

COMP 4021

Internet Computing

PHP 2

Dr. Kenneth LEUNG

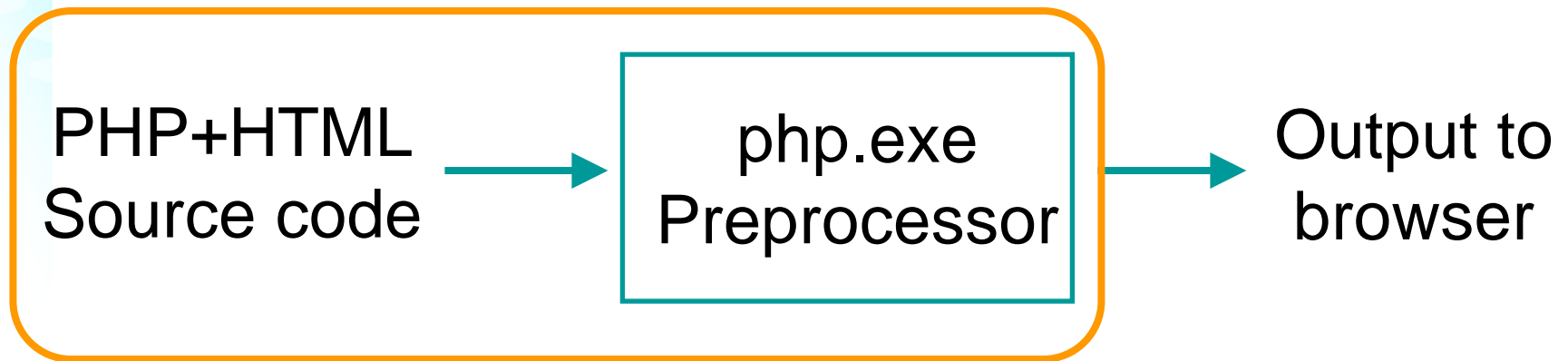
Slides created by Dr. David ROSSITER

Overview

- This presentation considers
 - Debugging PHP
 - X-Powered-By
 - Checking HTTP communication
 - Browser redirection
 - Handling cookies with PHP
 - Setting, Accessing, Displaying, Deleting

