
    $dish = $_REQUEST['dish'];
    echo "<br/>Thanks for your selection $username <hr/>" ;
    $msg = "You really enjoy $dish <br/>";
    $msg .= "- especially with a nice $color wine";
    echo $msg;
    echo "<hr/>";
    echo '<a href= " ' . $self . ' " > Back to Form </a>';
}
else
{
    ?>
```

```
13 $dish = $_REQUEST['dish'];
14 echo "<br/>Thanks for your selection $username <br/>" ;
15 $msg = "You really enjoy $dish <br/>";
16 $msg .= "- especially with a nice $color wine";
17 echo $msg;
18 echo "<br/>";
19 echo '<a href= " ' . $self . ' " > Back to Form </a>';
20 }
21 else
22 {
23     ?>
24     <html>
25     <head>
26     <title>Your Favorites</title>
27     </head><body>
28         <?php echo $_SERVER['PHP_SELF']; ?>" method="post">
29         <b>Please enter your first name:</b>
30         <input type="text" size="45" name="username" > <br/>
31         <b>Please select your favorite color wine:</b> <br/>
32         <input type="radio" name="color" value="white"> White
33         <input type="radio" name="color" value="rosé"> Rosé
34         <input type="radio" name="color" value="red"> Red <br/>
35         <b>Please enter your favorite dish:</b>
36         <input type="text" size="45" name="dish"><br/><br/>
37         <input type="submit" name = "button" value="Submit">
38         <input type="reset">
39     </form>
40 </body>
41 </html>
42 <?php
43 }
44 ?>
45
```

Code continued...the else statement



# ALL-IN-ONE FORM

Please enter your first name:

Please select your favorite color wine:  
☐ White ☐ Rosé ☒ Red

Please enter your favorite dish:

Form will display again  
If form incomplete

Response to  
completed form



Thanks for your selection John

You really enjoy fish  
- especially with a nice red wine

[Back to Form](#)



# QUESTIONS

What does this code mean?

```
<form action= "<?php echo  
$_SERVER['PHP_SELF']; ?>"  
method="post">
```

# QUESTIONS

What does this code mean & where is it placed?

```
if (isset($_REQUEST["buttonName"])) {
```

```
..}
```



# QUESTIONS

What does this code do?

```
$self = $_SERVER['PHP_SELF'];
```

# QUESTIONS

**Explain this code:**

```
if(empty($_REQUEST['username'])) {  
    header("Location: $self");  
    exit;  
}
```

# QUESTION

What does this code do?

```
echo '<a href= " ' . $self . ' " > Back to Form </a>';
```

# MANIPULATING SUBMITTED VALUES

## Example – Arithmetic Operation

- Form allows users to enter 3 numbers and select an arithmetical operation to perform on them
- When the form is submitted, these values are sent to the form handler *calc.php* for processing
- Two versions shown:
  - 1. **calc.html & calc.php**
  - 2. All-in-one version **calcForm.php**

# FORM TO ENTER NUMBERSV1

```
<html> <head> <title>Calculation Form</title> </head><body>  
  <form action = "calc.php" method = "post">  
    Value 1: <input type = "text" name = "val1" size = "10">  
    Value 2: <input type = "text" name = "val2" size = "10">  
    <br>  
    Calculation: <br>  
    <input type = "radio" name = "calc" value = "add"> Add  
    <input type = "radio" name = "calc" value = "sub"> Subtract  
    <input type = "radio" name = "calc" value = "mul"> Multiply  
    <input type = "radio" name = "calc" value = "div"> Divide  
    <hr>  
    <input type = "submit" name = "button" value = "Calculate">  
    <input type = "reset" value = "Clear">  
  </form>  
</body>  
</html>
```

calc.html

# MANIPULATE SUBMITTED VALUES V1

Finds whether the given variable is numeric

```
<html> <head> <title>Calculation Result</title> </head> <body>
<?php
```

```
$val1 = $_REQUEST['val1'];
$val2 = $_REQUEST['val2'];
$calc = $_REQUEST['calc'];
```

