

# 10 Practical PHP Regular Expression Recipes

211 readers  
BY FEEDBURNER

Follow Us!

[Follow us on Twitter](#)

**Latest update** Email validation seems quite tricky. Do you have a better solution? Let's have [email validation challenge](#). And give the user the best way to validate an email format.

Mastering [regex](#) or regular expression is one advantages for you as programmer, but not everyone has time to learn something.

Not knowing how regex works doesn't mean you couldn't use the power of regex in your project.

Bookmark this list, so you can always going back here whenever you need this regex recipes.

Update :

Never though, my post will reach so many people, and cause so many reaction. But thank you to everyone, as I myself learn from my mistake.

Some credit goes to [Joey Sochacki](#) as I did use and modify some of his regular expression.

## Ebooks

[CodeIgniter Tutorial Compilation](#)

## Popular Posts

[20 Great PHP Libraries You Need to Know](#)

[101 PHP Tutorials for PHP Programmer Wannabe](#)

[10 Practical PHP Regular Expression Recipes](#)

[42 Pagination Design Inspiration](#)

[9 great UX presentations to help you build better website](#)

[.htaccess What, Why, When and How](#)

[Codeigniter tutorial - To do list application](#)

[Sphinx search introduction](#)

[CodeIgniter Tutorial - To do list application#2](#)

[Linux Commands You Need to Know as Web Developer](#)

## Recent Posts

[10 Steps to Becoming a Great Web Developer](#)

[Ultimate Collection of PHP Libraries](#)

[Submitting Form and Progressive Enhancement](#)

[CodeIgniter Tutorial Compilation](#)

[Showing the weather with PHP and Google](#)

[Weather API](#)

 

### 1. Validate email addresses

This is only basic email validation. Not recommended to use. Validate perfect email format using regex is not really efficient.

```
$email = "test@example.com";
if (preg_match('/^[^0-9][a-zA-Z0-9_]+([.][a-zA-Z0-9_]+)*@[a-zA-Z0-9_]+([.])'
} else {
    echo "Wrong email address format";
}
```

A better solution for validate email syntax is using filter\_var. Thanks to [cx42net](#) for pointing me out to filter\_var.

```
if (filter_var('test+email@fexample.com', FILTER_VALIDATE_EMAIL)) {
    echo "Your email is ok.";
} else {
    echo "Wrong email address format.";
}
```

### 2. Validate usernames

Validate username, consist of alpha-numeric (a-z, A-Z, 0-9), underscores, and has minimum 5 character and maximum 20 character. You could change the minimum character and maximum character to any number you like.

```
$username = "user_name12";
if (preg_match('/^[a-z\d_]{5,20}$/i', $username)) {
    echo "Your username is ok.";
} else {
    echo "Wrong username format.";
}
```

### 3. Validate telephone numbers

Validate US phone number

```
$phone = "(021)423-2323";
if (preg_match('/^(?(\d{3})?[-\s.]?\d{3}[-\s.]?\d{4}/x', $phone)) {
    echo "Your phone number is ok.";
} else {
    echo "Wrong phone number.";
}
```

### 4. Validate IP addresses

```
$IP = "198.168.1.78";
if (preg_match('/^((([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}([1-9]?[0-9]|10[0-9]|11[0-9]|12[0-9]|13[0-9]|14[0-9]|15[0-9]|16[0-9]|17[0-9]|18[0-9]|19[0-9]|20[0-9]|21[0-9]|22[0-9]|23[0-9]|24[0-9]|25[0-5])$'
} else {
    echo "Wrong IP address.";
}
```

## 5. Validate postal codes

Validate US Postal codes

```
$zipcode = "12345-5434";
if (preg_match("/^[0-9]{5}(-[0-9]{4})?$/i",$zipcode)) {
    echo "Your Zip code is ok.";
} else {
    echo "Wrong Zip code.";
}
```

## 6. Validate SSN

Validate US Social Security Number

```
$ssn = "333-23-2329";
if (preg_match('/^\d{3}-\d{2}-\d{4}$/', $ssn)) {
    echo "Your SSN is ok.";
} else {
    echo "Wrong SSN.";
}
```

## 7. Validate credit card

```
$cc = "378282246310005";
if (preg_match('/^(?:4[0-9]{12}(?:[0-9]{3})?|5[1-5][0-9]{14}|6011[0-9]{12})|3[0-9]{13}$/', $cc)) {
    echo "Your credit card number is ok.";
} else {
    echo "Wrong credit card number.";
}
```