PHP Operation - Reminder

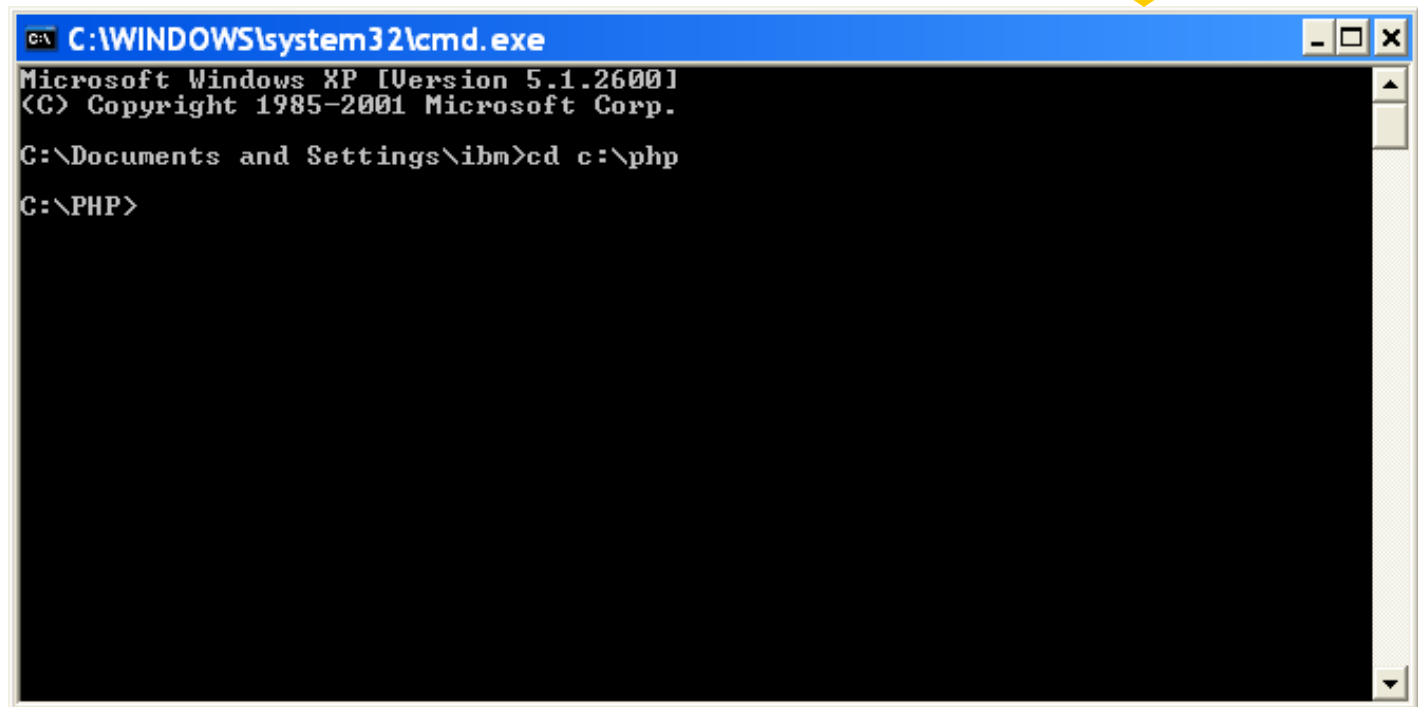
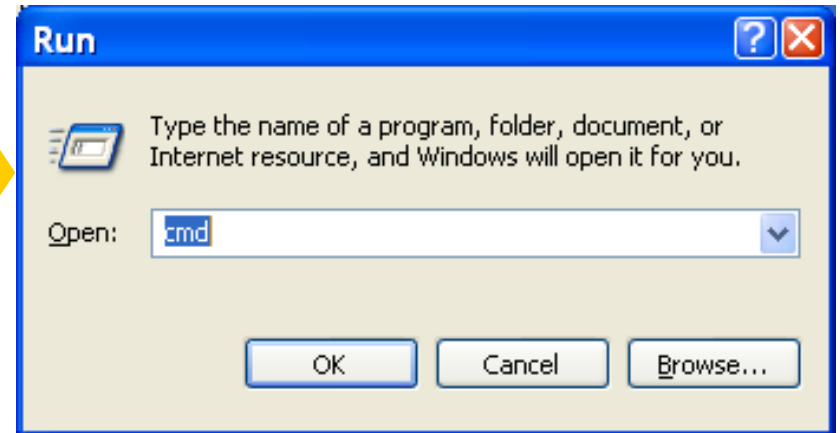
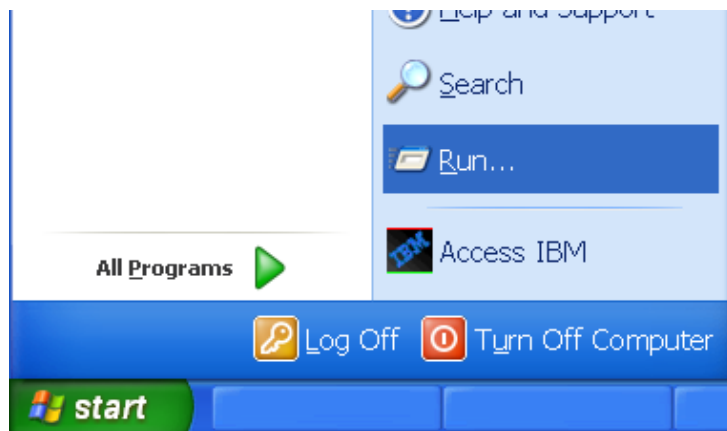
Server



- Output could be HTML, JavaScript, XML, . . .

