

[V8js »](#)
[« urldecode](#)

- [PHP Manual](#)
- [Function Reference](#)
- [Other Basic Extensions](#)
- [URLs](#)
- [URL Functions](#)

Change language: English ▼

[Submit a Pull Request](#) [Report a Bug](#)

urlencode

(PHP 4, PHP 5, PHP 7)

urlencode — URL-encodes string

Description ¶

urlencode (string \$string) : string

This function is convenient when encoding a string to be used in a query part of a URL, as a convenient way to pass variables to the next page.

Parameters ¶

string

The string to be encoded.

Return Values ¶

Returns a string in which all non-alphanumeric characters except `-_.` have been replaced with a percent (%) sign followed by two hex digits and spaces encoded as plus (+) signs. It is encoded the same way that the posted data from a WWW form is encoded, that is the same way as in `application/x-www-form-urlencoded` media type. This differs from the » [RFC 3986](#) encoding (see [rawurlencode\(\)](#)) in that for historical reasons, spaces are encoded as plus (+) signs.

Examples ¶

Example #1 urlencode() example

```
<?php
echo '<a href="mycgi?foo=' . urlencode($userinput) . '>';
?>
```

Example #2 urlencode() and [htmlentities\(\)](#) example

```
<?php
$query_string = 'foo=' . urlencode($foo) . '&bar=' . urlencode($bar);
echo '<a href="mycgi?' . htmlentities($query_string) . '>';
?>
```

Notes ¶

Note:

Be careful about variables that may match HTML entities. Things like ©, © and £ are parsed by the browser and the actual entity is used instead of the desired variable name. This is an obvious hassle that the W3C has been telling people about for years. The reference is here: [» http://www.w3.org/TR/html4/appendix/notes.html#h-B.2.2](http://www.w3.org/TR/html4/appendix/notes.html#h-B.2.2).

PHP supports changing the argument separator to the W3C-suggested semi-colon through the `arg_separator` .ini directive. Unfortunately most user agents do not send form data in this semi-colon separated format. A more portable way around this is to use & instead of & as the separator. You don't need to change PHP's `arg_separator` for this. Leave it as &, but simply encode your URLs using [htmlentities\(\)](#) or [htmlspecialchars\(\)](#).

See Also

- [urlencode\(\)](#) - Decodes URL-encoded string
- [htmlentities\(\)](#) - Convert all applicable characters to HTML entities
- [rawurlencode\(\)](#) - URL-encode according to RFC 3986
- [rawurldecode\(\)](#) - Decode URL-encoded strings
- [» RFC 3986](#)

[+ add a note](#)

User Contributed Notes 24 notes

[up](#)
[down](#)
 60

[davis dot peixoto at gmail dot com](#)

10 years ago

`urlencode` function and `rawurlencode` are mostly based on RFC 1738.

However, since 2005 the current RFC in use for URIs standard is RFC 3986.

Here is a function to encode URLs according to RFC 3986.

```
<?php
function myUrlEncode($string) {
    $entities = array('%21', '%2A', '%27', '%28', '%29', '%3B', '%3A', '%40', '%26', '%3D', '%2B',
'%24', '%2C', '%2F', '%3F', '%25', '%23', '%5B', '%5D');
    $replacements = array('!', '*', '"', '(', ')', ';', ':', '@', '&', '=', '+', '$', ',', '/',
'?', '%', '#', '[', ']');
    return str_replace($entities, $replacements, urlencode($string));
}
?>
```

[up](#)
[down](#)
 1

[no_gravity](#)

1 year ago

I think the description does not exactly match what the function does:

Returns a string in which all non-alphanumeric characters except `-_.` have been replaced with a percent (%) sign followed by two hex digits and spaces encoded as plus (+) signs.

`urlencode('ö')` gives me `'%C3%B6'`. So more than just a percent sign followed by two hex digits.

[up](#)
[down](#)

7

[daniel+php at danielnorton dot com ¶](#)

11 years ago

Don't use urlencode() or urldecode() if the text includes an email address, as it destroys the "+" character, a perfectly valid email address character.

Unless you're certain that you won't be encoding email addresses AND you need the readability provided by the non-standard "+" usage, instead always use rawurlencode() or rawurldecode().

[up](#)
[down](#)

6

[temu92 at gmail dot com ¶](#)

11 years ago

I needed encoding and decoding for UTF8 urls, I came up with these very simple fuctions. Hope this helps!

```
<?php
function url_encode($string){
    return urlencode(utf8_encode($string));
}

function url_decode($string){
    return utf8_decode(urldecode($string));
}
?>
```

[up](#)
[down](#)

1

[admin at server dot net ¶](#)

1 year ago

urlencode corresponds to the definition for application/x-www-form-urlencoded in RFC 1866 (<https://tools.ietf.org/html/rfc1866#section-8.2.1>), and not for url encoded parts in URI. Use only rawurlencode for encode raw URI parts (e.g. query/search part)!

[up](#)
[down](#)

6

[omid at omidsakhi dot com ¶](#)

10 years ago

I needed a function in PHP to do the same job as the complete escape function in Javascript. It took me some time not to find it. But findaly I decided to write my own code. So just to save time:

```
<?php
function fullescape($in)
{
    $out = '';
    for ($i=0;$i<strlen($in);$i++)
    {
        $hex = dehex(ord($in[$i]));
        if ($hex=='')
            $out = $out.urlencode($in[$i]);
        else
            $out = $out .'%.((strlen($hex)==1) ? ('0'.strtoupper($hex)):(strtoupper($hex)));
    }
    $out = str_replace('+','%20',$out);
    $out = str_replace('_','%5F',$out);
}
```

```

$out = str_replace('.', '%2E', $out);
$out = str_replace('-', '%2D', $out);
return $out;
}
?>

```

It can be fully decoded using the `unescape` function in Javascript.

[up](#)
[down](#)

2

[lekiagospel@gmail dot com ¶](#)

2 months ago

urlencode is useful when using certain URL shortener services.

The returned URL from the shortener may be truncated if not encoded. Ensure the URL is encoded before passing it to a shortener.

Example

```

$url = "https://www.notarealurl.com?id=50&name=namestring";
$encodedurl = urlencode($url);
$shorturl = UrlShortener::shortenUrl( $encodedurl);

```

[up](#)
[down](#)

2

[kL ¶](#)

14 years ago

Apache's `mod_rewrite` and `mod_proxy` are unable to handle urlencoded URLs properly -

http://issues.apache.org/bugzilla/show_bug.cgi?id=34602

If you need to use any of these modules and handle paths that contain `%2F` or `%3A` (and few other encoded special url characters), you'll have use a different encoding scheme.

My solution is to replace `"%"` with `""`.

```

<?php
function urlencode($u)
{
    return str_replace(array("", '%'), array('%27', ""), urlencode($u));
}

```

```

function urldecode($u)
{
    return urldecode(strtr($u, "", '%'));
}
?>

```

[up](#)
[down](#)

1

[david winiecki gmail ¶](#)

6 years ago

Since PHP 5.3.0, `urlencode` and `rawurlencode` also differ in that `rawurlencode` does not encode `~` (tilde), while `urlencode` does.

[up](#)
[down](#)

1

[izhankhalib at gmail dot com ¶](#)

7 years ago