

PI2 / DW2

I.16. PHP – Ficheiros

Nuno Miguel Gil Fonseca
nuno.fonseca@estgoh.ipc.pt

PHP - *Hypertext Preprocessor*

■ Ficheiros

■ *Upload* de ficheiros em PHP

- Pelos mais diversos motivos podemos necessitar de fazer *upload* de ficheiros para o nosso site:
 - Fotos de perfil;
 - Ficheiros com informação;
 - Etc.
- Para tal, recorreremos à *tag* HTML `<input type="file" ...>`
- Quando o formulário é submetido, o(s) ficheiro(s) estarão disponíveis na variável `$_FILES` e não em `$_POST` ou `$_GET`.
- Em todo o caso, o(s) ficheiro(s) serão guardados numa pasta temporária e teremos obrigatoriamente que os mover para o destino pretendido.

PHP - *Hypertext Preprocessor*

- Ficheiros
 - *Upload* de ficheiros em PHP

```
...  
<form action="xpto.php" method="POST" enctype="multipart/form-data">  
  <input type="hidden" name="MAX_FILE_SIZE" value="3000000">  
  <input name="myfile" type="file">  
  <input type="submit" value="enviar">  
</form>  
...
```

```
<?php  
if (isset($_FILES["myfile"])) {  
  
    // Investigar se os ficheiros são válidos  
  
    if (move_uploaded_file($_FILES["myfile"]["tmp_name"], "." . $_FILES["myfile"]["name"])){  
        print("Nome do ficheiro recebido:" . $_FILES["myfile"]["name"] . "<br />");  
        print("Tamanho do ficheiro recebido:" . $_FILES["myfile"]["size"] . " bytes");  
    } else{ print ("Ocorreu um erro a copiar o ficheiro"); }  
}  
?>
```

O nome definitivo do ficheiro deverá ser modificado para um valor que não se repita... exemplo: concatenação de um *timestamp* com um *id*...

PHP - Hypertext Preprocessor

■ Ficheiros

■ Upload de ficheiros em PHP - vários ficheiros

```
...  
<form action="xpto.php" method="POST" enctype="multipart/form-data">  
    <input type="hidden" name="MAX_FILE_SIZE" value="30000000">  
    <input name="myf[]" type="file"><br>  
    <input name="myf[]" type="file"><br>  
    <input name="myf[]" type="file"><br>  
    <input type="submit" value="enviar" >  
</form>  
...
```

```
<?php  
$number_of_files = count($_FILES["myf"]["name"]);  
for($i = 0; $i < $number_of_files; $i++){  
    if(strlen($_FILES["myf"]["name"][$i])>0){  
        if(move_uploaded_file($_FILES["myf"]["tmp_name"][$i], "./".$_FILES["myf"]["name"][$i])){  
            print("Nome do ficheiro recebido:".$_FILES["myf"]["name"][$i]."<br>");  
            print("Tamanho do ficheiro recebido:".$_FILES["myf"]["size"][$i]. " bytes<br><br>");  
        } else{ print ("Ocorreu um erro"); }  
    }  
}  
?>
```

O nome definitivo do ficheiro deverá ser modificado para um valor que não se repita... exemplo: concatenação de um *timestamp* com um *id*...

PHP - *Hypertext Preprocessor*

■ Ficheiros

■ *Upload* de ficheiros em PHP - vários ficheiros (HTML5)

```
...  
<form action="xpto.php" method="POST" enctype="multipart/form-data">  
    <input type="hidden" name="MAX_FILE_SIZE" value="30000000">  
    <input name="myf[]" type="file" multiple ><br>  
    <input type="submit" value="enviar" >  
</form>  
...
```

```
<?php  
$number_of_files = count($_FILES["myf"]["name"]);  
for($i = 0; $i < $number_of_files; $i++){  
    if(strlen($_FILES["myf"]["name"][$i])>0){  
        if(move_uploaded_file($_FILES["myf"]["tmp_name"][$i], "./".$_FILES["myf"]["name"][$i])){  
            print("Nome do ficheiro recebido:".$_FILES["myf"]["name"][$i]."<br>");  
            print("Tamanho do ficheiro recebido:".$_FILES["myf"]["size"][$i]. " bytes<br><br>");  
        } else{ print ("Ocorreu um erro"); }  
    }  
}  
?>
```

O nome definitivo do ficheiro deverá ser modificado para um valor que não se repita... exemplo: concatenação de um *timestamp* com um *id*...

