COMP 4021
Internet Computing

PHP 3

David Rossiter

Overview

- This presentation considers
  - PHP Classes
  - Mime types
  - PHP and XML/SVG
  - PHP to JavaScript
    - JavaScript to PHP
    - Handling whole files
    - Lower level file control

PHP Classes

- Like many programming languages, PHP supports classes and objects, i.e.,

```
class xmlHandler {
    var $filename;
        // the filename of the XML file
    …
    // constructor
    function xmlHandler($filename) {
        $this->filename = $filename;
        $this->root = null;
        $this->doc = null;
    }
    …
}
```

Attributes (=parameters)

Methods (=functions)

This means typical object oriented style techniques can be used - we use some of them in the Assignment 3

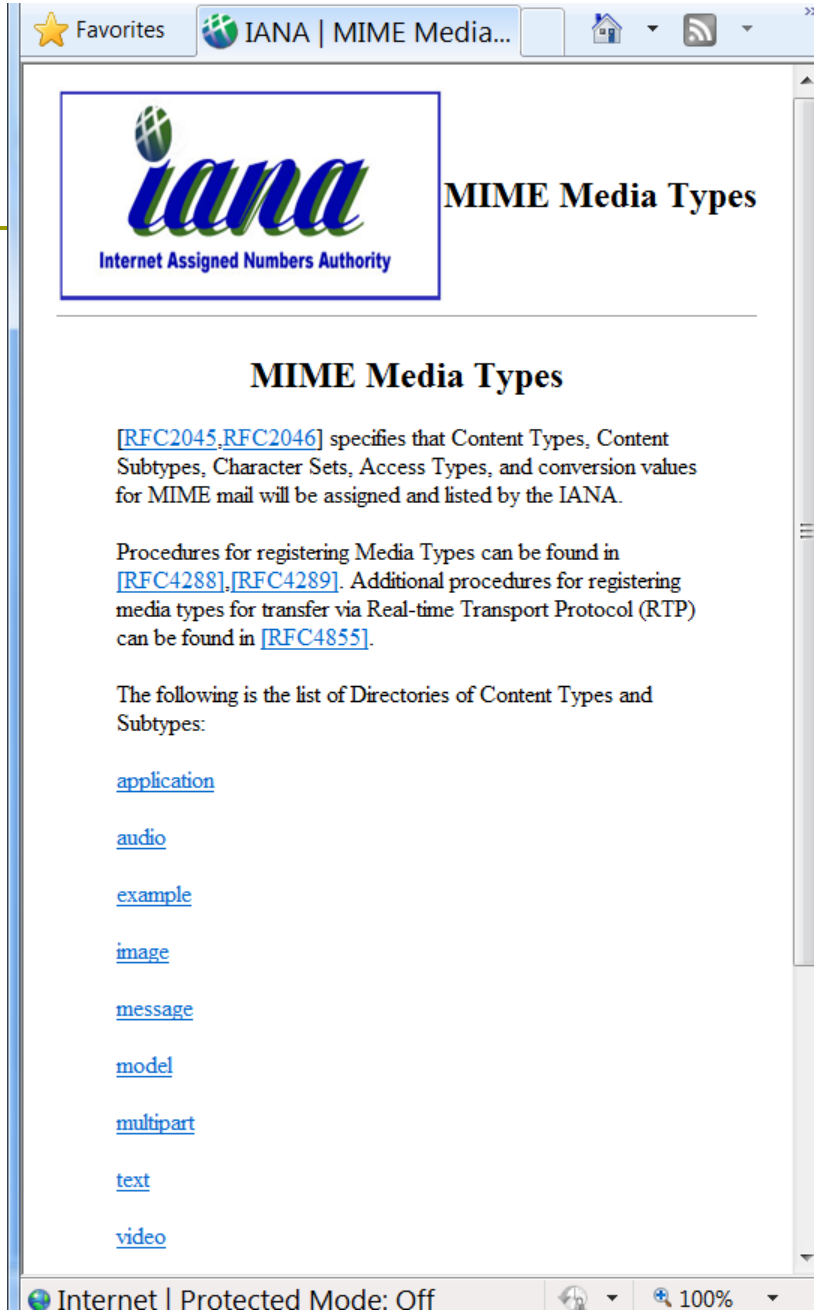
Generating XML

- PHP can be used to generate any kind of text
- So generating any HTML, XML, or XML based code (such as SVG, MathML, etc) is easy for PHP

