



COMP 4021

Internet Computing

HTML Forms

Dr. Kenneth LEUNG

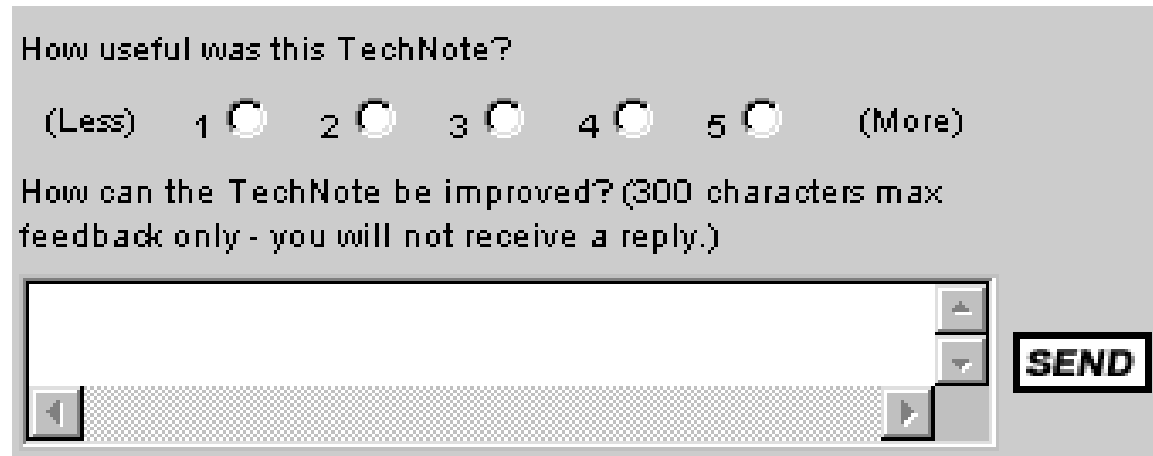
Slides created by Dr. David ROSSITER

What is a Form?

- Forms are the easiest way to send data from a browser to a server program
- The user can enter various types of information, then send them to the server
- You may already know some of this material – so this presentation file may be more useful as a kind of reference

Common Form Usage

- Typical example:



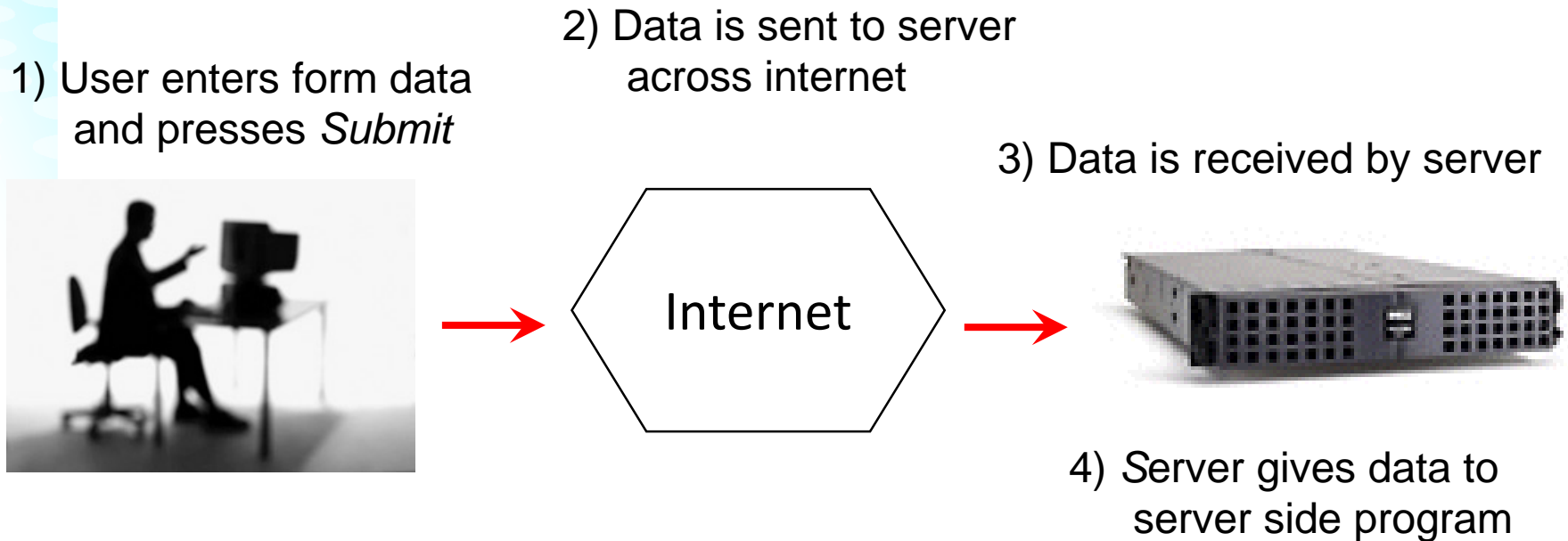
How useful was this TechNote?