Debugging PHP

- Sometimes you don't know where the problem is - the PHP code? The server? The browser?
- You can learn more by running PHP code via the windows DOS command line
- You will then see exactly what is being generated by the PHP engine, including the important HTTP header which you don't see when you use a browser
- The next slide shows how to access the Windows command window and move to the PHP directory



Using the Command Line

- After you have the windows command line display you can then give any PHP code to the PHP engine
- The PHP engine can execute a PHP file in the command line using a program called *php-cgi.exe*

```
C:\PHP>php-cgi.exe helloworld.php
```

- The exact output of the PHP code will then be shown, with the header
- This can help you work out what is wrong
- A complete example is given in the next slide

Example Execution of php-cgi.exe

PHP Code:

```
<html>
  <head>
    <title>
      Hello World
    </title>
  </head>
  <body>
    <p><?php
      print "Hello World";
    ?></p>
  </body>
</html>
```

header

body

PHP Output:

```
C:\PHP>php-cgi.exe helloworld.php
X-Powered-By: PHP/5.2.3
Content-type: text/html

<html>
  <head>
    <title>
      Hello World
    </title>
  </head>
  <body>
    <p>Hello World</p>
  </body>
</html>

C:\PHP>
```


X-Powered-By

- Often a server side program will tell the browser what language and version it is running in the *X-Powered-By* field
- Actually, this is not very clever, because a hacker can try to find tricks which work with that particular version
- Many systems deliberately disable this output, so the browser is not told what server side language and version is being used

```
C:\PHP>php-cgi.exe helloworld.php
X-Powered-By: PHP/5.2.3
Content-type: text/html

<html>
  <head>
    <title>
      Hello World
    </title>
  </head>
  <body>
    <p>Hello World</p>
  </body>
</html>

C:\PHP>
```


Using a HTTP Response Viewer

- Often you want to know exactly what is being sent out from a server
- You can use a HTTP viewer web site to send a request to a server and see exactly what is sent back
- If you want to test what is being sent from your own server and you don't know the English name of the machine, you can use the IP address of the server
- (These methods won't work if your server is running in a CS lab 1 machine because of the virtual image security)

HTTP Response Viewer

<http://www.rexswain.com/httpview.html>

Rex Swain's HTTP Viewer - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.rexswain.com/httpview.html

Customize Links Free Hotmail Windows Marketplace Windows Media Windows 303

Rex Swain's HTTP Viewer

See *exactly* what an HTTP request returns to your browser

URL

http://

Request Type

☒ GET (Header & Content)
☐ HEAD (Header only)
☐ TRACE (Reiterate request)

Version

☐ HTTP/1.0
☒ HTTP/1.1

Display Format

☒ Auto-Detect
☐ Text
☐ Hex

User-Agent (optional)

Accept-Encoding (optional)

