

Programación de headers con PHP

El siguiente script presenta un listado de los elementos de la matriz asociativa `$_SERVER`. Como verás hay elementos que recogen los valores de las cabecera de solicitud HTTP (todas aquellas que empiezan por `HTTP_`) y otro tipo de información como rutas y ubicaciones de script.

```
<?php
/**
 * Lista los diferentes valores de los elementos de la matriz del sistema $_SERVER,
 * entre
 * ellos están los elementos que guardan los valores de las solicitudes y respuestas
 * HTTP.
 */
$server_host=$_SERVER['HTTP_HOST'];
echo "<p><strong>Listado de los valores de los elementos de la matriz \$_SERVER
obtenidos al acceder a $server_host</strong></p>";
//print_r($_SERVER);

foreach($_SERVER as $key => $value) {
    echo "$key = $value.<br>";
}
echo "<br>";

/* Un resultado parecido pero usando la función getallheaders
*/
foreach (getallheaders() as $key => $value) {
    echo "$key = $value.<br>";
}

?>
```

Cabeceras de solicitud:

1. Una página web `idioma.php` con un texto corto, que con un script php seleccione el idioma de presentación de la página a partir de los valores recogidos a las cabeceras. La página la mostrará a ser posible en ese idioma y sino en el que tú programes por defecto.
2. Crea una página web de contenido sencillo llamada `navegador.php`, que con un script php intercepte la presentación de la página web. Si el navegador del cliente es Firefox mostrará la página y en cualquiera otro caso, no mostrará la página sino un mensaje de aviso. Podéis probarlo con el comando `wget` de linux.

Cabeceras de respuesta:

1. Crea una página llamada `productos.php` que reciba un parámetro llamado **id** para indicar el producto a mostrar. En el script, crea un array para almacenar productos. Si el producto pedido no existe en el array debe mostrar un mensaje advirtiéndolo y enviar la cabecera de respuesta **404**. Usa la función `http_response_code`.

```
$productos = ["1" => "Producto 1", "2" => "Producto 2", "3" => "Producto 3"];
```

Para saber si una clave está en un array puedes usar la función `array_key_exists`.

No confundir con la directiva `ErrorDocument` de apache.

- Supón que quieres sacar un listado de los productos que tienes en una base de datos en formato `csv` para posteriormente tratarlos con una hoja de cálculo. Este formato es muy sencillo: cada fila se representa en una única línea y en cada línea cada dato se puede separar, por ejemplo, con un `;` Cuando visitas la página, el navegador te debe dar la opción de abrirlo con la hoja de cálculo con el nombre `productos.csv` (no se debe mostrar como texto plano)

Haz un script llamado `csv.php` que a partir del array de productos, genere el csv apropiado.

Usa la función [header](#) aplicando las cabeceras apropiadas

- Es muy frecuente en aplicaciones web el uso de **redirecciones**. Supón que un usuario visita un página en la que debe estar registrado para poder acceder. La aplicación debe **redirigir** al usuario a la página `login.php` cuando visite dicha página. Esto se consigue mediante la función `header('Location: login.php');` Crea un script llamado `privada.php`. Esta página recibe (o no) un parámetro, llamado `dejameEntrar` con el valor `1` o `0`. Si es `1`, debemos mostrar la página con un mensaje diciendo "Bienvenido"; si es `0`, le hemos de redigir a una página llamada `login.php` en la que simplemente mostraremos el mensaje "Debes hacer login primero";

Cuidado. Antes de llamar a `header('Location: ');` no se debe haber enviado ningún código al navegador. Es decir no puede haber código HTML escrito o generado mediante `echo` o similares.

Cuidado II. Siempre que llames a `header('Location: ');` la siguiente instrucción debe ser `exit();` para acabar la ejecución del script.

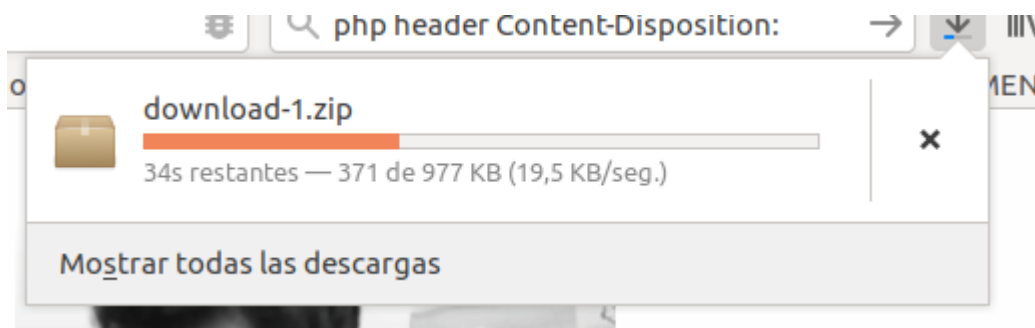
- En este ejemplo `download.php`, se muestra el uso de la cabecera `Content-length` para indicar al cliente el tamaño del archivo.

```
<?php
// it's a zip file
header('Content-Type: application/zip');
// 1 million bytes (about 1megabyte)
header('Content-Length: 1000000');
// load a download dialogue, and save it as download.zip
header('Content-Disposition: attachment; filename="download.zip"');

// 1000 times 1000 bytes of data
for ($i = 0; $i < 1000; $i++) {
    echo str_repeat(".",1000);

    // sleep to slow down the download
    usleep(50000);
}
?>
```

Si llamas a este script, se mostrará el diálogo de descarga y podrás ver el progreso completo.

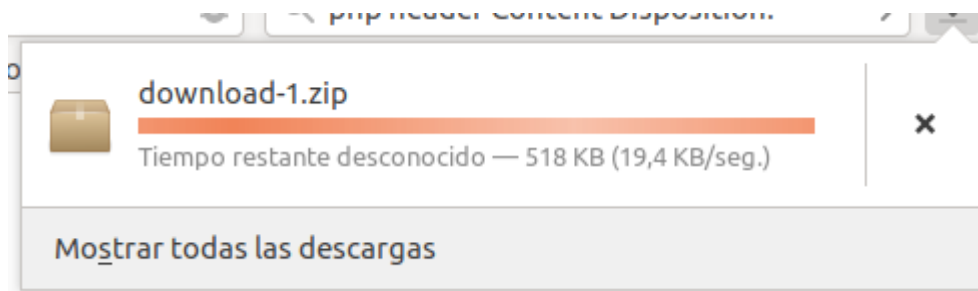


Ahora fíjate en qué ocurre cuando **comentamos** la línea `header('Content-Length: 1000000');`

```
<?php
// it's a zip file
header('Content-Type: application/zip');
// 1 million bytes (about 1megabyte)
// header('Content-Length: 1000000');
// load a download dialogue, and save it as download.zip
header('Content-Disposition: attachment; filename="download.zip"');

// 1000 times 1000 bytes of data
for ($i = 0; $i < 1000; $i++) {
    echo str_repeat(".",1000);

    // sleep to slow down the download
    usleep(50000);
}
?>
```



Ya no muestra correctamente el tiempo de descarga porque desconoce cuánto ocupa!!