

Curso POO PHP Operacións con ficheiros

Sumario

- 1 Operacións con ficheiros
 - ◆ 1.1 Sistema de ficheiros
 - ◆ 1.2 Directorios
 - ◆ 1.3 Iteradores do sistema de ficheiros

Operacións con ficheiros

Coa distribución de PHP inclúense extensións que permiten traballar directamente co sistema de arquivos:

- **Sistema de Ficheiros** forma parte do núcleo de PHP, e inclúe funcións para ler, escribir, copiar, borrar, renomear, etc., ficheiros do sistema.
- **Directorios** tamén forma parte do núcleo de PHP, e como o seu nome indica permite traballar cos directorios do sistema de ficheiros.
- **Fileinfo** inclúese por defecto coa distribución de PHP a partir da versión 5.3, aínda que debe ser activada no ficheiro de configuración php.ini. Inclúe funcións para descubrir o contido dos ficheiros.

Sistema de ficheiros

Dentro desta extensión inclúense un bo número de funcións. Entre outras, son moi empregadas **basename**, que dada unha ruta obtén o nome do arquivo ao que fai referencia (ou do último directorio se a ruta non inclúe arquivo), e **dirname**, que opera xusto ao contrario devolvendo a ruta que se lle pasa sen o arquivo (ou o directorio pai no caso de que a ruta orixinal non incluíra arquivo). Por exemplo:

```
echo basename('/var/www/blog/web/index.php'); // Amosa index.php
echo basename('/var/www/blog/web/') . PHP_EOL; // Amosa web
echo dirname('/var/www/blog/web/index.php') . PHP_EOL; // Amosa /var/www/blog/web
echo dirname('/var/www/blog/web/') . PHP_EOL; // Amosa /var/www/blog
```

A forma máis sinxela de ler o contido dun ficheiro dende PHP é empregando a función **file_get_contents**, que pasa o contido do ficheiro a unha cadea, ou a función **file**, que pasa o contido do ficheiro a un array (cada liña do ficheiro pasa a ser un elemento do array).

```
$fich = file_get_contents("proba.txt");
if($fich === false) die("ERRO! Non se puido encontrar o ficheiro.");
```

Calquera das dúas funcións admiten como parámetros ficheiros remotos empregando os protocolos HTTP ou FTP.

```
$fich = file_get_contents("http://www.google.es");
if($fich === false) die("ERRO! Non se puido encontrar a páxina.");
```

Igualmente, para escribir un ficheiro a forma máis sinxela é empregar a función **file_put_contents**, que pasa o contido dunha cadea a un ficheiro.

```
$fich = file_put_contents("proba.txt", "Contido a escribir no ficheiro");
if($fich === false) die("ERRO! Non se puido crear o ficheiro.");
```

Outro xeito de ler e escribir o contido dun ficheiro é empregar as funcións herdadas da linguaxe C, entre outras: **fopen**, **fread**, **fwrite**, **fgets**, **fputs**, **fseek**, **feof** e **fclose**. Por exemplo:

```
$fich = fopen("proba.txt", "r");
if($fich === false) die("ERRO! Non se puido ler o ficheiro.");

while(!feof($fich)) {
    $line = fgets($fich);
    echo $line;
}
```

Directorios

A extensión **Directorios** é semellante á anterior, pero centrándose no traballo cos directorios e o seu contido. Inclúe unha serie de funcións que poden funcionar de dous xeitos:

- Empregando variables de tipo *resource* para facer referencia a un directorio. Cando se abre un directorio empregando a función **opendir**, obtense unha variable que se pode empregar posteriormente nos accesos a ese mesmo directorio. Deste xeito podemos ter varios directorios abertos ao mesmo tempo, cadanseu coa sus propia variable.

Se non precisamos traballar con varios directorios de forma simultánea, podemos obviar a variable devolta por *opendir*. As posteriores chamadas as outras funcións como **readdir** ou **closedir** actuarán sobre o recurso aberto na última chamada a *opendir*.

```
if (opendir('.') !== false) {
    while (($item = readdir()) !== false) {
        echo $item . '<br />';
    }
    closedir();
}
```

- Abrendo o directorio coa función **dir** e empregando posteriormente as propiedades e os métodos do obxecto da **clase Directory** devolto.

```
if (($d = dir('.')) !== false) {
    while (($item = $d->read()) !== false) {
        echo $item . '<br />';
    }
    $d->close();
}
```

Iteradores do sistema de ficheiros

A partir da versión 5 de PHP inclúese coa distribución a **biblioteca estándar (SPL)**, que está formada por un conxunto de clases e interfaces de uso común. Entre eles temos a clase iterador **DirectoryIterator**, que permite percorrer de forma sinxela os obxectos do sistema de ficheiros. Por exemplo, para listar os ficheiros e directorios na ruta na que se atopa o script, poderías facer:

```
foreach (new DirectoryIterator('.') as $fich) {
    if($fich->isDot()) continue;
    echo $fich->getFilename() . "<br>\n";
}
```

A partires de PHP5.3 inclúese tamén o iterador **FilesystemIterator**, que herda de **DirectoryIterator** extendendo a súa funcionalidade.

--Víctor Lourido 16:08 15 jul 2013 (CEST)