☒ **Auto-Follow Location**

Submit

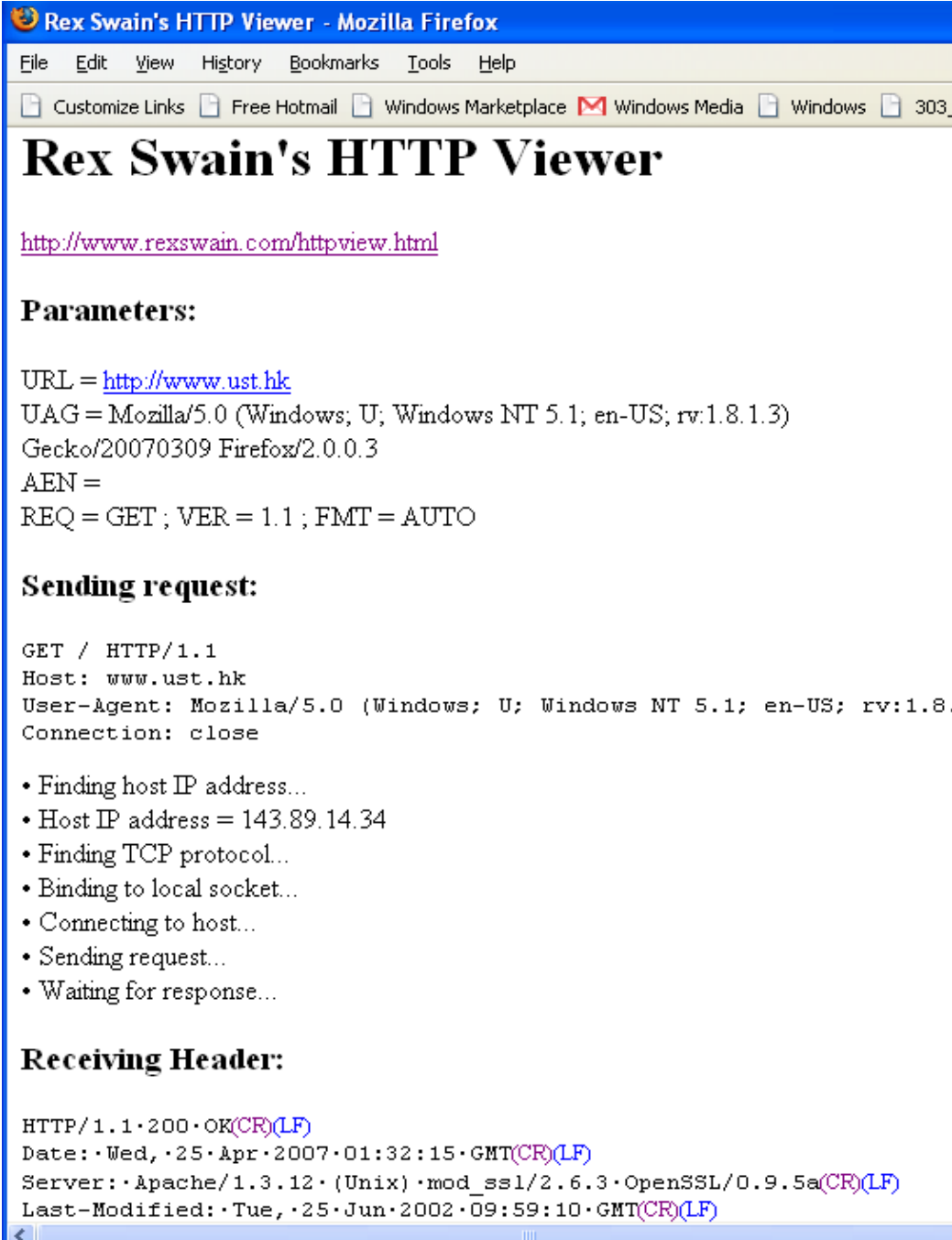
Instructions

Basically, just enter a URL and then click the **Submit** button. The fine print follows...

Done

Example 1/2

- In this example a GET request is sent to <http://www.ust.hk>
- The complete sequence of request & response is shown here



Rex Swain's HTTP Viewer - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Customize Links Free Hotmail Windows Marketplace Windows Media Windows 303

Rex Swain's HTTP Viewer

<http://www.rexswain.com/httpview.html>

Parameters:

URL = <http://www.ust.hk>
UAG = Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.3) Gecko/20070309 Firefox/2.0.0.3
AEN =
REQ = GET ; VER = 1.1 ; FMT = AUTO

Sending request:

```
GET / HTTP/1.1
Host: www.ust.hk
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.
Connection: close
```

- Finding host IP address...
- Host IP address = 143.89.14.34
- Finding TCP protocol...
- Binding to local socket...
- Connecting to host...
- Sending request...
- Waiting for response...