PHP - *Hypertext Preprocessor*

■ Ficheiros

■ *Upload* de ficheiros em PHP

- Se estivermos a fazer o *upload* de uma imagem, podemos pretender fazer um *preview* da mesma, antes de enviar para o servidor.
- Para tal, necessitaremos de recorrer a *JavaScript*:

```
<form action="">
  Nome: <input type="text" name="nome" value=""><br>
  <img id="output" style="max-width: 200px" ><br>
  Foto: <input id="foto" name="foto" type="file" accept="image/*" onchange="loadFile(event)">
  <br>
  <input type="submit" value="submeter">
</form>
<script>
  var loadFile = function(event) {
    var output = document.getElementById('output');
    output.src = URL.createObjectURL(event.target.files[0]);
  };
</script>
```

<https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/accept>

PHP - *Hypertext Preprocessor*

■ Ficheiros

■ *Upload* de ficheiros em PHP

- Os diferentes valores das variáveis associadas a cada ficheiro *uploaded*
 - `$_FILES["myfile"]["name"]` - O nome original do ficheiro no computador do cliente.
 - `$_FILES["myfile"]["type"]` - O mime-type do ficheiro, se o browser deu esta informação. Um exemplo pode ser *"image/gif"*.
 - `$_FILES["myfile"]["size"]` - O tamanho, em bytes, do ficheiro.
 - `$_FILES["myfile"]["tmp_name"]` - O nome temporário do ficheiro, como foi guardado no servidor.
 - `$_FILES["myfile"]["error"]` - O código de erro associado a este upload de ficheiro.
 - Lista de erros em: <http://pt.php.net/manual/en/features.file-upload.errors.php>

PHP - *Hypertext Preprocessor*

■ Ficheiros

■ *Upload* de ficheiros em PHP + MySQL

- Muitas das vezes necessitamos guardar informação sobre os ficheiros que acabamos de carregar em bases de dados
- A maneira mais simples de o fazer, passa por guardar na base de dados o nome do ficheiro em questão.

Utilizadores
Nome - VARCHAR(100)
Email - VARCHAR(100)
...
Foto - VARCHAR(100)

Exemplo de definição da tabela

Utilizadores
José Armando Costa
jac@xpto.pt
...
Fotografia_do_jac.gif

Exemplo de um registo

PHP - *Hypertext Preprocessor*

■ Ficheiros

■ *Upload* de ficheiros em PHP + MySQL

- Posteriormente, para aceder aos ficheiros carregados e guardados na base de dados basta fazer algo como:
 - Imagens

```
...
<?php
// assumindo que obtemos a informação da base de dados numa
// variável chamada registo
?>

" alt="fotografia">

" alt="fotografia">

" alt="fotografia">
...
```

PHP - *Hypertext Preprocessor*

■ Ficheiros

■ *Upload* de ficheiros em PHP + MySQL

- Posteriormente, para aceder aos ficheiros carregados e guardados na base de dados basta fazer algo como:
 - Outros ficheiros

```
...
<?php
// assumindo que obtemos a informação da base de dados numa
// variável chamada registo
?>

<a href="./ficheiros/<?php print($registo["ficheiro"])?>">Download</a>

<a href="./ficheiros/<?php print($registo["ficheiro"])?>" download>Download</a>

...
```

PHP - *Hypertext Preprocessor*

- **Ficheiros**

- Funções de manipulação de ficheiros
- Mover/renomear ficheiros

```
<?php
    $filename2 = $filename . ".old";
    rename($filename, $filename2);
?>
```

- **Copiar ficheiros**

```
<?php
    $filename2 = $filename . ".old";
    copy($filename, $filename2);
?>
```

PHP - *Hypertext Preprocessor*

- **Ficheiros**
 - **Funções de manipulação de ficheiros**
 - Apagar ficheiros

```
<?php
    if (unlink($filename)) {
        print("Ficheiro $filename apagado!\n");
    } else {
        print("Ficheiro $filename não apagado!\n");
    }
?>
```

- Apenas permite apagar ficheiros, não diretorias!

