



COMP 4021

Internet Computing

PHP

Dr. Kenneth LEUNG

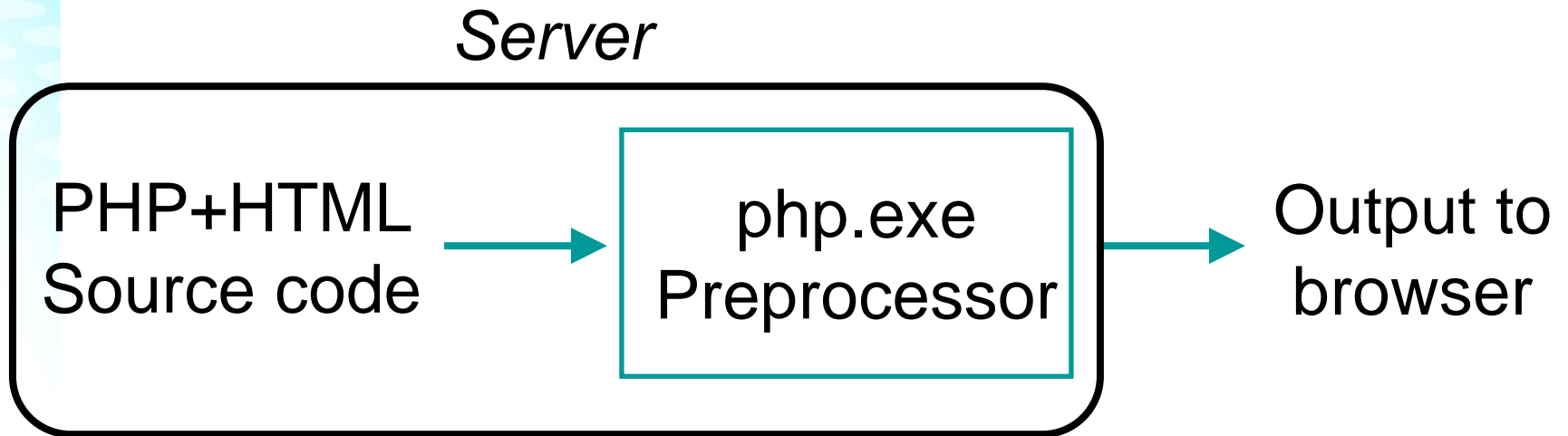
Slides created by Dr. David ROSSITER

PHP

'Hypertext' basically means 'has links' i.e. HTML

- PHP = Hypertext Preprocessor
- 'Preprocessor' means that a file gets processed on the server before it gets sent out to the browser
- So PHP is a server-side scripting language
- You usually embed PHP commands *inside* HTML
- Also, you could have a file which contains only PHP

PHP Operation



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

Configuring Apache For PHP

- To install PHP in an Apache Web Server, you have to add a few lines in the configuration file *httpd.conf*

```
ScriptAlias /php/ "C:/php/"
```

```
AddType application/x-httpd-php .php
```

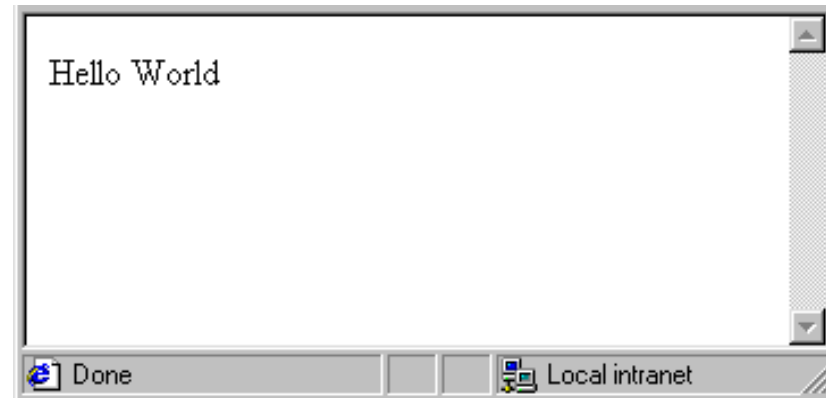
```
AddType application/x-httpd-php-source .phps
```

```
Action application/x-httpd-php "/php/php.exe"
```

- After Apache is configured, every file with an extension of *php* or *phps* (=PHP source code) is processed by *C:/php/php.exe* before the output is sent to the browser

Simple PHP Example

```
<html>
  <head>
    <title>PHP Test</title>
  </head>
  <body>
    <?php echo "Hello World";?>
  </body>
</html>
```

