

PHP Regular Expression

By : Suman Gangopadhyay

MSc | CCNP-R&S | CCNP-ISP | CCNA-Security | RHCE





REGULAR EXPRESSIONS are pattern match function of PHP.

Used with String to find out another string.

There are 2 type of Regular Expression :

- 1) POSIX Regular Expression
- 2) PERL Regular Expression



POSIX Regular Expression

BRACKETS

They are used to find range of characters in a string.

[0 - 9] Matches between 0 to 9

[a - z] Matches between a to z

[a - f] Matches between lowercase “a” to
uppercase “F”



QUANTIFIERS

The position of bracketed character sequences and single characters can be denoted by a special character known as Quantifiers.

^e It matches any string with “e” at the beginning of it.

e\$ It matches any string with “e” at the end.

E+ It matches any string with



\d matches any digit from 0 to 9

b[ia] Matches EITHER “i” or “a”
after “b”

[] Matches characters between
Square brackets.

METACHARACTER



The `.` is called the Metacharacter

`.a` Matches any character BEFORE
“a”

`a.` Matches any character AFTER “a”

RANGE OF CHARACTERS



d[lo] Matches any character either
“l” or “o” after “d”

[1-49] Searches for 1,2,3,4 and 9

NEGATION ^



- It searches for Character not in the string.
- It is done by the ^ symbol called “Caret”

t[^eo]d Matches any character which starts with “t” but not have “eo” and ends with “d”

MULTIPLIERS



- Allows us to increase number of times an item occur in the Regular Expression.

{5} Occurs 5 times

{3,7} Occurs between 3 or 7 times

{2,} Occurs at-least 2 times

***** Zero or more times

+ One or more times

? Zero or One time

ESCAPING METACHARACTER



\ Escaping Metacharacter

SHORTHAND CHARACTER CLASSES



\s Matches anything which is a
WHITESPACE including SPACE, TAB,
LINE BREAK.

\S Matches which is not a
WHITESPACE

\d Matches digits between [0-9]

SHORTHAND CHARACTER CLASSES



- 2

\D Matches which are not Digits including Alphabets, Special Characters, Whitespaces, Line breaks, Tabs.

\w Matches anything which is a character including underscore [A-Za-z0-9_]

\W Opposite of \w



NON PRINTING CHARACTERS

`\t` TAB

`\r` Carriage Return

`\n` New Line

- Carriage return function will move the cursor from EOL to BOL

PERL Regular Expression



PHP supports **PERL Style Regular Expression** using the PCRE library.

PHP requires that each pattern starts and ends with the same delimiter to differentiate it from a normal string.

Common Delimiter is “/”

Example

“/w+/”

preg_match()



`preg_match()` function checks whether a regular expression matches a specific string.

This function stops after the first match, so this is best suited for testing a regular expression more than extracting data.

Syntax

```
a = preg_match( pattern, input_string, matches_out )
```

COMMON REGEX CODES



E-MAIL VALIDATION

- a) `([a-z0-9])+\@+([a-z])(\.)+([a-z]{2,6})`
- b) `(\w+\@[a-zA-Z_]+?\.[a-zA-Z]{2,6})`
- c) `/^[^0-9][A-z0-9_]+(\.[A-z0-9_]+)*\@[A-z0-9_]+(\.[A-z0-9_]+)*[A-z]{2,4}$/`



COMMON REGEX CODES

URL VALIDATION

URL starting with HTTPS:// , HTTP:// , WWW. `((https://|http://|w{3}\.)+[a-z]{1,180}\.[a-z]{2,10})`

URL Starting with WWW `(www\.)+([a-z]{1,180}\.)+([a-z]{2,10})`

URL like google.com `\b[a-z]{1,180}\.[a-z]{1,3}\b`

COMMON REGEX CODES



INDIAN MOBILE NUMBER VALIDATION

`^[987]\d{9}$`

BANGLADESH MOBILE NUMBER VALIDATION

`0\d{10}`

COMMON REGEX CODES



INDIAN PASSPORT NUMBER VALIDATION

`\b[A-Z]{1}\d{7}\b`

INDIAN PAN CARD NUMBER VALIDATION

`\b[A-Za-z]{5}[0-9]{4}[A-Za-z]{1}\b`

COMMON REGEX CODES



INDIAN GST NUMBER VALIDATION

```
^\d{2}[A-Za-z]{5}\d{4}[A-Za-z]{1}\d{1}[a-zA-Z]{1}\d{1}$
```

INDIAN PINCODE NUMBER VALIDATION

```
\b[1-9]\d{5}\b
```

preg_match() - Example



```
<?php
```

```
    string = '/[a-zA-Z]+ \d+/';  
    if (preg_match(    string, 'June 24' )) {  
        echo 'Found a match!';  
    } else {  
        echo 'The regex pattern does not match.'; }  
}
```

```
?|
```

Email Validation !



```
<?php
    email = 'suman@example.com';
    regexp='/^([0-9][A-z0-9_]+)([.][A-z0-9_]+)*[@][A-z0-9_]+([.][A-z0-9_]+)*[.][A-z]{2,4}$ /';
if (preg_match(    regexp,    email)){
    echo 'Email address is valid.';
} else{
    echo 'Email address is <b|NOT</b| valid.';
}
?|
```

Password Checking



```
<?php
```

```
    password = 'Suman@ganguly234_****';
```

```
    if(preg_match('/^.*( ?=.{ 8,} )( ?=.*\d)( ?=.*[    a-z])( ?=.*[A-Z]).*    /',    password)) {
```

```
        echo 'Your passwords is strong.';
```

```
    } else {
```

```
        echo 'Your password is weak !';
```

```
    }
```

```
<?php
```