

HTML Forms

What are forms?

- `<form>` is just another kind of HTML tag
- HTML forms are used to create simple graphical user interfaces (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's `parameters` tell the browser how to send the information to the server (i.e. which method)

Forms and PHP

- The PHP language can be used to make pages that “do something”
 - PHP code snippets can be attached to various form elements for data validation
 - For example, you might want to check that a `zipcode` field contains a 5-digit integer before you send that information to the server
- HTML forms can be used without PHP, but why would you?
 - i.e. HTML cannot do anything with it
- The most important thing to do with a form, other than filling it out (even in the “real-world”)?
 - Submitting it!!

The `<form>` tag

- The `<form arguments> ... </form>` tag encloses form elements (and probably other HTML as well)
- The arguments on the form tells the processor 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"` (default/optional) means use the same window
 - There are a couple others, but work with HTML frames

```
<form action="./login.php" method="post">
```

```
...
```

```
</form>
```

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

Text input

A text field:

```
<input type="text" name="textfield"
      value="with an initial value" />
```

A text field:

A multi-line text field

```
<textarea name="textarea" cols="24"
          rows="2">Hello</textarea>
```

A multi-line text field

A password field:

```
<input type="password" name="textfield3" value="secret" />
```

A password field:

- Note that two of these use the `input` tag, but one uses `textarea`

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: 

A reset button: 

A plain button: 

- submit: send data (i.e. submit form for processing)
- reset: restore all form elements to their initial state
- button: take some action as specified (usually using JavaScript)

- Note that the HTML tag is ***input***, not "*button*" (attribute of the input tag)

Checkboxes

A checkbox:

```
<input type="checkbox" name= "options[ ]"  
      value= "somevalue" checked= "checked" />
```

A checkbox:

- type: "checkbox"
- name: used to reference this form element from scripting language
 - The [] brackets indicate that the checkbox is submitted as an array. To be visited in further courses.
- value: value to be returned when element is checked
 - (if the checkbox is selected, this value is placed in the array)
- Note that there is *no text* associated with the checkbox—you have to supply text in the surrounding HTML

Radio buttons

```
Radio buttons:<br/>  
<input type="radio" name= "gender" value= "m" />male<br/>  
<input type="radio" name= "gender" value= "f" checked="checked"  
/>female<br/>
```

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" selected= "selected"
>red</option>
  <option value="green">green</option>
  <option value= "blue">blue</option>
</select>
```

A menu or list: 

- 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 PHP) before the form is submitted

Simple Form Example - GET

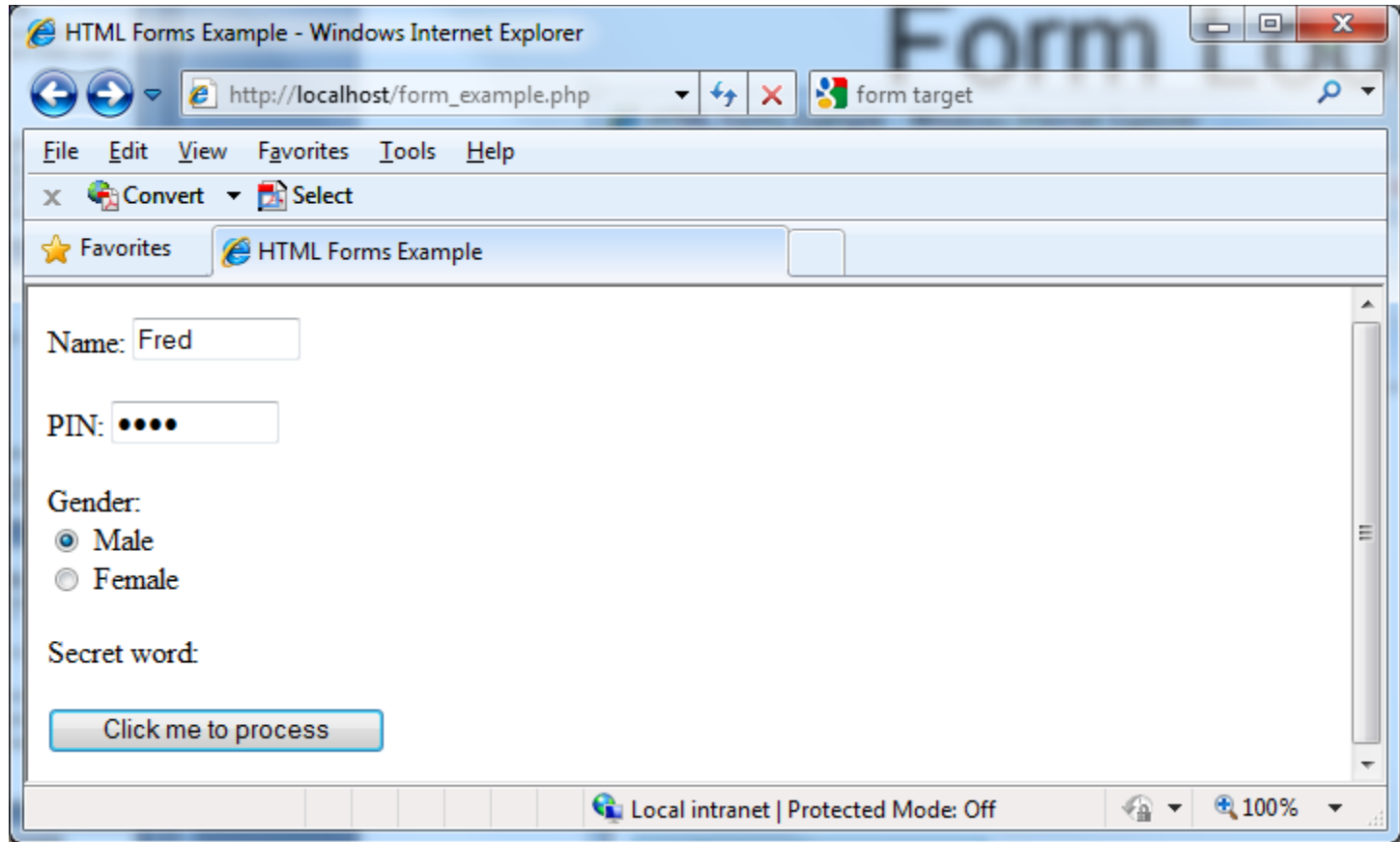
```
<?php
    $title = "HTML Forms Example";
    include "header.php";
?>

