Web Development II

Department of Software and Computing Systems

Math Functions
HTTP Headers
Uploading files to the server

Sergio Luján Mora
Jaume Aragonés Ferrero
Department of Software and Computing Systems
Math functions (I)

- Math Constants:
  - M_PI: 3.1415… (pi)
  - M_E: 2.7182… (e)
  - M_LOG2E: 1.4426…(log_2 e)
  - M_LOG10E: 0.4342…(log_10 e)
  - M_LN2: 0.6931…(log_e 10)
  - …
Math functions (II)

- Functions:
  - `Abs($num)`: Absolute value
  - Trigonometric functions: `acos`, `asin`, `atan`, `cos`, `sin`, `tan`, ...
  - `Ceil($num)`: Returns the next highest integer value by rounding up $num if necessary.
  - `floor($num)`: Returns the next lowest integer value by rounding down $num if necessary.
  - `Round($num, $precision)`: Returns the rounded value of $num to specified $precision (number of digits after the decimal point)
  - Logarithmics functions: `log`, `log10`, `log1p`.

Math functions (III)

- `Max()/min()`: If the first and only parameter is an array, returns the highest/lowest value in that array. If at least two parameters are provided, returns the biggest/smallest of these values.
- `rand(void), rand($max, $min)`: If called without the optional _min_, _max_ arguments `rand()` returns a pseudo-random integer between 0 and `getrandmax()`.
- `Srand($seed)`: Seeds the random number generator with seed or with a random value if no seed is given.
  - As of PHP 4.2.0, there is no need to seed the random number generator
- Base Conversion functions: `Decbin`, `dechex`, `dec2oct`, `bindec`, `hexdec`, etc.
- `Is_nan($num)`: Checks whether $num is 'not a number'.
- `Is_finite($num)`: Checks whether $num is a legal finite on this platform.
Math functions(IV)

• Examples:

```php
<?
    echo M_PI . '<br />';
    echo M_E . '<br />';

    echo round(M_PI, 2) . '<br />';
    echo floor(9.99) . '<br />';
    echo rand(1, 10) . '<br />';

    echo max(1, 3, 2, 6, 9, 8) . '<br />';
?>
```

Exercises

• Create a script generating random number everytime is loaded
• Create a script: with a given number, obtain its rounded value (2 decimals), and convert its integer part to binary and hexadecimal.
HTTP Headers (I)

- header() send raw HTTP headers to the client:
  
  ```php
  header("headerName: value");
  ```
- header() must be called before any output is sent.
- Headers:
  - cache-control, expires, last-modified, etc.
  - HTTP commands and errors.
  - Content definitions: type, length, etc.
  - Cookies
- http://en.wikipedia.org/wiki/Meta_tag

HTTP Headers (II)

- Examples:
  
  ```php
  header("location: http://www.ua.es");
  header("HTTP/1.0 404 Not Found");
  header("Pragma: no-cache");
  header("Content-type: text/html");
  header('Content-type: application/vnd.ms-excel');
  header('Content-Disposition: attachment; filename="myFile.ext"');
  ```
Web development II

Predefined variable $_FILE

- An associative array of items uploaded to the current script via the HTTP POST method.
- Usage: managing uploaded files from a client form to the server file system.
- Every HTML form must include an attribute and its value as follows:
  - `enctype="multipart/form-data"`
- In file `php.ini` we can find parameters to configure this feature:
  - `file_uploads`, Whether to allow HTTP file uploads.
  - `upload_tmp_dir`, Temporary directory for HTTP uploaded files
  - `upload_max_filesize`, Maximum allowed size for uploaded files

Web development II

Predefined variable $_FILE

- `$_FILES['myFile']['name']`: original name of the file in client file system.
- `$_FILES['myFile']['tmp_name']`: The temporary filename of the file in which the uploaded file was stored on the server.
- `$_FILES['myFile']['size']`: The size, in bytes, of the uploaded file.
- `$_FILES['myFile']['type']`: The mime type of the file, if the browser provided this information.
- `$_FILES['myFile']['error']`: The error code associated with this file upload.
Example uploading files (I)

```html
...<body>
<html>
Upload a picture:<h1>
<form action="photo_insert.php" method="post" enctype="multipart/form-data">
  
  <?
  echo "Picture name:<input type='text'
    name='pName' />">
  echo "<br />
  echo "Picture file: <input type='file'
    name='pfile' />">
  echo '<br />
  ?>
  <input type="submit" value="Send picture" />
</form></html>...
```

Example uploading files (II)

```php
...<body>
  
  <?
  echo "Name:     " . $_FILES['pFile']['name'] . "<br />
  echo "Temp name: " . $_FILES['pFile']['tmp_name'] . "<br />
  echo "Size:     " . $_FILES['pFile']['size'] . "<br />
  echo "Type:     " . $_FILES['pFile']['type'] . "<br />
  ?>
  
  Define ("PICDIR", "c:\pictures\")
  $uploadfile = PICDIR . $_FILES['pFile']['name'];

  if(move_uploaded_file($_FILES['pFile']['tmp_name'],
    $uploadfile))
    echo ' ok<br/>
  else
    echo ' error<br/>
  ?>
</body></html>
```
Web development II

Is_uploaded_file, move_uploaded_file

  - is_uploaded_file($filename), Tells whether the file was uploaded via HTTP POST
- the function is_uploaded_file() needs an argument like $_FILES['userfile']['tmp_name']
  - move_uploaded_file($file, $destination), Moves an uploaded file to a new location.
- This function checks to ensure that the file designated by filename is a valid upload file. If the file is valid, it will be moved to the filename given by destination.

Web development II

Predefined variable $_FILE

- Precautions, be careful with:
  - Permissions of writing in the temporal directory
  - Permissions of writing in the destination file directory
  - Files from users containing malware
    - Trojans, worms, viruses, scripts, executables, etc.
  - Users from others platforms:
Exercise

- Create a PHP program that allows to upload a picture to the server. The user will be able to propose the name the picture will have.
  - You need to code two PHP script:
    - The one with the HTML form and all the necessary form controls
    - The response PHP script which will receive the file and will put it in the destination folder with the name proposed by the user. This script will show the result of this operation, whether has worked OK or if any error has occurred.