Receiving Header:

```
HTTP/1.1 200 OK(CR)(LF)
Date: Wed, 25 Apr 2007 01:32:15 GMT(CR)(LF)
Server: Apache/1.3.12 (Unix) mod_ssl/2.6.3 OpenSSL/0.9.5a(CR)(LF)
Last-Modified: Tue, 25 Jun 2002 09:59:10 GMT(CR)(LF)
```


Example 2/2

FileEditViewHistoryBookmarksToolsHelp

Customize LinksFree HotmailWindows MarketplaceWindows MediaWindows303_s2007_a2_mark...

ETag: "1a0a64-e4-3d183eee"(CR)(LF)
Accept-Ranges: bytes(CR)(LF)
Content-Length: 228(CR)(LF)
Connection: close(CR)(LF)
Content-Type: text/html(CR)(LF)
(CR)(LF)

End of Header (Length = 276)

- Elapsed time so far: 3 seconds
- Waiting for additional response until connection closes...

Total bytes received = 504

Elapsed time so far: 4 seconds

Content (Length = 228):

<HTML>(LF)
<HEAD>(LF)
<TITLE>HKUST·Web·Server</TITLE>(LF)
<meta·http-equiv="Content-Type"·content="text/html">(LF)
<meta·HTTP-EQUIV="Refresh"·content="0"·URL=http://www.ust.hk/en/index.html">(LF)
</HEAD>(LF)
(LF)
<BODY·BGCOLOR="#FFFFFF">(LF)
(LF)
</BODY>(LF)
</HTML>(LF)
(LF)

Done

Total elapsed time: 4 seconds

<|||

Browser Redirection

- The HTML code in the previous slide shows one way to tell the browser to go to a different page, using HTML alone:

```
<meta http-equiv="refresh" content=
    "0; URL=http://www.google.com">
```

- A different method is by generating an appropriate message in the HTTP header. For example:

```
<?php
/* Redirect browser to another page */
header("Location: http://www.yahoo.com/");

/* Any PHP code below will not be appropriate */
/* Any HTML code below will not be seen in the browser */
echo "<h1>My web page. . .</h1>";
?>
```


Browser Redirection

- In this example, the instruction to go to a different web page is given in the HTTP header
- The browser sees the instruction and immediately goes to the new web site (any HTML is ignored)

Browser sees
this and goes
to new
location

Ignored
by browser

```
C:\PHP>php-cgi.exe location.php
Status: 302
X-Powered-By: PHP/5.2.3
Location: http://www.yahoo.com/
Content-type: text/html

<h1>My web page. . .</h1>
C:\PHP>_
```


Automatic Cookie Transfer

- Previously we did JavaScript programming of cookies - now we consider server-side programming of cookies
- For example, assume that the web site w3.org has previously set a cookie in the client side (in other words, in the browser)
- The next time that the browser visits a web page from w3.org the browser will send the cookie information to the web site together with the http command
- i.e. when 'GET index.html' is sent by the browser the cookie information is also sent to the server

Setting a Cookie

- Server side code sends cookie instructions to the browser in the HTTP message header
- Because the HTTP header is sent before the actual HTML is sent, this means you can't have any cookie instructions in your server side code if it has already generated the HTTP header
- In other words, if the PHP code has already output some HTML, it's too late to output a cookie

Setting a Cookie – Correct Example

```
<?php
```

```
// This example will correctly create a cookie.
```

```
// The cookie will expire 1 hour later.
```

```
setcookie("last_message", "Bye for now", time()+3600);
```

```
?>
```


Setting a Cookie - Server Output

```
C:\PHP>php-cgi.exe 02_set_cookie.php
X-Powered-By: PHP/5.2.3
Set-Cookie: last_message=Bye+for+now; expires=Mon, 06-Apr-2009 05:10:25 GMT
Content-type: text/html

C:\PHP>
```

- This shows the HTTP header which gets sent to the browser – the cookie is appropriately set
- In this example no HTML is generated