```
if( is_numeric( $val1 ) && is_numeric( $val2 ) ){
```

```
    if( $calc != null ){
```

```
        switch( $calc ){
```

```
            case "add" : $result = $val1 + $val2; break;
```

```
            case "sub" : $result = $val1 - $val2; break;
```

```
            case "mul" : $result = $val1 * $val2; break;
```

```
            case "div" : $result = $val1 / $val2; break;
```

```
        }
```

```
        echo "Calculation result: $result" ;
```

```
    } }
```

```
else{
```

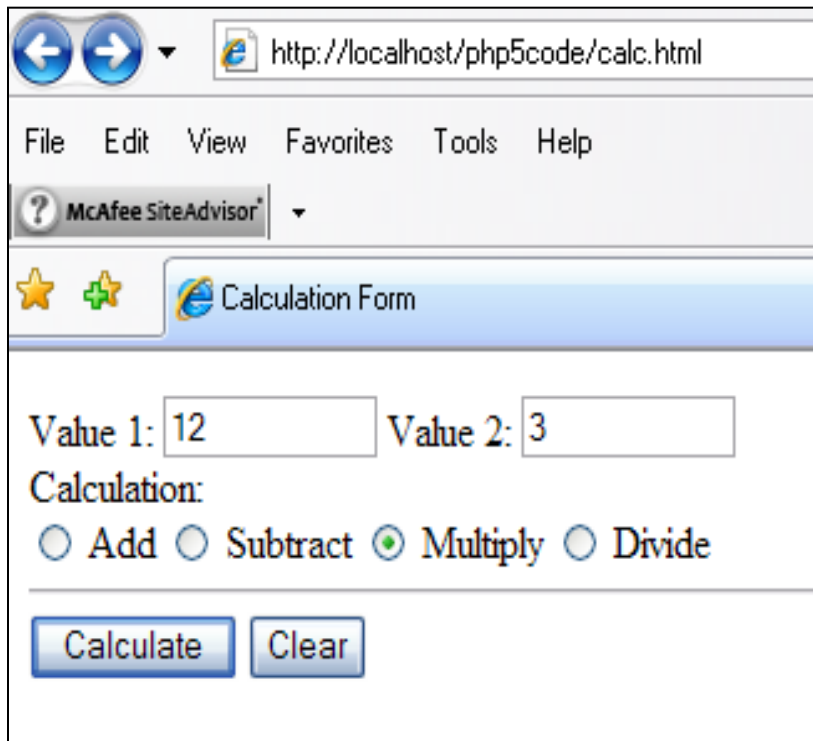
```
    echo "Invalid entry - please retry" ;
```

```
} ?>
```

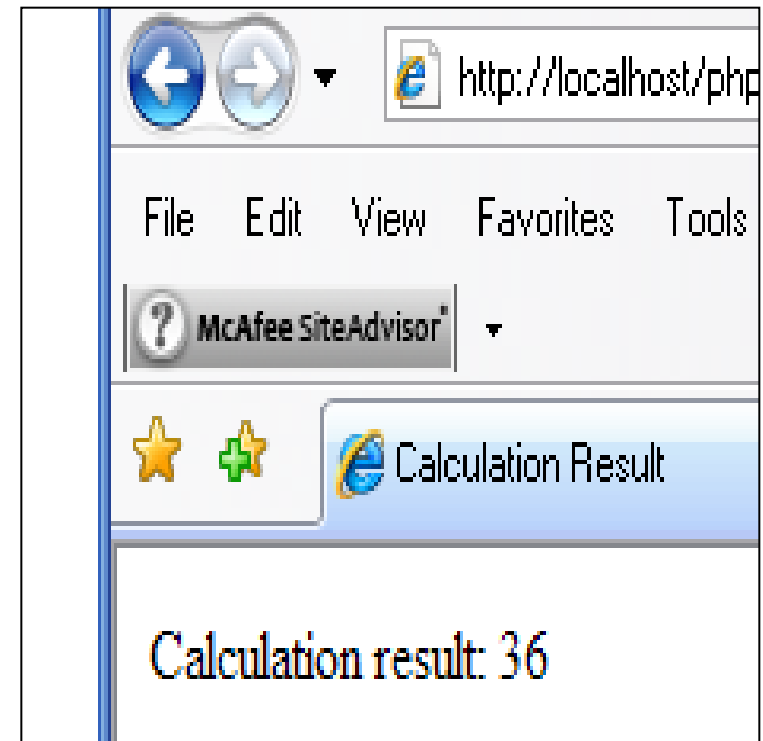
```
</body></html>
```

**calc.php**

# FORM AND RESULT



A screenshot of a web browser window displaying a form titled "Calculation Form". The browser's address bar shows the URL "http://localhost/php5code/calc.html". The form contains two input fields: "Value 1:" with the number "12" and "Value 2:" with the number "3". Below these fields is a "Calculation:" section with four radio buttons: "Add", "Subtract", "Multiply" (which is selected), and "Divide". At the bottom of the form are two buttons: "Calculate" and "Clear".



A screenshot of a web browser window displaying the result of the calculation. The browser's address bar shows the URL "http://localhost/php". The page title is "Calculation Result". The main content area displays the text "Calculation result: 36".

# ALL-IN-ONE FORM LOGIC

calcForm.php

## Pseudocode

### If Submit button pressed

If any fields empty, display HTML form again & exit  
    Retrieve values from fields entered  
    If values are numeric & all OK  
        process calculations, display results & hyperlink  
    Else  
        display error message & hyperlink

### Else

Display the HTML form



# ALL-IN-ONE FORM

calcForm.php

```
<?php
if (isset($_REQUEST["button"]))
{ if ((empty($_REQUEST["val1"])) || (empty($_REQUEST["val2"])) || (empty($_REQUEST["calc"])))
    { /// display the form again }
    // if values are numeric & all OK
    // do calculation
    // display results & link back to form
    // else – display error message & link to form
}
else{
?>
<html>...<form action = "<?php echo $_SERVER['PHP_SELF']; ?>" method = "post">
.....
<input type = "submit" name = "button" value = "Calculate">
</form></body></html>
<?php
}
?>
```