(Less) 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ (More)

How can the TechNote be improved? (300 characters max feedback only - you will not receive a reply.)

- After the *Submit* button is pressed, the data is (usually) sent to a particular program running on a particular server, somewhere on the internet

Sending the Form Data



JavaScript Processing

- Commonly, forms are ‘dead’
 - User just types/ selects, then submits
- Can do client-side processing of form data
 - Could use JavaScript to do some kind of action, such as assess checking data when an event occurs
 - Don’t actually *have* to send data to server, could use a form just for text input to JavaScript, etc

Useful JavaScript Events

- You already know lots of events for handling the mouse/keyboard, etc - here are some events which can be used for handling forms:
- onsubmit - fires when the user submits the form data (fires just before the data is actually sent)
- onreset - fires when the user resets all the fields in a form (fires just before the data is actually reset)
- onchange - fires just after the contents of something is changed
- onselect – fires when user selects text in a text field

Example JavaScript Processing

- Example JavaScript checking when user triggers *onsubmit*:
 - Checking student ID field has correct number of characters
 - Checking telephone number field only has digits

- Example JavaScript change of image when user triggers *onchange*:

- Nothing yet entered, dull image:

Search!

- As soon as text entered, bright image:

Search!

Constructing a Form

method is the method used to transfer the data to the server

- Either *get* or *post*
- Differences discussed later

action is the location of the server side script

```
<form method="get"  
action="http://ihome.ust.hk/~rossiter/cgi-bin/show_environment.php"  
>
```


```
...  
</form>
```

} HTML tags for form content go here


- As you know, you should use lower case letters for all HTML, which includes forms

Reset & Submit Buttons

- For a reset button use *input type="reset"*
- Reset buttons are optional - not appropriate if there are only a few simple fields on the form
- Example:

Press `<input type="reset" value="here"/>` to clear the form. Press  to clear the form.

- For a submit button use *input type=submit*
- Example:

Press `<input type="submit" value="here"/>` to submit your data. Press  to submit your data.

Text Input Field

- Use *input type="text"* for a text field
- Various possibilities: ([] = optional)

```
<input [type=text] name="text-id" [size=nn]  
      [maxlength=nn] [value="default text"] />
```

- Example:

Name:

Name: `<input name="name" size="32"
 value="(type your name here)" maxlength="255"/>`

Text Area Field

- Use *textarea* for text input bigger than one line
- Various possibilities: ([] = optional)

```
<textarea name="text-id" [cols=mm] [rows=nn]>  
  default text</textarea>
```

- **Example:**

Any comments you would
like to make:


```
<textarea name="comments"  
  rows="3" cols="32"> </textarea>
```

Any comments you would like to make:



