mb-ereg-replace-callback

mb_ereg_replace_callback

Perform a regular expression search and replace with multibyte support using a callback

ENTITY

string

false

null

mb_ereg_replace_callback

string

pattern

callable

callback

string

string

string

null

options

ENTITY

Scans `string` for matches to `pattern`, then replaces the matched text with the output of `callback` function.

The behavior of this function is almost identical to `mb_ereg_replace`, except for the fact that instead of `replacement` parameter, one should specify a `callback`.

ENTITY

`pattern`
The regular expression pattern.

Multibyte characters may be used in `pattern`.

`callback`
A callback that will be called and passed an array of matched elements in the `string` string. The callback should return the replacement string.

You'll often need the `callback` function for a `mb_ereg_replace_callback` in just one place. In this case you can use an anonymous function to declare the callback within the call to `mb_ereg_replace_callback`. By doing it this way you have all information for the call in one place and do not clutter the function namespace with a callback function's name not used anywhere else.

`string`
The `string` being checked.

`options`
The search option. See `mb_regex_set_options` for explanation.

ENTITY

The resultant `string` on success, or ENTITY on error. If `string` is not valid for the current encoding, ENTITY is returned.

ENTITY

ENTITY	ENTITY
8.0.0	`options` is nullable now.
7.1.0	The function checks whether `string` is valid for the current encoding.

ENTITY

`mb_ereg_replace_callback` example

```
<?php
// this text was used in 2002
```

```
// we want to get this up to date for 2003
$text = "April fools day is 04/01/2002\n";
$text.= "Last christmas was 12/24/2001\n";
// the callback function
function next_year($matches)
{
  // as usual: $matches[0] is the complete match
  // $matches[1] the match for the first subpattern
  // enclosed in '(...)' and so on
  return $matches[1].($matches[2]+1);
}
echo mb_ereg_replace_callback(
            "(\d{2}/\d{2}/)(\d{4})",
            "next_year",
            $text);

?>
```

ENTITY

```
April fools day is 04/01/2003
Last christmas was 12/24/2002
```

`mb_ereg_replace_callback` using anonymous function

```
<?php
// this text was used in 2002
// we want to get this up to date for 2003
$text = "April fools day is 04/01/2002\n";
$text.= "Last christmas was 12/24/2001\n";

echo mb_ereg_replace_callback(
            "(\d{2}/\d{2}/)(\d{4})",
            function ($matches) {
               return $matches[1].($matches[2]+1);
            },
            $text);
?>
```

ENTITY ENTITY

ENTITY

`mb_regex_encoding`, `mb_ereg_replace`, Anonymous functions

□□□ #2
□ □□□ □□□ 2026-10-31 17:52:50 UTC
□ □□□ □□□ 2026-10-31 17:52:52 UTC