

PHP

CREACION Y MANEJO DE DIRECTORIOS

DIRECTORIOS

- Php permite crear tanto archivos, como carpetas en las cuales guardar los archivos.
- El procesamiento sería similar al de ARCHIVOS:
 - Se abre directorio.
 - Después se procesa cada entrada del mismo.
 - Se cierra el directorio.

Abrir directorios

- Al igual que ocurre con los archivos, hay una serie de funciones se permiten dejar preparado el directorio, para poder procesar los ficheros que están en el. Esto lo hacemos de la siguiente manera:
 - `$manejador=opendir("ruta/nombre carpeta");`
- Con este paso hemos abierto el directorio, una vez abierto aplicamos las funciones necesarias para realizar los cambios, y después lo cerramos mediante:
 - `closedir($manejador);` Si no se especifica el descriptor se asumirá la última conexión abierta por *opendir()*.
- Tal como ocurría con los archivos, se crea aquí también un puntero interno que permite desplazarse por los distintos elementos del directorio. A continuación veremos las funciones que pueden utilizarse con el directorio abierto.

Directorios : Manejo de directorios

- **Funciones:**

- `is_dir("ruta/nombre_carpeta");`
 - Comprueba si la carpeta especificada en el argumento existe, devuelve el booleano `true` si ésta existe o `false` si no existe.
- `mkdir("ruta/nombre_carpeta");`
 - Crea una nueva carpeta en la ruta y con el nombre especificado. Si la carpeta **ya existe no hace nada (devuelve false)**, o puede dar un mensaje de error.
- `mkdir("ruta/nombre_carpeta");`
 - Borra la carpeta indicada en la ruta siempre que esta **exista y esté vacía**. En caso contrario devuelve `false` y da un mensaje de error.

Directorios : Manejo de directorios

–Lectura

- readdir(\$descriptor) Booleano:
 - Devuelve el nombre **del siguiente fichero** en el directorio.
 - Los nombres de archivo **son devueltos en el orden en que están** en el sistema de archivos.
 - Si no se especifica el descriptor , se usa la última conexión abierta por *opendir()*.

Funciones de directorios

- chdir:

chdir (string \$directory)

Cambia el directorio actual al directorio *\$directory*.

- getcwd string getcwd (void):

Obtiene el directorio actual.

string getcwd (void)

EJEMPLOS

```
<?PHP
```

```
echo getcwd() . "\n"; // me da el nombre del directorio actual
```

```
chdir('dwes20'); // cambio al directorio dwes20
```

```
print '<br>';
```

```
echo getcwd() . "\n"; // me indica que estoy en dwes20
```

```
C:\wamp\www
```

```
C:\wamp\www\dwes20
```

Funciones de directorios

- Basename: La función `basename ()` devuelve el nombre de archivo de una ruta.

```
string basename (string $path [, string $suffix ])
```

`$path`: Especifica una ruta de archivo.

`$suffix`: Es una extensión de archivo. Si el nombre de archivo tiene esta extensión de archivo, la extensión de archivo se cortará.

```
$directorio = "este/es/un/directorio/cualquiera.csv";  
echo basename($directorio, '.csv'); // Devuelve cualquiera
```


Ejemplo básico:

Mostrar los ficheros de un directorio con su tamaño

```
<head>
<meta charset='utf-8'/>
</head>
<?php
$directorio = opendir("."); //Abro el directorio de actual
print"El directorio actual es:"; echo getcwd() ;print" <h1> Su contenido es </h1>";
while($archivo = readdir($directorio)) //obtenemos un archivo tras otro con
lecturas sucesivas
    {
        if(is_dir($archivo)) //verificamos si es o no un directorio
        {
            echo "[".$archivo . "<br/>"; } //lo envolvemos entre corchetes
        else{ //si es un fichero visualizo nombre y tamaño

            echo $archivo . "-con un tamaño de :".filesize($archivo) ."bytes

<br/>";}
    }
Closedir($directorio);
?>
```

Resumen de Funciones

1. `opendir("ruta/nombre carpeta")` → Abre un **directorio**
2. `closedir($manejador)` → Cierra un gestor de directorio abierto
3. `chdir(directorio)` → Cambia de directorio.
4. `getcwd(directorio)` → el directorio actual.
5. `mkdir(directorio)` → Crea un directorio.
6. `rmdir(directorio)` → Elimina un directorio.
7. `ls_dir(directorio)` → Verifica si el nombre de archivo especificado es un directorio.
8. `basename` → Devuelve la parte del path correspondiente al nombre del archivo.
9. `rewinddir(directorio)` → Rebobina el puntero interno de un directorio para que apunte nuevamente al comienzo del mismo, si no se especifica el gestor, se asumirá la última conexión abierta por *opendir()*.