Radio Button Input

- Use *input type="radio"* for radio buttons
- Possibilities: `<input type="radio" name="radio-set-id" value="choice-id" [checked] />`
- Example: Student: ☐ Masters ☒ Undergraduate

Student:

```
<input type="radio" name="degree" value="masters" />Masters  
<input type="radio" name="degree" value="undergraduate"  
checked/>Undergraduate
```

Checkbox Input

- Use *input type="checkbox"* for a series of “on” or “off” switches
- Possibilities: `<input type="checkbox" name="box-set-id" value="choice-id" [checked] />`

- Example:

Indicate your area(s)
of specialty

Indicate your area(s) of speciality
Click on one or more areas

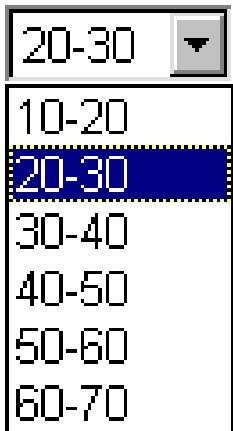
☐ Management
☐ Sales

```
<input type="checkbox" name="speciality" value="1"/>Management<br/>
<input type="checkbox" name="speciality" value="2"/>Sales<br/>
```

Select Input

- Use *input type="select"* when one single option needs to be selected from several possibilities. Example:

```
Age range: <select name="age_range">  
  <option>10-20</option>  
  <option selected>20-30</option>  
  <option>30-40</option>  
  <option>40-50</option>  
  <option>50-60</option>  
  <option>60-70</option>  
</select>
```

Age range: 

Complete Form Code 1/5

```
<html>
<body>
<h1>HTML Form Example</h1>

<form method="get"
action="http://ihome.ust.hk/~rossiter/cgi-bin/
                                show_environment.php">
```

Name:

```
<input name="name" size="32" value="(type your name here)"
      maxlength="255" />
```

Complete Form Code 2/5

<p> Student:

<input type="radio" name="degree" value="masters" />Masters

<input type="radio" name="degree" value="undergraduate" checked />
Undergraduate</p>

<p> Age range:

<select name="age_range">

<option>10-20</option>

<option selected>20-30</option>

<option>30-40</option>

<option>40-50</option>

<option>50-60</option>

<option>60-70</option>

</select> </p>

Complete Form Code 3/5

<p>Indicate your area(s) of specialty

<i><small>Click on one or more areas</small></i>

<input type="checkbox" name="specialty" value="1"/>Management

<input type="checkbox" name="specialty" value="2"/>Sales

<input type="checkbox" name="specialty" value="3"/>Human
Resources

<input type="checkbox" name="specialty" value="4"/>IT/ database

<input type="checkbox" name="specialty" value="5"/>IT/ networking

<input type="checkbox" name="specialty" value="6"/>IT/ internet

<input type="checkbox" name="specialty" value="7"/>IT/ general

Complete Form Code 4/5

```
<br/>
```

Any comments you would like to make:

```
<br/>
```

```
<textarea name="comments" rows="3" cols="32"/>
```

```
</p>
```

```
<p>
```

```
<input type="hidden" name="username" value="rossiter"/>
```

```
<input type="image" name="image" src="send_button.gif" align="centre"/>
```

```
<br/>
```

```
<br/>
```

```
</p>
```

To be discussed in
the next few slides

Complete Form Code 5/5

Press `<input type="submit" value="here"/>` to submit your data.

`
`

`
`

Press `<input type="reset" value="here"/>` to clear the form.

`
`

`
`

`</form>`

`</body>`

`</html>`

Complete Form Display

- After making the selections shown on the right the form is then submitted by clicking on the submit button
- The response is shown in the following two slides

HTML Form Example

Name:

Student: ☐ Masters ☒ Undergraduate

Age range:

Indicate your area(s) of specialty

Click on one or more areas

- ☐ Management
- ☒ Sales
- ☒ Human Resources
- ☒ IT/ database
- ☐ IT/ networking
- ☐ IT/ internet
- ☐ IT/ general

Any comments you would like to make:

Send

Press to submit your data.