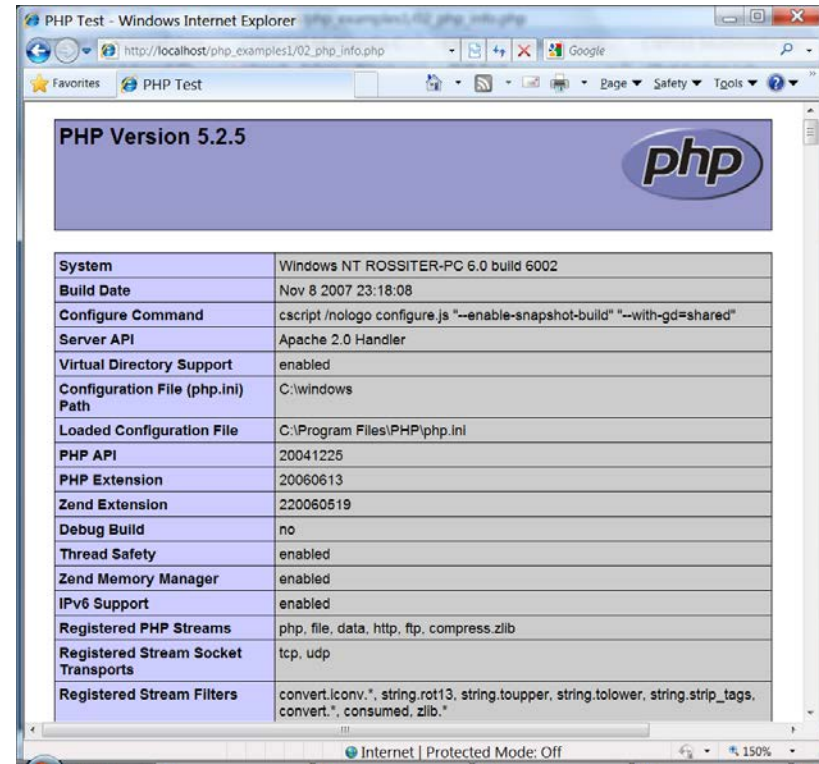


Demo – 01_starting_example.php

PHP General Information

- The instruction `phpinfo()` simply displays some useful information about PHP

`<?php echo phpinfo() ?>`



PHP Version 5.2.5

System	Windows NT ROSSITER-PC 6.0 build 6002
Build Date	Nov 8 2007 23:18:08
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--with-gd=shared"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\windows
Loaded Configuration File	C:\Program Files\PHP\php.ini
PHP API	20041225
PHP Extension	20060613
Zend Extension	220060519
Debug Build	no
Thread Safety	enabled
Zend Memory Manager	enabled
IPv6 Support	enabled
Registered PHP Streams	php, file, data, http, ftp, compress.zlib
Registered Stream Socket Transports	tcp, udp
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, zlib.*

Demo – 02_php_info.php

PHP Types

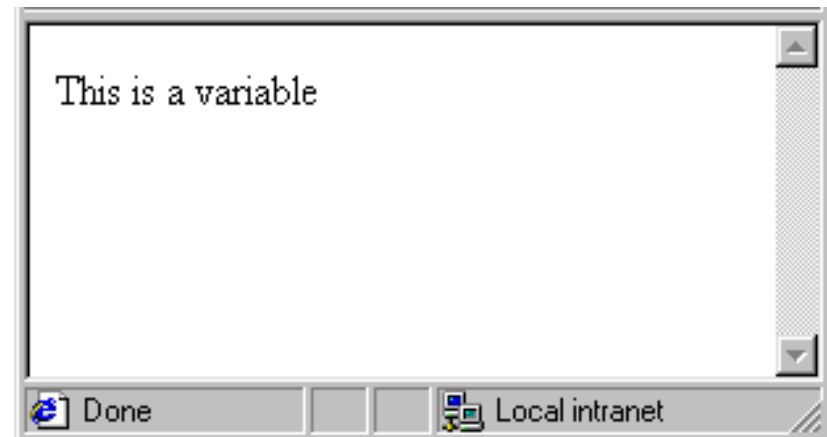
- PHP is a large language, supporting all the usual things you would expect to find:
 - arrays
 - floating-point numbers
 - integers
 - objects
 - strings

PHP Variables

- Simple example:

```
<?php  
$va= "This is a variable";  
echo $va;  
?>
```

- A PHP variable always begins with a dollar-sign



Demo – 03_variable.php

PHP Multi-Dimension Arrays

- Like many languages, PHP supports single dimension arrays as well as multi-dimension arrays. Examples:

`$a[1] = $f;` # one dimensional

`$a[1][0] = $f;` # two dimensional

`$a["foo"][2] = $f;` # two dimensional

you can mix numeric and associative indices

`$a["foo"][4]["bar"][0] = $f;` # four dimensional

If Statement

strstr(str1, str2) is a function for searching a string for another string

```
<?php if( strstr($HTTP_USER_AGENT,"MSIE")) {?>  
    <center>  
        <b>You are using Internet Explorer</b>  
    </center>  
<?php } else {?>  
    <center>  
        <b>You are not using Internet Explorer</b>  
    </center>  
<?php } ?>
```

If Statement – Checking the browser

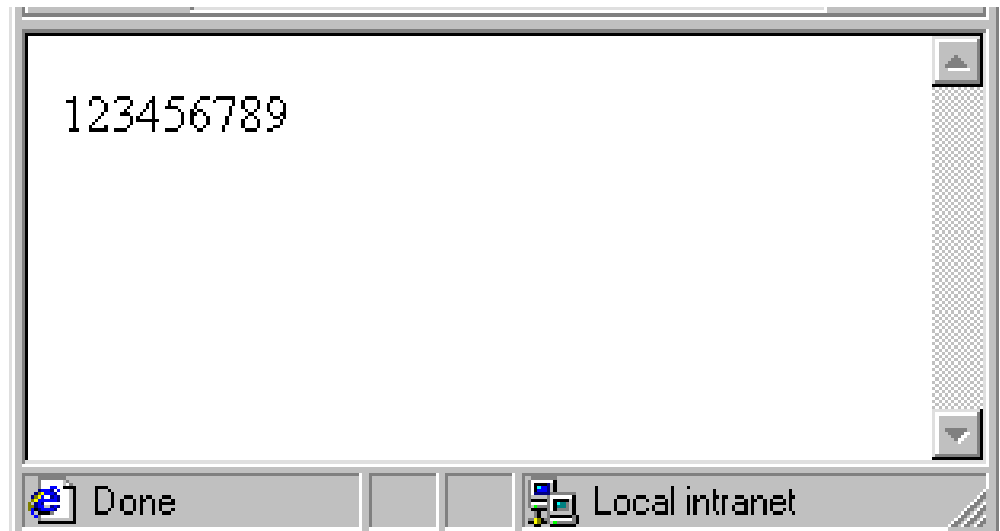


Demo – 04_check_browser.php

For Statement

- We want to print 1, 2, 3, ..., 9

```
<?php
  for ($i=1;$i<10;$i++)
    print $i;
?>
```