<form method="get" action="./form_process.php" >

    <p>Name:      <input type="text" name="identity" size="10" /></p>
    <p>PIN:      <input type="password" name="pin" size="10" /></p>
    <p>Gender:<br/>
                <input type="radio" name="sex" value="m" /> Male<br />
                <input type="radio" name="sex" value="f" /> Female
    </p>
    <p>Secret word: <input type="hidden" name="word" value="bird" />
    </p>
    <input type="submit" value="Click me to process" />
</form>

<?php
    include "footer.php";
?>
```

Completed Form Looks Like



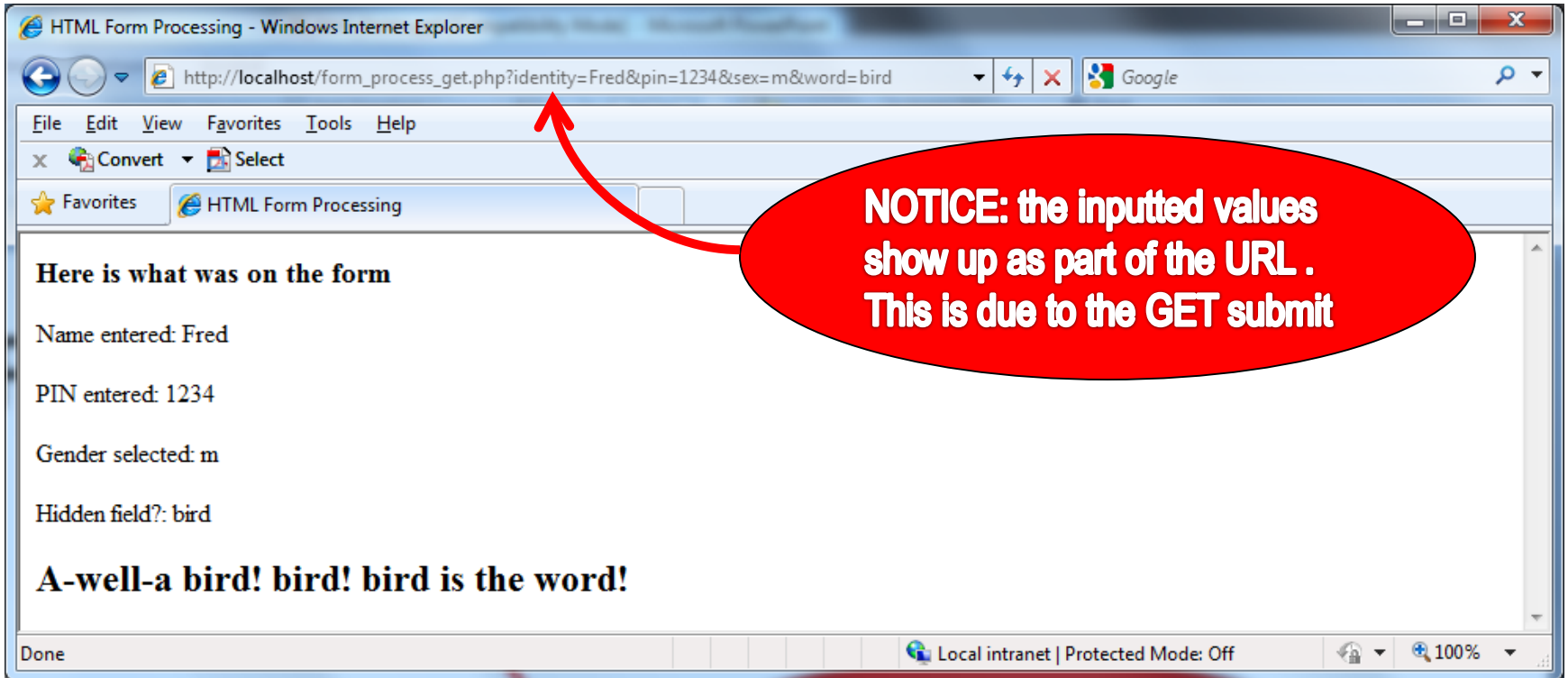
- N.B. Fred entered 1234 for his PIN

Simple Form Processing - GET

