



- [Downloads](#)
- [Documentation](#)
- [Get Involved](#)
- [Help](#)

[PHP 5.4.27 Released](#)

[Getting Started](#)

[Introduction](#)

[A simple tutorial](#)

[Language Reference](#)

[Basic syntax](#)

[Types](#)

[Variables](#)

[Constants](#)

[Expressions](#)

[Operators](#)

[Control Structures](#)

[Functions](#)

[Classes and Objects](#)

[Namespaces](#)

[Exceptions](#)

[Generators](#)

[References Explained](#)

[Predefined Variables](#)

[Predefined Exceptions](#)

[Predefined Interfaces and Classes](#)

[Context options and parameters](#)

[Supported Protocols and Wrappers](#)

[Security](#)

[Introduction](#)

[General considerations](#)

[Installed as CGI binary](#)

[Installed as an Apache module](#)

[Filesystem Security](#)

[Database Security](#)

[Error Reporting](#)

[Using Register Globals](#)

[User Submitted Data](#)

[Magic Quotes](#)

[Hiding PHP](#)

[Keeping Current](#)

[Features](#)

[HTTP authentication with PHP](#)

[Cookies](#)

[Sessions](#)[Dealing with XForms](#)[Handling file uploads](#)[Using remote files](#)[Connection handling](#)[Persistent Database Connections](#)[Safe Mode](#)[Command line usage](#)[Garbage Collection](#)[DTrace Dynamic Tracing](#)[Function Reference](#)[Affecting PHP's Behaviour](#)[Audio Formats Manipulation](#)[Authentication Services](#)[Command Line Specific Extensions](#)[Compression and Archive Extensions](#)[Credit Card Processing](#)[Cryptography Extensions](#)[Database Extensions](#)[Date and Time Related Extensions](#)[File System Related Extensions](#)[Human Language and Character Encoding Support](#)[Image Processing and Generation](#)[Mail Related Extensions](#)[Mathematical Extensions](#)[Non-Text MIME Output](#)[Process Control Extensions](#)[Other Basic Extensions](#)[Other Services](#)[Search Engine Extensions](#)[Server Specific Extensions](#)[Session Extensions](#)[Text Processing](#)[Variable and Type Related Extensions](#)[Web Services](#)[Windows Only Extensions](#)[XML Manipulation](#)[Keyboard Shortcuts](#)[?](#)

This help

[j](#)

Next menu item

[k](#)

Previous menu item

[g p](#)

Previous man page

[g n](#)

Next man page

[G](#)

Scroll to bottom

g g

Scroll to top

g h

Goto homepage

g s

Goto search
(current page)

/

Focus search box

[« Differences From Perl
PCRE Functions »](#)

- [PHP Manual](#)
- [Function Reference](#)
- [Text Processing](#)
- [PCRE](#)
- [PCRE Patterns](#)
- [PCRE Patterns](#)
 - [PCRE regex syntax](#)
 - [Possible modifiers in regex patterns](#)
 - [Differences From Perl](#)
 - [Differences from POSIX regex](#)

Change language: English

[Edit Report a Bug](#)

Differences from POSIX regex ¶

As of PHP 5.3.0, the [POSIX Regex](#) extension is deprecated. There are a number of differences between POSIX regex and PCRE regex. This page lists the most notable ones that are necessary to know when converting to PCRE.

1. The PCRE functions require that the pattern is enclosed by [delimiters](#).
2. Unlike POSIX, the PCRE extension does not have dedicated functions for case-insensitive matching. Instead, this is supported using the *i* (*PCRE_CASELESS*) [pattern modifier](#). Other pattern modifiers are also available for changing the matching strategy.
3. The POSIX functions find the longest of the leftmost match, but PCRE stops on the first valid match. If the string doesn't match at all it makes no difference, but if it matches it may have dramatic effects on both the resulting match and the matching speed. To illustrate this difference, consider the following example from "Mastering Regular Expressions" by Jeffrey Friedl. Using the pattern *one(self)?(selfsufficient)?* on the string *oneselfsufficient* with PCRE will result in matching *oneself*, but using POSIX the result will be the full string *oneselfsufficient*. Both (sub)strings match the original string, but POSIX requires that the longest be the result.
4. The POSIX definition of a "character class" differs from that of PCRE. Simple bracket expressions to match a set of explicit characters are supported in the form of PCRE [character classes](#) but POSIX collating elements, character classes and character equivalents are not supported. Supplying an

expression with a character class that both starts and ends with `;`, `.` or `=` characters to PCRE is interpreted as an attempt to use one of these unsupported features and causes a compilation error.

Function replacements

POSIX PCRE

[ereg_replace\(\)](#) [preg_replace\(\)](#)

[ereg\(\)](#) [preg_match\(\)](#)

[eregi_replace\(\)](#) [preg_replace\(\)](#)

[eregi\(\)](#) [preg_match\(\)](#)

[split\(\)](#) [preg_split\(\)](#)

[spliti\(\)](#) [preg_split\(\)](#)

[sql_regcase\(\)](#) *No equivalent*

 [add a note](#)

User Contributed Notes 2 notes

[up](#)

[down](#)

2

[cdragon at dracoventions dot com ¶](#)

8 months ago

In regards to the previous comment that says "there are several other differences including different meaning for the symbols ([different rules for which symbols need escaping", as far as I can tell from reading

<http://www.tin.org/bin/man.cgi?section=7&topic=regex>

there are absolutely no "other differences" except in what the man page calls "Obsolete ("basic'') regular expressions". However, PHP doesn't appear to use the obsolete form of expressions in its POSIX functions.

PCRE even supports all of the POSIX named character classes such as `[:space:]`.

[up](#)

[down](#)

0

[jasen at treshna dot com ¶](#)

2 years ago

there are several other differences

including different meaning for the symbols ([different rules for which symbols need escaping (they can't be the same as both standard posix and extended posix)

you should read the full documentation for PCRE before chaging any posix regex to use pcre.

 [add a note](#)

- [Copyright © 2001-2014 The PHP Group](#)
- [My PHP.net](#)
- [Contact](#)

- [Other PHP.net sites](#)
- [Mirror sites](#)
- [Privacy policy](#)