Demo – 05_for_loop.php

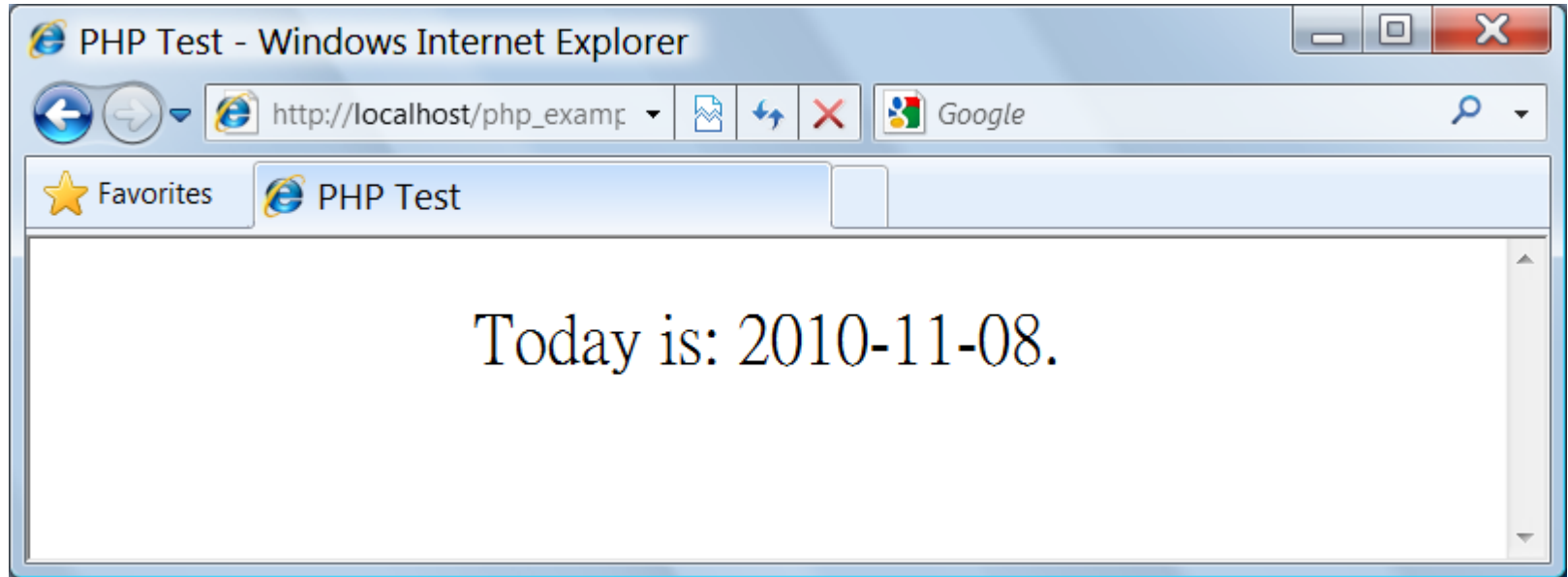
Functions

- Example:

```
<?php
function print_date(){
    $today = date("Y-m-d");
    print "<center>Today is:
        $today.</center>";
}
print_date();
?>
```

date(str) returns a string containing a date formatted according to the parameters

Functions



Demo – 06_function.php

Functions – Returning a Value

- General syntax for defining a function:

```
function name($arg_1, $arg_2, ..., $arg_n)
{
    //put your code here
    return value; //return a value
}
```

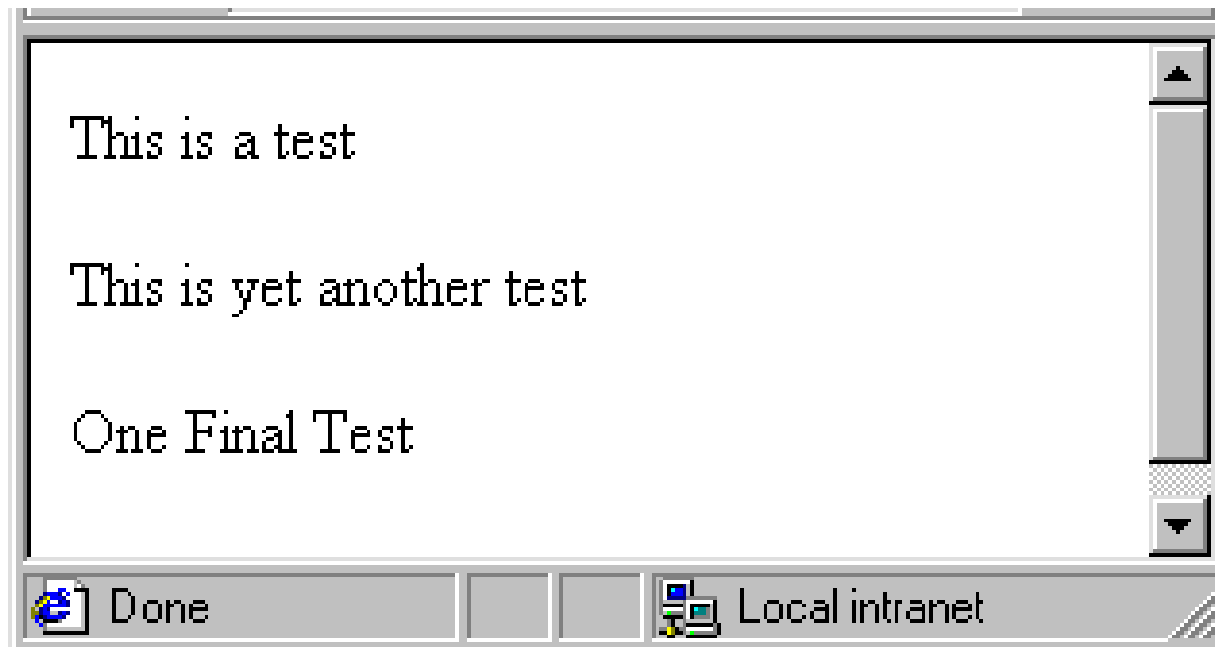
- The `return` statement will return a value, like C

PHP Comments

- Example:

```
<?php echo "<p>This is a test</p>";  
    // This is a one-line c++ style comment  
    /* This is a multi line comment  
    which includes these words */  
    echo "<p>This is yet another test</p>";  
    echo "<p>One Final Test</p>";  
    # This is the last comment  
?>
```


Comments - Example



Demo – 07_comments.php

Handling Forms

- Handling HTML forms is very easy with PHP
- Example on the next few slides

Handling Forms – HTML code

```
<form action="09_action.php" method="post">  
  Your name: <input type="text" name="name">  
  You age: <input type="text" name="age">  
  <input type="submit">  
</form>
```