Press to clear the form.

Data Seen By Server Side Program 1/2

This PHP program sees the following environment variables....

- Environment var **PATH** is **/usr/local/bin:/usr/bin:/bin**
- Environment var **DOCUMENT_ROOT** is **/usr/local/apache-1.3.28/htdocs**
- Environment var **HTTP_ACCEPT** is
text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
- Environment var **HTTP_ACCEPT_CHARSET** is **ISO-8859-1,utf-8;q=0.7,*;q=0.7**
- Environment var **HTTP_ACCEPT_ENCODING** is **gzip,deflate**
- Environment var **HTTP_ACCEPT_LANGUAGE** is **en-us**
- Environment var **HTTP_CONNECTION** is **keep-alive**
- Environment var **HTTP_HOST** is **ihome.ust.hk**
- Environment var **HTTP_KEEP_ALIVE** is **300**
- Environment var **HTTP_USER_AGENT** is **Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.3) Gecko/20070309 Firefox/2.0.0.3**

Data Seen By Server Side Program 2/2

- Environment var **REMOTE_ADDR** is **218.102.38.10**
- Environment var **REMOTE_PORT** is **4111**
- Environment var **SCRIPT_FILENAME** is **/home/rossiter/public_html/cgi-bin/show_environment.php**
- Environment var **SERVER_ADDR** is **143.89.14.183**
- Environment var **SERVER_ADMIN** is **cchelp@ust.hk**
- Environment var **SERVER_NAME** is **ihome.ust.hk**
- Environment var **SERVER_PORT** is **80**
- Environment var **SERVER_SOFTWARE** is **Apache/1.3.28 (Unix) mod_jk/1.2.15 FrontPage/5.0.2.2634 mod_ssl/2.8.15 OpenSSL/0.9.7c**
- Environment var **TZ** is **GMT-8**
- Environment var **GATEWAY_INTERFACE** is **CGI/1.1**
- Environment var **SERVER_PROTOCOL** is **HTTP/1.1**
- Environment var **REQUEST_METHOD** is **GET**
- Environment var **QUERY_STRING** is **name=Donald°ree=undergraduate&age_range=20-30&speciality=2&speciality=3&speciality=4&comments=Great&version_date=15_feb_2007**
- Environment var **REQUEST_URI** is
/~rossiter/cgi-bin/show_environment.php?name=Donald°ree=undergraduate&age_range=20-30&speciality=2&speciality=3&speciality=4&comments=Great&version_date=15_feb_2007
- Environment var **SCRIPT_NAME** is **/~rossiter/cgi-bin/show_environment.php**
- Environment var **PHP_SELF** is **/~rossiter/cgi-bin/show_environment.php**

The Fields Which Contain Form Data

- In this example, the GET method is being used to send data to the server
- You can see from the last two slides that the QUERY_STRING field contains the form data
- Because the GET method is being used, the REQUEST_URI field also has the form data
- The server side program can extract the data from either of those strings

Images in Forms

- If you just want to put an image in a form for visual effect and nothing more, just use the usual `` instruction
- If you want to use an image to act as a submit button, use *input type="image"*
- For example:

```
<input type="image" name="image"
      src="send_button.gif" />
```



Images in Forms

- Possible syntax:

```
<input type="image"  
name="image-id" src="image-url" [align=alignment]/>
```

- There is another advantage to using images
- The exact (x, y) position that the user clicks on is automatically sent to the server side program along with all the other form data
- The server side program may choose to use it or ignore it

Images in Forms

What is your favourite Hong Kong shopping area?



