

Codificación UTF-8 con PHP

utf-8

Esta es un mini post muy rápido para comentar algo muy importante, que todos sabemos pero que no está de más recordar. Cuando escribimos para la web a menudo surgen problemas con los caracteres acentuados y otros propios de codificaciones locales, como el español. Se visualizan incorrectamente en el navegador.

Tradicionalmente esto se venía solucionando sustituyendo, en HTML, esos caracteres por sus correspondientes entidades. Así, si queremos que en el texto de la página, por ejemplo, aparezca la palabra página, en el HTML la escribimos cómo p´gina. Esto, además de ser una solución obsoleta, es tremendamente engorrosa en textos largos, y siempre se nos "pasa" algún carácter. Cuando trabajamos en PHP, la solución pasa por "filtrar" una palabra o frase con la función `htmlspecialchars()`. Bueno. No deja de ser un apañío. Funciona (en general), pero no es lo ideal.

Lo suyo es que podamos tomar el texto libremente, tanto si procede de PHP cómo si está incrustado, directamente, en HTML. Para lograr esto, hay que tener presentes, principalmente, dos cosas:

El editor de texto o el IDE que empleemos para escribir el código de nuestra página debe guardarlo codificado en UTF-8. Esto es algo a lo que muchos desarrolladores no le dan importancia, pero no tener en cuenta esto desde el principio causa muchos problemas. Cada IDE emplea un modo propio para establecer la codificación de los archivos. Revisa el manual del que tu uses para saber cómo hacer que se graben en UTF-8.

Además de lo anterior, cada script PHP debería incorporar, al principio del código, las siguientes líneas:

```
header('Content-Type: text/html; charset=UTF-8');  
mb_internal_encoding("UTF-8");
```

Si tu página incluye código HTML, dentro de la cabecera (sección `<head>`) deberás incluir, además, la siguiente línea:

```
<meta charset="UTF-8">
```

Si, además, tienes acceso a la configuración de Apache (por ejemplo, en tu localhost o si cuentas con un servidor dedicado), añade, al final del archivo `.htaccess`, la siguiente línea:

```
AddDefaultCharset utf-8
```

Otro punto que hay que tener en cuenta es el uso de bases de datos. Si tu script PHP lee los datos de una base de datos SQL (normalmente se emplea MySQL), tanto la base de datos cómo las tablas y los campos de estas deberán estar correctamente codificados. Para crear correctamente una base de datos emplea:

```
CREATE DATABASE mi_base_de_datos DEFAULT CHARACTER SET 'utf8' DEFAULT COLLATE 'utf8_general_ci';
```

Esto hará que, por defecto, las tablas se creen con la codificación correcta. De todos modos, para crearlas puedes emplear:

```
CREATE TABLE miTabla (  
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(20) CHARACTER SET utf8 COLLATE utf8_general_ci DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARACTER SET=utf8;
```

De este modo establecemos la codificación correcta para la tabla y los campos que nos interesen. Si empleas PHPMyAdmin (una herramienta que hoy día empleamos todos) puedes establecer la codificación mediante las opciones que te ofrece. Visita [este enlace](#). Si tienes en cuenta estos consejos, no volverás a tener problemas de visualización de caracteres especiales en tu página.