Handling Forms – PHP code

- The form data gets passed to the following PHP:

```
<html>
```

```
<head><title>PHP Test</title></head>
```

```
<body>
```

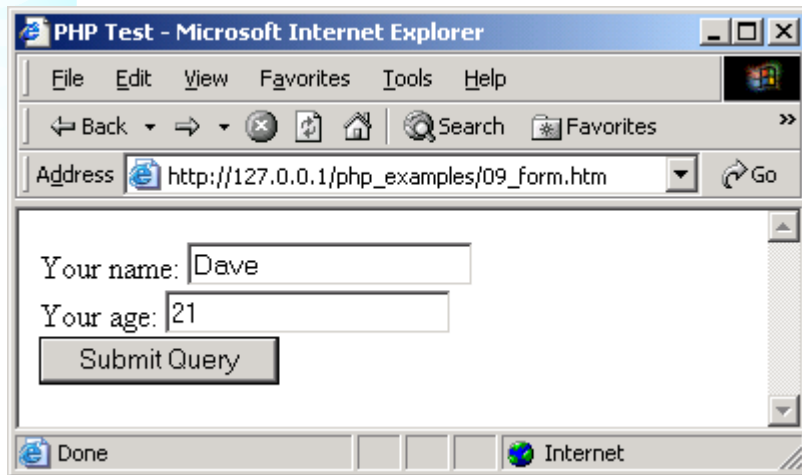
```
  Hi <?php echo $_POST['name'];?>. You are  
    <?php echo $_POST['age'];?> years old.
```

```
</body>
```

```
</html>
```

An Illustration of the Process

Open the form, input your details,
submit the data



Demo – 09_form.htm

PHP program response



Demo – 09_action.php

Handle File Upload – HTML code

- Example form:

```
<form enctype="multipart/form-data"
      action="10_file_upload.php" method="post">
```

```
<input type="hidden"
      name="MAX_FILE_SIZE" value="640000">
```

Send this file:

```
<input name="userfile" TYPE="file">
<input type="submit" VALUE="Send File">
</form>
```

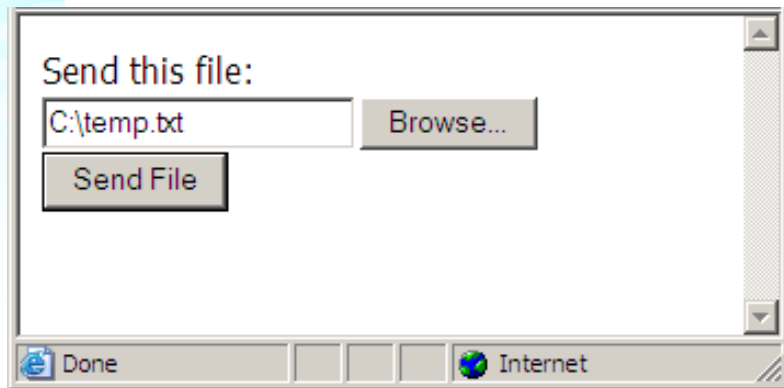
Handle File Upload – PHP code

- The form data is passed to this PHP:

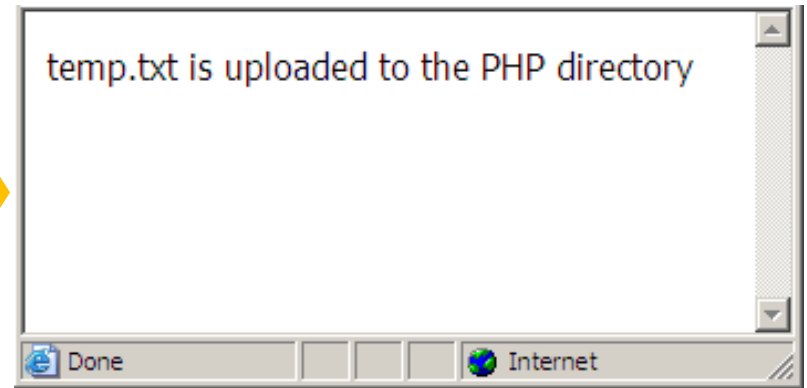
```
<?php
    move_uploaded_file(
        $_FILES['userfile']['tmp_name'],
        $_FILES['userfile']['name']);

    echo $_FILES['userfile']['name'],
        " is uploaded to the PHP directory";
?>
```

An Illustration of the Process



Demo – 10_file.html



Demo – 10_file_upload.php