

- [PHP Manual](#)
- [Function Reference](#)
- [Variable and Type Related Extensions](#)
- [Filter](#)
- [Types of filters](#)

Change language: English ▼

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

Sanitize filters ¶

List of filters for sanitization				
ID	Name	Flags	Description	
<code>FILTER_SANITIZE_EMAIL</code>	"email"		Remove all characters except letters, digits and <code>!#\$%&'*+-=?^_`{ }~@.[]</code> .	
<code>FILTER_SANITIZE_ENCODED</code>	"encoded"	<code>FILTER_FLAG_STRIP_LOW</code> , <code>FILTER_FLAG_STRIP_HIGH</code> , <code>FILTER_FLAG_STRIP_BACKTICK</code> , <code>FILTER_FLAG_ENCODE_LOW</code> , <code>FILTER_FLAG_ENCODE_HIGH</code>	URL-encode string, optionally strip or encode special characters.	
<code>FILTER_SANITIZE_MAGIC_QUOTES</code>	"magic_quotes"		Apply addslashes() . (<i>DEPRECATED</i> as of PHP 7.3.0 and <i>REMOVED</i> as of PHP 8.0.0, use <code>FILTER_SANITIZE_ADD_SLASHES</code> instead.)	
<code>FILTER_SANITIZE_ADD_SLASHES</code>	"add_slashes"		Apply addslashes() . (Available as of PHP 7.3.0)	
<code>FILTER_SANITIZE_NUMBER_FLOAT</code>	"number_float"	<code>FILTER_FLAG_ALLOW_FRACTION</code> , <code>FILTER_FLAG_ALLOW_THOUSAND</code> , <code>FILTER_FLAG_ALLOW_SCIENTIFIC</code>	Remove all characters except digits, <code>+-</code> and optionally <code>.,eE</code> .	
<code>FILTER_SANITIZE_NUMBER_INT</code>	"number_int"		Remove all characters except digits, plus and minus sign.	
<code>FILTER_SANITIZE_SPECIAL_CHARS</code>	"special_chars"	<code>FILTER_FLAG_STRIP_LOW</code> , <code>FILTER_FLAG_STRIP_HIGH</code> , <code>FILTER_FLAG_STRIP_BACKTICK</code> , <code>FILTER_FLAG_ENCODE_HIGH</code>	HTML-encode <code>'"<>&</code> and characters with ASCII value less than 32, optionally strip or encode other special characters.	
<code>FILTER_SANITIZE_FULL_SPECIAL_CHARS</code>	"full_special_chars"	<code>FILTER_FLAG_NO_ENCODE_QUOTES</code> ,	Equivalent to calling htmlspecialchars() with <code>ENT_QUOTES</code> set. Encoding quotes can be disabled by setting <code>FILTER_FLAG_NO_ENCODE_QUOTES</code> . Like htmlspecialchars() , this filter is aware of the default_charset and if a sequence	

ID	Name	Flags	Description
<code>FILTER_SANITIZE_STRING</code>	"string"	<code>FILTER_FLAG_NO_ENCODE_QUOTES</code> , <code>FILTER_FLAG_STRIP_LOW</code> , <code>FILTER_FLAG_STRIP_HIGH</code> , <code>FILTER_FLAG_STRIP_BACKTICK</code> , <code>FILTER_FLAG_ENCODE_LOW</code> , <code>FILTER_FLAG_ENCODE_HIGH</code> , <code>FILTER_FLAG_ENCODE_AMP</code>	of bytes is detected that makes up an invalid character in the current character set then the entire string is rejected resulting in a 0-length string. When using this filter as a default filter, see the warning below about setting the default flags to 0.
<code>FILTER_SANITIZE_STRIPPED</code>	"stripped"		Strip tags and HTML-encode double and single quotes, optionally strip or encode special characters. Encoding quotes can be disabled by setting <code>FILTER_FLAG_NO_ENCODE_QUOTES</code> . (<i>Deprecated</i> as of PHP 8.1.0, use htmlspecialchars() instead.)
<code>FILTER_SANITIZE_URL</code>	"url"		Alias of "string" filter. (<i>Deprecated</i> as of PHP 8.1.0, use htmlspecialchars() instead.) Remove all characters except letters, digits and \$- _.+!*'(),{ \\^~[]`<>#%";/?:@&=.
<code>FILTER_UNSAFE_RAW</code>	"unsafe_raw"	<code>FILTER_FLAG_STRIP_LOW</code> , <code>FILTER_FLAG_STRIP_HIGH</code> , <code>FILTER_FLAG_STRIP_BACKTICK</code> , <code>FILTER_FLAG_ENCODE_LOW</code> , <code>FILTER_FLAG_ENCODE_HIGH</code> , <code>FILTER_FLAG_ENCODE_AMP</code>	Do nothing, optionally strip or encode special characters. This filter is also aliased to <code>FILTER_DEFAULT</code> .

