Regular Expression

Types

- POSIX Regular Expressions
- PERL Style Regular Expressions

Brackets

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.

Quantifiers

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 more p's. This is just an alternative way to use p*.
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

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.

POSIX function

Sr.No	Function & Description
	ereg()
1	The ereg() function searches a string specified by string for a string specified by pattern, returning true if the pattern is found, and false otherwise.
	ereg_replace()
2	The ereg_replace() function searches for string specified by pattern and replaces pattern with replacement if found.
	eregi()
3	The eregi() function searches throughout a string specified by pattern for a string specified by string. The search is not case sensitive.
	eregi_replace()
4	The eregi_replace() function operates exactly like ereg_replace(), except that the search for pattern in string is not case sensitive.
	split()
5	The split() function will divide a string into various elements, the boundaries of each element based on the occurrence of pattern in string.
	spliti()
6	The spliti() function operates exactly in the same manner as its sibling split(), except that it is not case sensitive.
	sql_regcase()
7	The sql_regcase() function can be thought of as a utility function, converting each character in the input parameter string into a bracketed expression containing two characters.

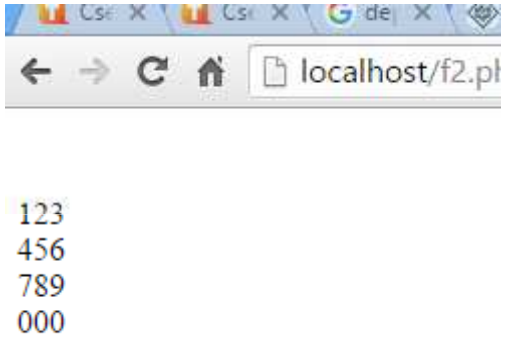
split

```
<?php

  $ip = "123.456.789.000"; // some IP address
  $iparr = split ("\.", $ip);

  print "$iparr[0] <br />";
  print "$iparr[1] <br />" ;
  print "$iparr[2] <br />"  ;
  print "$iparr[3] <br />"  ;

?>
```



PERL Compatible functions

Sr.No	Function & Description
	preg_match()
1	The preg_match() function searches string for pattern, returning true if pattern exists, and false otherwise.
	preg_match_all()
2	The preg_match_all() function matches all occurrences of pattern in string.
	preg_replace()
3	The preg_replace() function operates just like ereg_replace(), except that regular expressions can be used in the pattern and replacement input parameters.
	preg_split()
4	The preg_split() function operates exactly like split(), except that regular expressions are accepted as input parameters for pattern.
	preg_grep()
5	The preg_grep() function searches all elements of input_array, returning all elements matching the regexp pattern.
	preg_quote()
6	Quote regular expression characters

grep

```
<?php
  $foods = array("pasta", "steak", "fish", "potatoes");

  // find elements beginning with "p", followed by one or more letters.
  $p_foods = preg_grep("/p(\w+)/", $foods);

  print "Found food is " . $p_foods[0];
  print "Found food is " . $p_foods[1];
?>
```