## 8. Validate domain

```
$url = "http://komunitasweb.com/";
if (preg_match('/^(http|https|ftp):\/\/([A-Z0-9][A-Z0-9_-]*(?:\.[A-Z0-9][A-Z0-9_-]+)+)$/i', $url)) {
    echo "Your url is ok.";
} else {
    echo "Wrong url.";
}
```

## 9. Extract domain name from certain URL

```
$url = "http://komunitasweb.com/index.html";
preg_match('@^(?:http://)?([^\s@]+)@i', $url, $matches);
$host = $matches[1];

echo $host;
```

## 10. Highlight a word in the content

```
$text = "Sample sentence from KomunitasWeb, regex has become popular in web";
$text = preg_replace("/\b(regex)\b/i", '<span style="background:#5fc9f6">\1<');
echo $text;
```

Please note that some of this regex recipe doesn't really validate to all conditions, therefore not a perfect solution, but it will cover most of the case. It's not really worth to build perfect validation to all case, and waste your time in the process. Well that just my opinion. Feel free to share your regex magic and your opinion.

Tags: [regex](#)

If you like this article/tutorial, please share it with the world

[Del.icio.us](#) | [Stumble](#)

---

March 18th, 2009 | [Comments\(23\)](#)

**Related post**

- [Email Validation Challenge](#)
- [Getting started with XML/SWF Charts](#)
- [101 PHP Tutorials for PHP Programmer Wannabe](#)
- [PHP Tips and Tricks](#)
- [Full text search introduction](#)

## 23 Responses to "10 Practical PHP Regular Expression Recipes"



Some notice :

\_ You should use some implemented tools like ctype and filter\_var.

Filter\_var will allow you to check many strings, like emails, urls, ips, types (int, array, etc).

\_ The validator for the email address is wrong. In fact, according to the rfc, brackets are allowed too ("{", "}"). But yes, I'd like to know someone that use these :p

\_ For username, what about "Tom" ?

\_ You should mention that the validator for phone numbers is for US format.

For example, a French version would be : 01 23 45 67 89

You also can put an international version of your phone, and your validator wouldn't work for this. (Using +33 1 23 45 67 89 OR 00 33 1 23 45 67 89 for an international french phone (+ or 00 are identical)).

Comment by [cx42net](#) — March 18, 2009 @ [9:55 am](#)



@cx42net Thanks, I didn't know about filter\_var, and according to documentation it need PHP 5 >= 5.2.0. But filter\_var make validation easier.

- I note that validator email not the perfect one. To tell you the truth I never see email with bracket. Even if you have one, I really doubt you can pass your email when you register with most of the site.

- For username validation, I use /i modifier, so it will be case-insensitive.

- Yap. It is US phone number.

Comment by [Gilang Chandrasa](#) — March 18, 2009 @ [10:15 am](#)



Yes you right about emails with bracket, it's just the rfc that tells that, but I'm sure all mail hosting won't allow you to use some, so ... :p

For username, I was thinking about a smaller size than 5. But after all, I think anyone that will use this regex will modify it according to his though. I don't know how many users have a username with less than 5 chars ...

it would be interesting to know that :)

Comment by [cx42net](#) — March 18, 2009 @ [10:30 am](#)



Add additional information about username validation. Thanks to cx42net.

Comment by [Gilang Chandrasa](#) — March 18, 2009 @ [10:57 am](#)



Nice stuff! If you can include them in a class it will be very good.

Comment by [The ace](#) — March 18, 2009 @ [12:56 pm](#)

[...] the rest here: [10 Practical PHP Regular Expression Recipes | KomunitasWeb](#) Tags: - htaccess-what, css, database, javascript, jquery, phone, project, recipe, recipes, regex, [...]

Pingback by [10 Practical PHP Regular Expression Recipes | KomunitasWeb |](#)

[OurBrownies.com](#) — March 18, 2009 @ [8:26 pm](#)

[...] komunitasweb Comparte esta [...]

Pingback by [10 Expresión regular de PHP que usted debe saber — Tu Tecnólogo](#) — March 18, 2009 @ [10:23 pm](#)



Your first 8 examples claim to validate things, but validating data like e-mail addresses using regular expressions might not be the smartest thing to do. At most you can do a syntax check but validation...?

Checking e-mail addresses using regular expressions should result in either an insanely large regular expression (<http://www.ex-parrot.com/~pdw/Mail-RFC822-Address.html>) or in a syntax check that only checks the basics (e.g. `/^\S+\@\S+\.[a-z]{2,5}$/ID`). But validating them would require a check with the smtp server.

