

Funciones con directorios

PHP dispone de funciones que permiten obtener información sobre los contenidos de los directorios del servidor.

Algunas de estas funciones son las siguientes:

Pseudo-objeto `dir`

Mediante la expresión:

`$b = dir (path)`

en la que **path** es la *ruta absoluta*:
(../dir/subdir/subsubdir)
o *relativa*
(../subdir)
hasta el directorio del que vamos a obtener información, recogemos en la variable **`$b`** información sobre el directorio en cuestión de una forma un tanto peculiar.

`$b` se comporta como **objeto** y, como tal *objeto*, posee los **métodos** y **funciones** que indicamos:

handle

Devuelve una cadena con la descripción del identificador del recurso.

path

Devuelve la **ruta** del directorio especificado.

read()

Realiza una lectura secuencial de los nombres y extensiones de los ficheros contenidos en el directorio especificado.

rewind()

Posiciona el puntero en la posición inicial.

close()

Cierra el *identificador* de directorio.

Otras funciones

`$f = opendir (path)`

Recoge en la cadena **`$f`** un identificador que permitirá utilizar las restantes funciones.

`$z = readdir ($f)`

Hace una lectura **secuencial** del

El pseudo-objeto `$b=dir("path")`

Este es un ejemplo de utilización de los métodos de éste pseudo-objeto. Observa que tanto el método **handle** como el método **path** se invocan sin utilizar **()** mientras que tanto **read()** como **rewind()** como **close()** sí utilizan esos paréntesis al *invocar* sus métodos.

```
<?
$directorio = dir("../images");
# en el caso de los objetos la manera de invocar
# uno de sus métodos y/o funciones requiere una sintaxis
# específica con la que vera a lo largo de este ejemplo:
# es $objeto->metodo que equivale a la tradicional llamada
# a una variable en la forma $variable.
echo "Handle: ".$directorio->handle."<br>\n";
echo "Path: ".$directorio->path."<br>\n";
while($fichero=$directorio->read()) {
    echo $fichero."<br>\n";
}
$directorio->rewind();
echo "nuevo listado del directorio despues de rebobinar<br>" ;
while($fichero=$directorio->read()) {
    echo $fichero."<br>";
}
$directorio->close();
?>
```

ejemplo106.php

Otras funciones de directorios

Existen otras funciones -indicadas al margen- que permiten obtener la misma información que la que obtuvimos en el ejemplo anterior.

Antes de efectuar la lectura de un directorio es necesario *abrirlo* con la función **opendir**, y una vez finalizada, es aconsejable *cerrarlo* utilizando la función **closedir**.

```
<?

#abrimos el identificador de directorio

$f = opendir("../images");

#leemos el primer fichero que será "." (recuerda la estructura
#                                     de los directorios de MS-DOS)

    $fichero=readdir($f);

    echo $fichero."<br>";

#leemos el fichero siguiente que será ".." (recuerda la estructura
#                                     de los directorios de MS-DOS)

    $fichero=readdir($f);

    echo $fichero."<br>";

#leemos el fichero siguiente (el primer fichero "real")

    $fichero=readdir($f);

    echo $fichero."<br>";

#rebobinamos, enviando el puntero al primer fichero

rewinddir($f);
```

directorio indicado por el identificador **\$f1**.

A medida que efectúa la lectura secuencial el *puntero de lectura* va desplazándose al fichero que sigue al último leído.

rewinddir(\$f1)

Rebobina haciendo que el puntero apunte al **primer fichero** del directorio.

closedir(\$f1)

Cierra el identificador del directorio.

```
echo "Lista de TODOS los ficheros usando un bucle while<br>";

#leemos todos los ficheros

while($fichero=readdir($f)) {
    echo $fichero."<br>";
}
closedir($f);

?>
```

ejemplo107.php

Anterior



Índice



Siguiente

