

# CADENAS DE CARACTERES MULTIBYTE

## Introducción ¶

Aunque existen muchos idiomas en los cuales cada carácter necesario puede ser representado por una referencia uno a uno a un valor de 8 bits, existen también bastantes idiomas que requieren tantos caracteres para la comunicación escrita que no pueden ser representados dentro del rango que un mero byte puede codificar (un byte se compone de ocho bits. Cada bit puede contener solamente dos valores distintos, uno o cero. Debido a esto, con un byte solo se pueden representar 256 valores únicos ( $2$  a la  $8^{\text{a}}$  potencia)). Los esquemas de codificación multibyte fueron desarrollados precisamente para expresar más de 256 caracteres en el sistema de codificación regular a nivel de bits.

Cuando se manipulan cadenas de caracteres (trim, split, splice, etc.) en una codificación multibyte, es necesario utilizar funciones especiales, ya que dos o más bytes consecutivos pueden representar un único carácter en tal esquema de codificación. Si, de lo contrario, se usa una función que no considera caracteres multibyte con la cadena de caracteres, es probable que falle al detectar el comienzo o el final del carácter multibyte, y que se termine con una cadena de caracteres corrupta que probablemente pierda su significado original.

`mbstring` proporciona funciones específicas para cadenas de texto multibyte que ayudan a tratar codificaciones multibyte en PHP. Además, `mbstring` controla la conversión de la codificación de caracteres entre los posibles esquemas de codificación. `mbstring` está diseñada para manejar codificaciones basadas en Unicode, tales como UTF-8 y UCS-2, y, por conveniencia, varias codificaciones de un solo byte (enumeradas más adelante).

## Codificaciones de caracteres admitidas ¶

Actualmente, el módulo `mbstring` admite entre otras las siguientes codificaciones de caracteres. Cualquiera de estas codificaciones de caracteres se puede especificar en el parámetro `encoding` de las funciones de `mbstring`.

UTF-32\*

UTF-32BE\*

UTF-32LE\*

UTF-16\*

UTF-16BE\*

UTF-16LE\*

UTF-7

UTF7-IMAP

UTF-8\*

ASCII\*

ISO-2022-JP

ISO-2022-JP-MS

ISO-8859-1\*

ISO-8859-2\*

ISO-8859-3\*

ISO-8859-4\*

ISO-8859-5\*

ISO-8859-6\*

ISO-8859-7\*

ISO-8859-8\*

ISO-8859-9\*

ISO-8859-10\*

ISO-8859-13\*

ISO-8859-14\*

ISO-8859-15\*

ISO-8859-16\*

BASE64

HTML-ENTITIES

Windows-1251 (CP1251)

Windows-1252 (CP1252)

CP866 (IBM866)

\* denota codificaciones que también pueden ser utilizadas en expresiones regulares.

\*\* denota codificaciones que están disponibles desde PHP 5.4.0.

## Algunas funciones

- [mb\\_strlen](#) — Obtiene la longitud de una cadena de caracteres
- [mb\\_strpos](#) — Busca la posición de la primera ocurrencia de un string en otro string
- [mb\\_strrchr](#) — Busca la última ocurrencia de un carácter de un string dentro de otro
- [mb\\_strrichr](#) — Busca la última ocurrencia de un carácter de un string dentro de otro, insensible a mayúsculas/minúsculas
- [mb\\_strripos](#) — Busca la posición de la última ocurrencia de un string dentro de otro string, insensible a mayúsculas/minúsculas
- [mb\\_strrpos](#) — Busca la posición de la última ocurrencia de un string en otro string
- [mb\\_strstr](#) — Busca la primera ocurrencia de un string dentro de otro
- [mb\\_strtolower](#) — Convierte una cadena de caracteres a minúsculas
- [mb\\_strtoupper](#) — Convierte un string en mayúsculas
- [mb\\_substr\\_count](#) — Cuenta el número de ocurrencias de un substring
- [mb\\_substr](#) — Obtiene parte de una cadena de caracteres
- [mb\\_split](#) — Divide cadenas de caracteres multibyte usando una expresión regular
- [mb\\_str\\_split](#) — Given a multibyte string, return an array of its characters
- [mb\\_strcut](#) — Obtener parte de un string
- [mb\\_http\\_input](#) — Detecta la codificación de caracteres de entrada HTTP
- [mb\\_http\\_output](#) — Establece/obtiene la codificación de caracteres de salida HTTP
- [mb\\_internal\\_encoding](#) — Establece/obtiene la codificación de caracteres interna