```
<?php //PHP section
    $title = "HTML Form Processing";
    include "header.php";

    //we are using $_GET because that
    //is the method the form was submitted in
    $name = $_GET['identity'];        //'identity' is from the name attribute
    on the form
    $number = $_GET['pin'];           //same thing with 'pin' and 'gender' and
    'word'
    $gender = $_GET['sex'];
    $secret = $_GET['word'];
?>
<!-- HTML section with some PHP output -->
<h3>Here is what was on the form</h3>
<p>Name entered: <?php echo $name; ?></p>
<p>PIN entered: <?php echo $number; ?></p>
<p>Gender selected: <?php echo $gender; ?></p>
<p>Hidden field?: <?php echo $secret; ?></p>
<h2><?php echo "A-well-a " . $secret . "! " . $secret . "! " . $secret . "
    is the word!"; ?></h2>
<?php
    include "footer.php";
?>
```

Processed Form Looks Like - GET



The screenshot shows a Windows Internet Explorer browser window titled "HTML Form Processing - Windows Internet Explorer". The address bar displays the URL: `http://localhost/form_process_get.php?identity=Fred&pin=1234&sex=m&word=bird`. A red arrow points from a red callout bubble to the URL. The callout bubble contains the text: "NOTICE: the inputted values show up as part of the URL. This is due to the GET submit". The browser's menu bar includes "File", "Edit", "View", "Favorites", "Tools", and "Help". The address bar also shows "Convert" and "Select" options. The Favorites bar contains "HTML Form Processing". The main content area displays the following text:

Here is what was on the form

Name entered: Fred

PIN entered: 1234

Gender selected: m

Hidden field?: bird

A-well-a bird! bird! bird is the word!

The status bar at the bottom shows "Done", "Local intranet | Protected Mode: Off", and "100%".

Simple Form Example - POST

```
<?php
    $title = "HTML Forms Example";
    include "header.php";
?>
<h1>Exact same example, but using POST this time</h1>
<form method="post" action="./form_process.php" >