```

switch($calc ) {

    case "add" : $result = $val1 + $val2; break;
    case "sub" : $result = $val1 - $val2; break;
    case "mul" : $result = $val1 * $val2; break;
    case "div" : $result = $val1 / $val2; break; }
    echo "<title> Calculation Result </title>";
    echo "<h2> Calculation Result </h2><hr/>";

echo "The result of $calc sum of $val1 and $val2 is: <b>$result </b>" ;

    }
    } // is_numeric

else {

    echo "Invalid entry - please retry" ;

    //Hyperlink to form

    }

}

else { .....
form action = "<?php echo $_SERVER['PHP_SELF']; ?>" method = "post">
.....
{
?php }
?>

```

**calcForm.php**

# ALL-IN-ONE FORM – SAMPLE INPUT

Calculation Form

Value 1:  Value 2:

Calculation:

☐ Add ☒ Subtract ☐ Multiply ☐ Divide



Calculation Form

Value 1:  Value 2:

Calculation:

☐ Add ☐ Subtract ☐ Multiply ☐ Divide

Value 1:  Value 2:

Calculation:

☐ Add ☒ Subtract ☐ Multiply ☐ Divide



Invalid entry - please retry

[Back to Form](#)

# PROCESSING ARRAYS

In some cases the user may be able to select **more than one entry** from a listbox, checkbox, etc.



How do you  
process this entry?

# PROCESSING ARRAYS

For example, the user can select more than one band as follows:

Bands

I like U2: ☒

I like Rubberbandits: ☐

I like Coldplay: ☒

Submit Query

Reset

# PROCESSING ARRAYS

Its quite simple, just store the checkbox values as an array as follows:

```
<html>
<head><title>Form example</title></head>
<body>
<form method="get" action= "checkbox.php" >
<fieldset><legend>Bands</legend>
I like U2: <input type ="checkbox" value = "U2" name = "band" /></br>
I like Rubberbandits:  <input type = "checkbox" value= "Rubberbandits" name =
"band" /></br>
I like Coldplay:<input type = "checkbox" value = "Coldplay" name = "band" /></br>
</fieldset>
<br/><br/>
<input type="submit" name="button"/>
<input type="reset"/>
</form></body></html>
```

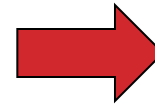
# PROCESSING ARRAYS

Checkbox.php is as follows:

```
<?php
if (isset($_REQUEST['button']))
{
    $self = $_SERVER['PHP_SELF'];
    $band = $_REQUEST['band'];

    // band, using a foreach loop
    echo "<b>The Band(s) selected is/are:</b>";
    echo "<ul>";
    if (isset($band)){
        foreach($band as $value){
            echo "<li> $value </li>";
        }
    }
    else{
        echo "No band was selected.";
    }
    echo "</ul>";
}

?>
```



The Band(s) selected is/are:

- U2
- Coldplay

Macromedia Dreamweaver 8 - [M:\Maria Brennan\Web Applications\USBWebserver v8.5\root\lect3\Week3Forms\Formsb\calcForm.php]

File Edit View Insert Modify Text Commands Site Window Help












Common

fav.php favForm.php calcForm.php calc.php calc.html

Code Split Design Title: Calculation Form

```
1 <?php
2 if (isset($_REQUEST["button"]))
3 {
4     $self = $_SERVER['PHP_SELF'];
5     if ((empty($_REQUEST["val1"])) || (empty($_REQUEST["val2"])) || (empty($_REQUEST["calc"])))
6     {
7         header("Location:$self");
8         exit;
9     }
10    $val1 = $_REQUEST['val1'];
11    $val2 = $_REQUEST['val2'];
12    $calc = $_REQUEST['calc'];
13    if(is_numeric( $val1 ) && is_numeric( $val2 ))
14    {
15        if( $calc != null )
16        {
17            switch($calc )
18            {
19                case "add" : $result = $val1 + $val2; break;
20                case "sub" : $result = $val1 - $val2; break;
21                case "mult" : $result = $val1 * $val2; break;
22                case "div" : $result = $val1 / $val2; break;
23            }
24
25            echo "<title> Calculation Result </title>";
26            echo " <h2> Calculation Result </h2><br/>";
27            echo "The result of $calc sum of $val1 and $val2 is: <b>$result </b> <br/><br/>" ;
28            echo '<a href= " ' . $self . ' " > Do another calculation</a>';
29        }
30        // is_numeric
31    else
32    {
33        echo "Invalid entry - please retry <br/><br/>" ;
34        echo '<a href= " ' . $self . ' " > Back to Form </a>';
35    }
36 }
```

2K / 1 sec



EN 10:53 15/02/2013



# Summary

Form  
Validation

Form Processing

Get and Post

Manipulating  
Values

All-in-one  
processing



# Summary

**Forms and  
Tables**

**Forms - HTML**

**File Uploading**



**Input Tags  
Textfields, Radio buttons  
Checkboxes**

**Fieldsets**

**Textareas**

**Dropdown Menu  
List box (basic HTML )**