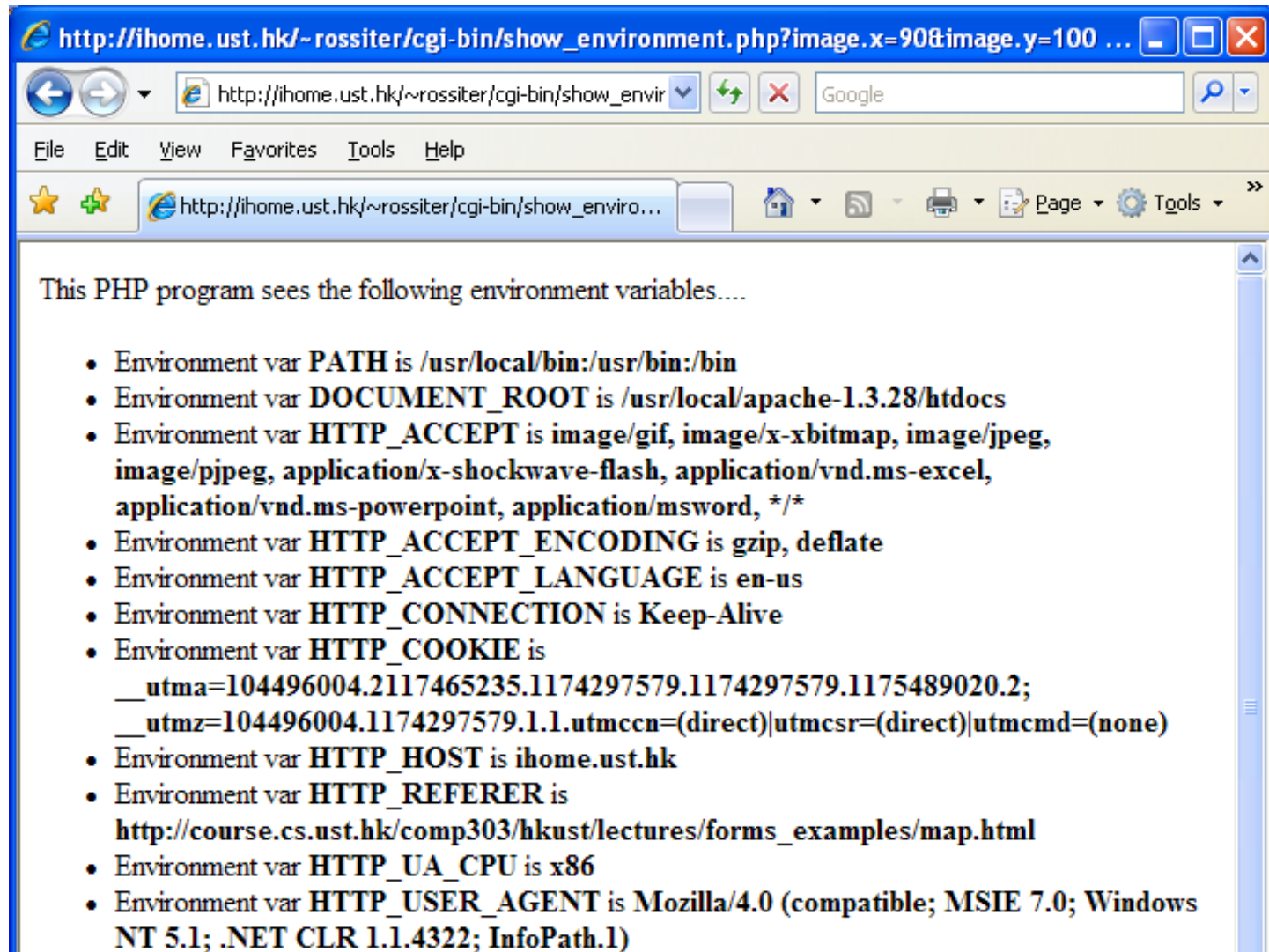
- Example clickable map:

What is your favourite
Hong Kong shopping area?


```
<form method="get"
action="http://ihome.ust.hk/~rossiter/cgi-bin/
                                show_environment.php">
<input type="image" name="image" src="harbour.gif"/>
</form>
```