Another Example, With HTML Output

```
<?php
// This example will correctly create a cookie.
setcookie("last_message", "Bye for now", time()+3600);
?>
```

```
<html>
<head>
</head>
```

```
<body>
<p>This is the web page.</p>
</body>
</html>
```

- This is another example of correctly generating a cookie
- In this example HTML is also sent to the browser in addition to the cookie instruction

Setting a Cookie – Incorrect

```
<html>  
<head> </head>
```

Here the cookie cannot be set,
because the HTTP header has
already been sent to the browser

```
<?php  
// This example will correctly create a cookie.  
setcookie("last_message", "Bye for now", time()+3600);  
?>  
<body>  
<p>This is the web page.</p>  
</body>  
</html>
```


PHP Output for Previous Slide Code

```
C:\PHP>php-cgi.exe 04_set_cookie_incorrect.php
```

```
X-Powered-By: PHP/5.2.3
```

```
Content-type: text/html
```

The HTTP header

```
<html>
```

```
<head>
```

```
</head>
```

```
<br />
```

```
<b>Warning</b>: Cannot modify header information - headers already sent by <output started at C:\PHP\04_set_cookie_incorrect.php:5> in <b>C:\PHP\04_set_cookie_incorrect.php</b> on line <b>12</b><br />
```

```
<body>
```

```
<p>This is the web page.</p>
```

```
</body>
```

```
</html>
```

The main part of the message, including error message from the PHP engine

```
C:\PHP>
```


Accessing a Cookie in PHP

- In PHP cookie values are provided to the programmer in an array called `_COOKIE`

```
<?php
```

```
    echo $_COOKIE["last_message"];
```

```
?>
```


Printing all Cookies

- Useful for debugging:

```
print_r($_COOKIE); // print all cookies
```

- ‘print_r’ means ‘print recursively, in a nice easy to understand format’

Example of print_r

```
<pre>
```

```
<?php
```

```
$a = array ('a' => 'apple', 'b' => 'banana', 'c' => array ('x', 'y', 'z'));
```

```
print_r ($a);
```

```
?>
```

```
</pre>
```

- <pre> is html, meaning 'previously formatted'
- Anything in <pre> is therefore shown on the web page with the same spacing as the source text

```
<pre>
```

```
Array
```

```
(
```

```
  [a] => apple
```

```
  [b] => banana
```

```
  [c] => Array
```

```
  (
```

```
    [0] => x
```

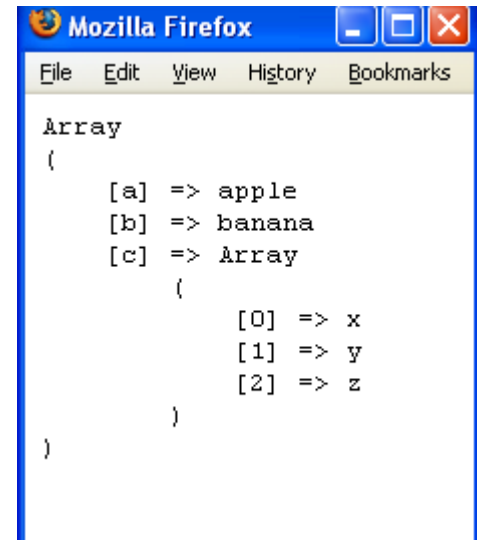
```
    [1] => y
```

```
    [2] => z
```

```
  )
```

```
)
```

```
</pre>
```



Deleting a Cookie

- In PHP, you can delete a cookie by giving it no parameters
- The browser assumes that means you don't want it any more, and deletes it

```
<?php setcookie ("last_message"); ?>
```

```
C:\PHP>php-cgi.exe 08_delete_cookie.php
X-Powered-By: PHP/5.2.3
Set-Cookie: last_message=
Content-type: text/html

C:\PHP>
```