Warning

When using one of these filters as a default filter either through your ini file or through your web server's configuration, the default flags is set to `FILTER_FLAG_NO_ENCODE_QUOTES`. You need to explicitly set `filter.default_flags` to 0 to have quotes encoded by default. Like this:

Example #1 Configuring the default filter to act like htmlspecialchars

```
filter.default = full_special_chars
filter.default_flags = 0
```

Changelog

Version	Description
8.1.0	<code>FILTER_SANITIZE_STRING</code> and <code>FILTER_SANITIZE_STRIPPED</code> have been deprecated.

Version	Description
8.0.0	<code>FILTER_SANITIZE_MAGIC_QUOTES</code> has been removed.
7.3.0	<code>FILTER_SANITIZE_ADD_SLASHES</code> was added as a replacement for <code>FILTER_SANITIZE_MAGIC_QUOTES</code>
7.3.0	<code>FILTER_SANITIZE_MAGIC_QUOTES</code> has been deprecated.

[+ add a note](#)

User Contributed Notes 15 notes

[up](#)

[down](#)

60

[Anonymous ¶](#)

7 years ago

`FILTER_SANITIZE_STRING` doesn't behavior the same as `strip_tags` function. `strip_tags` allows less than symbol inferred from context, `FILTER_SANITIZE_STRING` strips regardless.

```
<?php
$smaller = "not a tag < 5";
echo strip_tags($smaller);    // -> not a tag < 5
echo filter_var ( $smaller, FILTER_SANITIZE_STRING); // -> not a tag
?>
```

[up](#)

[down](#)

58

[googlybash24 at aol dot com ¶](#)

10 years ago

Remember to `trim()` the `$_POST` before your filters are applied:

```
<?php

// We trim the $_POST data before any spaces get encoded to "%20"

// Trim array values using this function "trim_value"
function trim_value(&$value)
{
    $value = trim($value);    // this removes whitespace and related characters from the beginning and end of the string
}
array_filter($_POST, 'trim_value');    // the data in $_POST is trimmed
```

```

$postfilter = // set up the filters to be used with the trimmed post array
    array(
        'user_tasks' => array('filter' => FILTER_SANITIZE_STRING, 'flags' => !FILTER_FLAG_STRIP_LOW), //
        removes tags. formatting code is encoded -- add nl2br() when displaying
        'username' => array('filter' => FILTER_SANITIZE_ENCODED, 'flags' => FILTER_FLAG_STRIP_LOW), // we
        are using this in the url
        'mod_title' => array('filter' => FILTER_SANITIZE_ENCODED, 'flags' => FILTER_FLAG_STRIP_LOW), // we
        are using this in the url
    );

```

```

$revised_post_array = filter_var_array($_POST, $postfilter); // must be referenced via a variable which is now an array that takes the place
of $_POST[]
echo (nl2br($revised_post_array['user_tasks'])); //-- use nl2br() upon output like so, for the ['user_tasks'] array value so that the
newlines are formatted, since this is our HTML <textarea> field and we want to maintain newlines
?>

```

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

31

[Willscrl ¶](#)

6 years ago

To include multiple flags, simply separate the flags with vertical pipe symbols.

For example, if you want to use `filter_var()` to sanitize `$string` with `FILTER_SANITIZE_STRING` and pass in `FILTER_FLAG_STRIP_HIGH` and `FILTER_FLAG_STRIP_LOW`, just call it like this:

```

$string = filter_var($string, FILTER_SANITIZE_STRING, FILTER_FLAG_STRIP_HIGH | FILTER_FLAG_STRIP_LOW);

```

The same goes for passing a flags field in an options array in the case of using callbacks.

```

$var = filter_var($string, FILTER_SANITIZE_SPECIAL_CHARS,
array('flags' => FILTER_FLAG_STRIP_LOW | FILTER_FLAG_ENCODE_HIGH));

```

Thanks to the Brain Goo blog at popmartian.com/tipsntricks/ for this info.

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

9

[ipse at sergiosantos dot me ¶](#)

2 years ago

Although it's specifically mentioned in the above documentation, because many seem to find this unintuitive it's worth pointing out that `FILTER_SANITIZE_NUMBER_FLOAT` will remove the decimal character unless you specify `FILTER_FLAG_ALLOW_FRACTION`:

```
<?php
$number_string = '12.34';

echo filter_var( $number_string, FILTER_SANITIZE_NUMBER_FLOAT ); // 1234

echo filter_var( $number_string, FILTER_SANITIZE_NUMBER_FLOAT, FILTER_FLAG_ALLOW_FRACTION ); // 12.34
?>
```

[up](#)
[down](#)
22

[marcus at synchronedia dot co dot uk ¶](#)

13 years ago

It's not entirely clear what the LOW and HIGH ranges are. LOW is characters below 32, HIGH is those above 127, i.e. outside the ASCII range.

```
<?php
$a = "\tcafé\n";
//This will remove the tab and the line break
echo filter_var($a, FILTER_SANITIZE_STRING, FILTER_FLAG_STRIP_LOW);
//This will remove the é.
echo filter_var($a, FILTER_SANITIZE_STRING, FILTER_FLAG_STRIP_HIGH);
?>
```

[up](#)
[down](#)
13

[AntonioPrimera ¶](#)

6 years ago

Please be aware that when using `filter_var()` with `FILTER_SANITIZE_NUMBER_FLOAT` and `FILTER_SANITIZE_NUMBER_INT` the result will be a string, even if the input value is actually a float or an int.