PHP - *Hypertext Preprocessor*

- **Ficheiros**
 - **Funções de manipulação de ficheiros**
 - Verificar se determinado ficheiro existe

```
<?php
    if (file_exists("xpto.png")) {
        print("O ficheiro existe!\n");
    } else {
        print("O ficheiro não existe!\n");
    }
?>
```

PHP - *Hypertext Preprocessor*

- **Ficheiros**

- **Funções de manipulação de ficheiros**

- `readfile()`

- Esta função permite-nos abrir um ficheiro e mostrar imediatamente o seu conteúdo...

```
<?php
    readfile("/home/paul/test.txt");
    readfile("c:\\boot.ini");
?>
```

PHP - *Hypertext Preprocessor*

- **Ficheiros**

- **Funções de manipulação de ficheiros**

- `file_get_contents ()`

- Esta função permite-nos abrir um ficheiro para leitura, e passar todo o seu conteúdo para uma variável.

```
<?php
    $filestring = file_get_contents($filename);
    print $filestring;
?>
```

PHP - *Hypertext Preprocessor*

- **Ficheiros**

- **Funções de manipulação de ficheiros**

- `file()`

- Esta função permite-nos abrir um ficheiro para leitura, e passar todo o seu conteúdo para um *array*, sendo que cada elemento deste *array* é uma linha do ficheiro.

```
<?php
    $filearray = file($filename);
?>
```


PHP - *Hypertext Preprocessor*

■ Ficheiros

■ Funções de manipulação de ficheiros

- `file_put_contents()`
 - Esta função permite-nos de uma forma simples escrever conteúdo num ficheiro.

```
<?php
    $conteudo = "abcd";
    file_put_contents($filename, $conteudo);
?>
```

- Por defeito, esta função apaga o conteúdo já existente no ficheiro. Caso não queiramos que tal aconteça, deverá ser especificado um terceiro parâmetro com o valor `FILE_APPEND`.

```
<?php
    $conteudo = "abcd";
    file_put_contents($filename, $conteudo, FILE_APPEND);
?>
```


PHP - *Hypertext Preprocessor*

- **Ficheiros**

- **Funções de manipulação de ficheiros**

- `fopen ()`

- Esta função permite-nos abrir um ficheiro de diferentes modos (leitura, escrita, ...). Devolve um *handle* que nos permite manipular o ficheiro.

```
<?php
    $handle = fopen($filename, "a");
?>
```

PHP - *Hypertext Preprocessor*

■ Ficheiros

■ Funções de manipulação de ficheiros

■ Modos de abertura dos ficheiros

- **r** – Abre o ficheiro apenas para leitura, o “ponteiro” aponta para o início do ficheiro
- **r+** - Abre o ficheiro para leitura e escrita, o “ponteiro” aponta para o início do ficheiro
- **w** - Abre o ficheiro apenas para escrita, apaga o conteúdo já existente. Cria o ficheiro, caso este não exista.
- **w+** - Abre o ficheiro para leitura e escrita, apaga o conteúdo já existente. Cria o ficheiro, caso este não exista.
- **a** – Abre o ficheiro só para escrita com o ponteiro no final do conteúdo já existente.
- **a+** – Abre o ficheiro para leitura e escrita com o ponteiro no final do conteúdo já existente.
- **x** - Abre o ficheiro apenas para escrita com o ponteiro no início do ficheiro. Dá erro caso o ficheiro já exista.
- **x+** - Abre o ficheiro para leitura e escrita com o ponteiro no início do ficheiro. Dá erro caso o ficheiro já exista.