- After clicking on the image data is sent to the server program
- This particular server side program simply lists all the data it knows about – see next two slides

Data Seen By Server Side Program 1/2



Data Seen By Server Side Program 1/2

- Environment var **REMOTE_ADDR** is 218.102.38.10
- Environment var **REMOTE_PORT** is 3949
- Environment var **SCRIPT_FILENAME** is /home/rossiter/public_html/cgi-bin/show_environment.php
- Environment var **SERVER_ADDR** is 143.89.14.183
- Environment var **SERVER_ADMIN** is cchelp@ust.hk
- Environment var **SERVER_NAME** is ihome.ust.hk
- Environment var **SERVER_PORT** is 80
- Environment var **SERVER_SOFTWARE** is Apache/1.3.28 (Unix) mod_jk/1.2.15 FrontPage/5.0.2.2634 mod_ssl/2.8.15 OpenSSL/0.9.7c
- Environment var **TZ** is GMT-8
- Environment var **GATEWAY_INTERFACE** is CGI/1.1
- Environment var **SERVER_PROTOCOL** is HTTP/1.1
- Environment var **REQUEST_METHOD** is GET
- Environment var **QUERY_STRING** is image.x=90&image.y=100
- Environment var **REQUEST_URI** is /~rossiter/cgi-bin/show_environment.php?image.x=90&image.y=100
- Environment var **SCRIPT_NAME** is /~rossiter/cgi-bin/show_environment.php
- Environment var **PHP_SELF** is /~rossiter/cgi-bin/show_environment.php

The server side program can extract the form data (i.e., X and Y values) from either the QUERY_STRING or REQUEST_URI fields

Hidden Fields

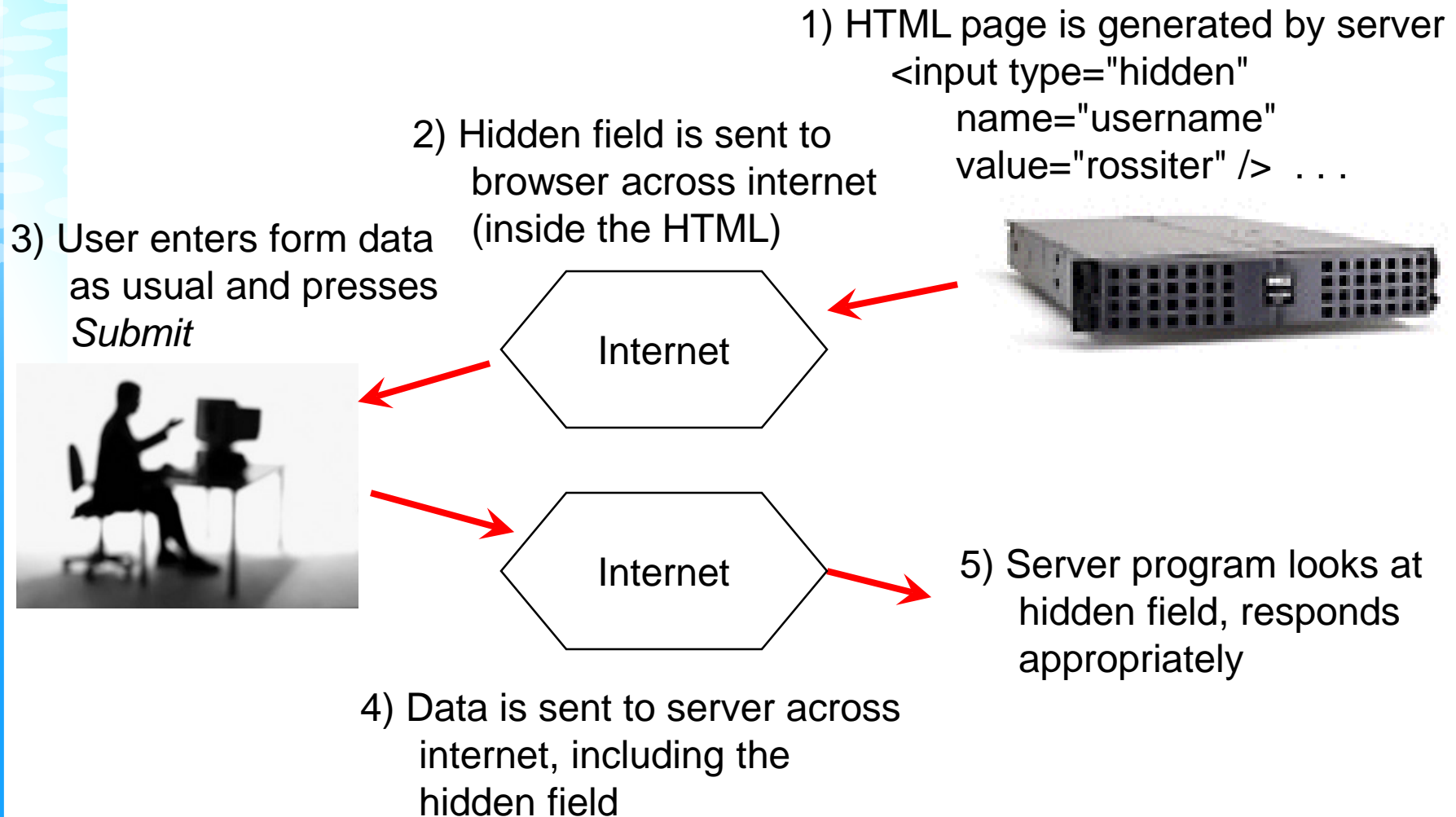
- Use *input type="hidden"* for hidden fields
- The field does not appear in the visual display
- The user can't change the value of the field
- Most users won't know about it
- This method can be used for storing some kind of 'state' information
- This can be very useful, particularly if the user/ browser does not permit cookies on their machine
- Example:

```
<input type="hidden" name="creation_time" value="15_feb_2010_14:32" />
```

