

PHP - Regular Expressions Summarized with some Youtube link

<https://www.youtube.com/watch?v=3RIDZEa6gW8>

Regular expressions are nothing more than a sequence or pattern of characters itself. They provide the foundation for pattern-matching functionality.

Using regular expression you can search a particular string inside a another string, you can replace one string by another string and you can split a string into many chunks.

PHP offers functions specific to two sets of regular expression functions, each corresponding to a certain type of regular expression. You can use any of them based on your comfort.

- POSIX Regular Expressions
- PERL Style Regular Expressions

POSIX Regular Expressions

The structure of a POSIX regular expression is not dissimilar to that of a typical arithmetic expression: various elements (operators) are combined to form more complex expressions.

The simplest regular expression is one that matches a single character, such as g, inside strings such as g, hagggle, or bag.

Lets give explanation for few concepts being used in POSIX regular expression. After that we will introduce you with regular expression related functions.

Brackets

Brackets ([]) have a special meaning when used in the context of regular expressions. They are used to find a range of characters.

Sr.No	Expression & Description
1	[0-9] It matches any decimal digit from 0 through 9.

2	[a-z] It matches any character from lower-case a through lowercase z.
3	[A-Z] It matches any character from uppercase A through uppercase Z.
4	[a-Z] It matches any character from lowercase a through uppercase Z.

The ranges shown above are general; you could also use the range [0-3] to match any decimal digit ranging from 0 through 3, or the range [b-v] to match any lowercase character ranging from b through v.

Quantifiers

The frequency or position of bracketed character sequences and single characters can be denoted by a special character. Each special character having a specific connotation. The +, *, ?, {int. range}, and \$ flags all follow a character sequence.

Sr.No	Expression & Description
1	p+ It matches any string containing at least one p.
2	p* It matches any string containing zero or more p's.
3	p? It matches any string containing zero or one p's.
4	p{N} It matches any string containing a sequence of N p's
5	p{2,3} It matches any string containing a sequence of two or three p's.

6	p{2, } It matches any string containing a sequence of at least two p's.
7	p\$ It matches any string with p at the end of it.
8	^p It matches any string with p at the beginning of it.

Examples

Following examples will clear your concepts about matching characters.

Sr.No	Expression & Description
1	[^a-zA-Z] It matches any string not containing any of the characters ranging from a through z and A through Z.
2	p.p It matches any string containing p, followed by any character, in turn followed by another p.
3	^. {2}\$ It matches any string containing exactly two characters.
4	(.) It matches any string enclosed within and .
5	p(hp)* It matches any string containing a p followed by zero or more instances of the sequence php.

Predefined Character Ranges

For your programming convenience several predefined character ranges, also known as character classes, are available. Character classes specify an entire range of characters, for example, the alphabet or an integer set –

Sr.No	Expression & Description
1	[[:alpha:]] It matches any string containing alphabetic characters aA through zZ.
2	[[:digit:]] It matches any string containing numerical digits 0 through 9.
3	[[:alnum:]] It matches any string containing alphanumeric characters aA through zZ and 0 through 9.
4	[[:space:]] It matches any string containing a space.