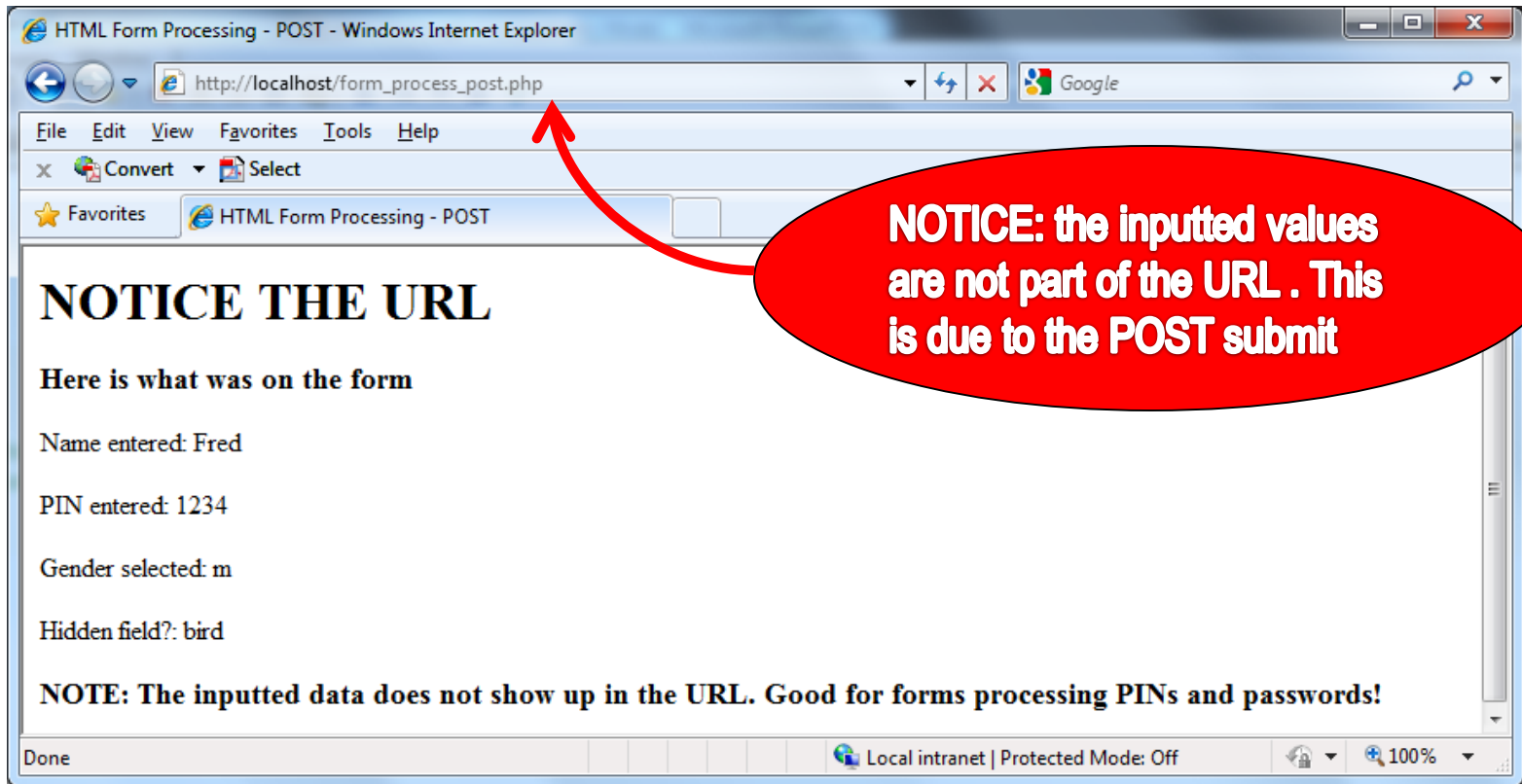
    <p>Name:      <input type="text" name="identity" size="10" /></p>
    <p>PIN:      <input type="password" name="pin" size="10" /></p>
    <p>Gender:<br/>
                <input type="radio" name="sex" value="m" /> Male<br />
                <input type="radio" name="sex" value="f" /> Female
    </p>
    <p>Secret word: <input type="hidden" name="word" value="bird" />
    </p>
    <input type="submit" value="Click me to process" />
</form>

<?php
    include "footer.php";
?>
```


Simple Form Processing-POST

```
<?php //PHP section
    $title = "HTML Form Processing - POST";
    include "header.php";
    //we are using $_POST because that
    //is the method the form was submitted in
    $name = $_POST['identity'];           //'identity' is from the name
    attribute on the form
    $number = $_POST['pin'];             //same thing with 'pin' and 'gender'
    and 'word'
    $gender = $_POST['sex'];
    $secret = $_POST['word'];
?>
<!-- HTML section with some PHP output -->
<h3>Here is what was on the form</h3>
<p>Name entered: <?php echo $name; ?></p>
<p>PIN entered: <?php echo $number; ?></p>
<p>Gender selected: <?php echo $gender; ?></p>
<p>Hidden field?: <?php echo $secret; ?></p>
<h3>NOTE: The inputted data does not show up in URL. Good for forms
    processing PINs and passwords</h3>
<?php
    include "footer.php";
?>
```

Processed Form Looks Like - POST



POST is like submitting a paper-form in an envelope. To protect your privacy.