- Remember that the MIME type must be correctly stated at the start, in the header
- This is so that the browser knows what kind of data is coming from the server, so it can do something appropriate

MIME Types

- The official MIME type list for all web data is http://www.iana.org/assignments/media-types/
- However, this web site is often out of date – sometimes it is better to search the web



Example of Generating XML in PHP

```
<?php
    header ("Content-Type: text/xml");
    print ('<?xml version="1.0" encoding="utf-8"?>');
?>
<xml>
  <message>This file will be interpreted as XML</message>
</xml>
```

The browser will receive the XML and do something with it, e.g., display it

Generating SVG with PHP and CSS

```
<?php header("Content-type: image/svg+xml"); ?>

<?xml-stylesheet href="style.css" type="text/css"?>
<svg . . . >
        . . .
        <text x="6" y="30">Text!</text>
    . . .
</svg>
```

Tell the browser to load the style sheet file

This is just some example SVG

PHP and SVG with Embedded CSS

```
<?php header("Content-type: image/svg+xml"); ?>
<svg . . . >
<style type="text/css">
<?php
   include("style.css");
?>
</style>
. . .
<text x="6" y="30">Text!</text>'
. . .
</svg>
```

- The style sheet file is ‘copy and pasted’ by the PHP directly into the SVG sent out to the browser
- It is a different style of web programming compared to the last slide

PHP to JavaScript

- How to transfer a variable from PHP to JavaScript?
- Example code:

```
<?php print "var word1='$word1';";  ?>
```

- The basic idea is for the PHP code to create a JavaScript variable of the same name, and give it the same value

Complete Example

```
<html> <body>

<?php
$name = 'donald';
?>
<script>
<?php print "var name='$name';";  ?>
   alert("The variable name has now been transferred
         from php to JavaScript, it has the value " + name );
</script>
</body> </html>
```

A variable in PHP

```
<script>
var name='donald'
alert("The … " + name)
</script>
```

JavaScript to PHP

- How to transfer a variable from JavaScript to PHP?
- Remember how PHP is executed (once only)
- The browser never sees the actual PHP code
- It only sees the generated output i.e. HTML
- Therefore the only way to transfer a value back to PHP is to send the data back to the server, to your PHP code (similar to sending form data)

Outputting Whole Files

- If you are simply outputting whole files from the server to the client then this is relatively easy

```
<?php
    // Tell the browser a PDF file is coming
    header('Content-type: application/pdf');

    // For the browser, it will be called downloaded.pdf
    header('Content-Disposition: attachment;
          filename="downloaded.pdf"');

    // The PDF source (on the server) is original.pdf
    readfile('original.pdf');  // File is sent to the browser
?>
```

Sending File from Server to Browser

- For the previous slide, the PDF file is transmitted from server to client with an appropriate header
- This is totally a PDF transmission, there's no html involved
- On the client side, the browser receives the file and handles it appropriately i.e. opens it with a PDF reader program

Lower Level File Control

- You could use file read commands for more exact control
- The following code sends an image to the browser

```
<?php
$fp = fopen("flowers.jpg", "rb");                    // Open the file

$mybuf = fread($fp, filesize("flowers.jpg"));  // Read file, store in an array
fclose($fp);   //Close the file

header("Content-type: image/jpeg");          // Give browser the MIME

print("$mybuf");   // Output the JPEG image
?>
```

High Level File Access

- Perhaps the easiest way to give the user access to a file is simply to provide a link to the file
- Example:

```
<?php
$filelist = array("flowers.jpg", "car.jpg", "dog.doc");
for ($i = 0; $i < count($filelist); $i++) {
    $file = $filelist[$i];
    print "Click <a href='$file' target='_blank'>here</a>  to open file ".($i + 1);
    print "<br />";
}
?>
```

Click here to open file 1
Click here to open file 2
Click here to open file 3

Take Home Message

- Make use of PHP processing power to create dynamic content and control browsers
- PHP + JavaScript can be interleaved to generate the desired content (HTML + JavaScript code)
- Appropriate header must be created to tell the browser about the content type