Use `FILTER_VALIDATE_FLOAT` and `FILTER_VALIDATE_INT`, which will convert the result to the expected type.

[up](#)
[down](#)
9

[galvao at galvao dot eti dot br ¶](#)

9 years ago

Just to clarify, since this may be unknown for a lot of people:

ASCII characters above 127 are known as "Extended" and they represent characters such as greek letters and accented letters in latin alphabets, used in languages such as pt_BR.

A good ASCII quick reference (aside from the already mentioned Wikipedia article) can be found at: <http://www.asciicodes.com/>

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

5

[david dot drakulovski at gmail dot com ¶](#)

8 years ago

Here is a simpler and a better presented ASCII list for the <32 or 127> filters
(if wikipedia confused the hell out of you):

<http://www.danshort.com/ASCIImap/>

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

0

[darren at daz-web dot com ¶](#)

4 years ago

For those looking for a simple way around filtering POST forms that have textarea elements in them. If you also need tab for example you can extend quite easily.

```
<?php
//create an array of all relevant textareas
$textareas = array("ta1");

foreach($_POST as $k => $v)
{
    $v = trim($v);//so we are sure it is whitespace free at both ends

    //preserve newline for textarea answers
    if(in_array($k,$textareas))$v=str_replace("\n","[NEWLINE]",$v);

    //sanitize string
    $v = filter_var($v, FILTER_SANITIZE_STRING, FILTER_FLAG_STRIP_LOW | FILTER_FLAG_STRIP_HIGH | FILTER_FLAG_STRIP_BACKTICK);

    //now replace the placeholder with the original newline
    $_POST[$k] = str_replace("[NEWLINE]","\n",$v);
}
```

```
}

//simple form for testing submital

?><!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Filter test</title>
</head>

<body>

<form action="" method="post">
  <p>
    <textarea name="ta1" cols="30" rows="10"><?php echo $_POST['ta1']; ?></textarea>
  </p>
  <p>
    <input type="text" name="txt1" size="30" value="<?php echo $_POST['txt1']; ?>" />
  </p>
  <p>
    <input type="submit" />
  </p>
</form>

</body>
```

</html>

[up](#)

[down](#)

-5

[adellemfrank at hotmail dot com](#) ¶

10 years ago

A good list of which ASCII characters are < 32 and > 127 can be found at: http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters

[up](#)

[down](#)

-5

[Rodrigo Guariento](#) ¶

2 years ago

To get ONLY numbers from a string use this code:

```
echo preg_replace('/[^0-9]/', '', '123456-789');
```

[up](#)

[down](#)

-4

[Anonymous ¶](#)

2 years ago

```
<?php
```

```
/*
```

```
    filter all ascii and save juste 0-9 a-Z and @ . _
```

```
*/
```

```
echo alphanum('abcdefghABCDEFGH0123456789/!:@._');
```

```
// return abcdefghABCDEFGH0123456789@._
```

```
function alphanum( $string , $x='') {
```

```
    $h=strlen($string);
```

```
    for($a=0; $a<$h; $a++) {
```

```
        $i = ord($string[$a]);
```

```
        if(
```

```
            ($i==46) || // .
```

```
            ($i==64) || // @
```

```
            ($i==95) || // _
```

```
            ($i > 47 && $i < 58) || //0123456789
```

```
            ($i > 64 && $i < 91) || //ABCDEFGH..Z
```

```
            ($i > 96 && $i < 123) //abcdefgh..z
```

```
        ) { $x .= $string[$a]; }
```

```
    }
```

```
    return $x;
```

```
}
```

```
?>
```

[up](#)

[down](#)

-19

[Anonymous ¶](#)

9 years ago

Support for FILTER_SANITIZE_FULL_SPECIAL_CHARS was added from version 5.3.3

[up](#)

[down](#)

-7

[anonymous](#)**2 years ago**

In the "FILTER_SANITIZE_URL" section where it says, "Remove all characters except letters, digits and \$-_.+!*'(),{|\\^~[]`<>#%";/?:@&=." is there a reason why there is a double backslash (\\)? Shouldn't there only be one backslash if it's saying that backslashes are allowed?

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

-35

[Dmitry Snytkine](#)**11 years ago**

Beware that FILTER_FLAG_STRIP_LOW strips NEWLINE and TAG and CARRIAGE RETURN chars. If you have a form that accepts user input in plaintext format, all the submitted text will lose all the line breaks, making it appear all on one line. This basically renders this filter useless for parsing user-submitted text, even in plain text.

[+ add a note](#)

- [Types of filters](#)
 - [Validate filters](#)
 - [Sanitize filters](#)
 - [Other filters](#)
 - [Filter flags](#)
- [Copyright © 2001-2023 The PHP Group](#)
- [My PHP.net](#)
- [Contact](#)
- [Other PHP.net sites](#)
- [Privacy policy](#)