Hidden Fields/ Cookies/ Sessions

- Hidden fields are useful for implementing a system 'variable' (i.e., something which both browser and server code can read and change)
- However, cookies and sessions are more commonly used, to be discussed later

Example Hidden Field Usage



Get or Post

- A reminder - there's two ways to send information from client to server; 'get' or 'post'

`<form method="get" action="http://ihome.ust.hk/~rossiter/cgi-bin/show_environment.php">`

`...`

`</form>`

Example Get

- When the form data is sent, all the parameter names and values are shown on the browser location line



Google **Web** [Images](#) [Groups](#) [News](#) [more »](#)
xml specification [Advanced Search](#)
[Preferences](#)
Search: ☒ the web ☐ pages from Hong Kong

Web

Results 1 - 10

[Extensible Markup Language \(XML\) 1.0 \(Second Edition\)](#)

... This version of the **XML specification** may be distributed freely, as long as all text and legal notices remain intact. 1.2 Terminology. ...

[www.w3.org/TR/2000/REC-xml-20001006](#) - 101k - [Cached](#) - [Similar pages](#)

The Get Method

- Good points
 - All the parameter names and values are shown - helpful for debugging a system you are developing
 - Easy for a program to build up the single text sequence and send it
- Bad points
 - Maximum length of a URL is about 2000 characters, sometimes is not enough
 - Users can see all the parameter names and values on the browser location line, even the 'hidden' ones
 - Users can easily change the values and send them to your system, could cause trouble for you

The Post Method

- When the form data is sent using the post method, all the parameter names and values, etc, are added after the HTTP header
- Good points
 - User can't see any parameter names/ values
 - There is no limit on how many bytes you can send from client to server i.e. can send many large MP3 files, etc, using the Post method
- Bad points
 - A little bit harder to program for some languages (on the server side)