The same goes for phonenumbers, the regular expression doesn't check valid area codes so it simply check the syntax.

So these recipes are nice examples but using regular expressions should be a tool, not a goal.

Comment by [Takkie](#) — March 19, 2009 @ 2:52 am



@Takkie, I aware that this is not a perfect solution, especially for email. And yes, this is only validate syntax. Well just saying if you mean validate is real validation that you have to call each phone number to verify real phone number.

Most site will only validate basic email syntax, then they send verification through that email. If you try to register using weird email format, I believe it won't pass the validation in most site.

The goal of validation process is to reduce errors as many as possible, not to eliminate the errors.

I agree regular expression is a tool not a goal. Knowing how to use and when to use regular expression is more powerful tool.

Comment by [Gilang Chandrasa](#) — March 19, 2009 @ 3:51 am



Hello,

Please add your site at <http://www.sweebs.com>. Sweebs.com is a place where other people can find you among the best sites on the internet!

Its just started and we are collecting the best found on the net! We will be delighted to have you in the sweebs listings.

Regards

Kris

Comment by [Kris](#) — March 19, 2009 @ 4:15 am



Are you clearly sure that your regular expression for "syntax-validating" of Email address NEVER reject valid email address?

For example, [foo+bar@gmail.com](mailto:foo+bar@gmail.com), it has a plus sign in it, it's valid and commonly used, but it seems your regular expression says it's "Wrong"

Your theory, real validation can be done only through sending real email, sounds good, but what if your script reject a valid email address, then how can you go through registration process?

You must guarantee that your script never reject valid addresses. (your script can say "It's okay" for wrong addresses, but your script must not say "It's wrong" for valid addresses)

I believe it's better to edit your entry for correction, not here as one of replies. If you leave your script without correction, someday someone with a valid address will surely encounter a trouble, because of you.

Comment by [John Doe](#) — March 19, 2009 @ 4:26 am





A bunch of those look strikingly similar to mine. :P Nice.

Comment by [Joey Sochacki](#) — March 19, 2009 @ [4:37 am](#)



@John D0e. I have add additional information to email validation. Thanks.

@Joey Sochacki. I did look to your post, and modify some of them. I have updated my post and give credit for you. Thanks

Comment by [Gilang Chandrasa](#) — March 19, 2009 @ [5:21 am](#)

#### 10 распространенных регулярных выражения для PHP...

Thank you for submitting this cool story – Trackback from prog.ru...

Trackback by [prog.ru](#) — March 19, 2009 @ [6:59 am](#)



Good

Comment by [Infinite open source](#) — March 19, 2009 @ [8:35 am](#)



Regarding phone number validation, please take a look at this post on my blog. It's not ideal either, but I am looking to make it ideal one day. I need people's comments and ideas for that though.

<http://bit.ly/oyTqX>

Thanks!

Comment by [Dude with hat \(aka BTS\)](#) — March 19, 2009 @ [8:55 am](#)

[...] Go here to read the rest: [10 Practical PHP Regular Expression Recipes | KomunitasWeb](#)

[...]

Pingback by [10 Practical PHP Regular Expression Recipes | KomunitasWeb](#) — March 19, 2009 @ [4:08 pm](#)



To extract domain name from URL use `parse_url()` function ([http://www.php.net/parse\\_url](http://www.php.net/parse_url))

Comment by [German](#) — March 20, 2009 @ [10:11 am](#)

[...] [10 Practical PHP Regular Expression Recipes | KomunitasWeb](#) [...]

Pingback by [Wayne State Web Communications Blog » Blog Archive » \[Friday Links\] The Programming Edition](#) — March 21, 2009 @ [8:10 am](#)

[...] En este blog nos muestran 10 snippets para validación con PHP y Regexp. Hay validación de nombres, teléfonos, direcciones IP, dominios y tarjetas de crédito. Demo [...]

Pingback by [Destacados del desarrollo web #1 | minimalart](#) — March 21, 2009 @ [2:52 pm](#)

[...] [10 Practical PHP Regular Expression Recipes | KomunitasWeb](#) [...]

Pingback by [Mar-19-2009 php links | w3feeds](#) — March 23, 2009 @ [10:17 am](#)

[...] [10 Practical PHP Regular Expression Recipes](#) [...]

Pingback by [Saturday Geek Links on a Monday - 3/24/09 | Geeks with Tricks](#) — March 23, 2009 @ [5:59 pm](#)