PHP - *Hypertext Preprocessor*

■ Ficheiros

■ Funções de manipulação de ficheiros

- `fread()`
 - Esta função permite-nos ler uma determinada quantidade do conteúdo de um ficheiro.

```
<?php
    $conteudo = fread($handle, filesize($filename));
?>
```

- `fgets()`
 - Lê uma linha de um ficheiro.

```
<?php
    ...
    while (!feof($handle)) {
        $buffer = fgets($handle[, 4096]);
        echo $buffer;
    }
?>
```

PHP - *Hypertext Preprocessor*

- **Ficheiros**
 - **Funções de manipulação de ficheiros**
 - `fwrite ()`
 - Permite escrever conteúdo num ficheiro previamente aberto

```
<?php
    $handle = fopen($filename, "wb");
    $numbytes = fwrite($handle, $mystring);
    fclose($handle);
?>
```

PHP - *Hypertext Preprocessor*

- **Ficheiros**

- **Funções de manipulação de ficheiros**

- `rewind()`
 - Recoloca o ponteiro no início de determinado ficheiro

```
<?php
    $handle = fopen($filename, "w+");
    fwrite($handle, "xxxxxxxxxx\n");
    rewind($handle);
?>
```

PHP - *Hypertext Preprocessor*

■ Ficheiros

■ Funções de manipulação de ficheiros

■ `fseek()`

- Recoloca o ponteiro para uma determinada posição específica do ficheiro

```
<?php
    fseek($handle, 1);
    fseek($handle, 2, SEEK_CUR);
?>
```

- O número, especifica o número de bytes que se pretende que o ponteiro avance, tendo em conta que a posição inicial é 0.

PHP - *Hypertext Preprocessor*

- **Ficheiros**

- **Funções de manipulação de ficheiros**

- `fclose()`

- Todos os ficheiros abertos deverão ser fechados! Esta é a função que nos permite fechar os ficheiros.

```
<?php
    fclose($handle);
?>
```

PHP - *Hypertext Preprocessor*

- **Ficheiros**
 - **Funções de manipulação de ficheiros**
 - Aceder a ficheiros remotos

```
<?php
    $handle = fopen("http://www.sapo.pt", "r");
    $contents = stream_get_contents($handle);
    print($contents);
    fclose($handle);
?>
```

PHP - *Hypertext Preprocessor*

- **Ficheiros**
 - **Funções de manipulação de ficheiros**
 - Devolver informações sobre a localização de um ficheiro

```
<?php
    $fileinfo = pathinfo($filename);
    var_dump($fileinfo);
?>
```

```
array(3) {
    ["dirname"]=>
    string(22) "/home/paul/sandbox/php"
    ["basename"]=>
    string(7) "foo.txt"
    ["extension"]=>
    string(3) "txt"
}
```

PHP - *Hypertext Preprocessor*

- **Ficheiros**
 - **Funções de manipulação de ficheiros**
 - Outras funções

```
bool is_readable ( string filename)  
bool is_writable ( string filename)  
bool is_executable ( string filename)  
bool is_file ( string filename)  
bool is_dir ( string filename)  
void clearstatcache ( void )
```

PHP - *Hypertext Preprocessor*

- **Ficheiros**
 - **Trabalhar com directorias**
 - Criar directorias

```
<?php
    mkdir("/directoria_a_criar");
?>
```

- Remover directorias

```
<?php
    rmdir("/directoria_a_remover");
?>
```

- Nota: a função `rmdir()` apenas permite remover directorias que se encontrem vazias...

PHP - *Hypertext Preprocessor*

- **Ficheiros**
 - **Trabalhar com directorias**
 - Abrir uma determinada directoria

```
<?php
    $handle = opendir("/directoria_a_abrir");
?>
```

- Devolver o próximo ficheiro de determinada directoria

```
<?php
    $file = readdir($handle);
?>
```

- Fechar uma determinada directoria

```
<?php
    closedir($handle);
?>
```

PHP - *Hypertext Preprocessor*

- **Ficheiros**

- **Trabalhar com diretorias**

- O seguinte bloco de código escreve o nome de todos os ficheiros de uma determinada diretoria

```
<?php
    $handle = opendir("/fotos");
    if ($handle) {
        while (false !== ($file = readdir($handle))) {
            print ("$file <BR />\n");
        }
    }
    closedir($handle);
?>
```