Regular Expression(Regex)

A regular expression is a pattern of characters.

Syntax of Regex:

/pattern/modifier(s)

test() : This method is called using pattern object and returns true if string is a part of pattern.
syntax:

RegExpObject.test(string)

The String is Required. It is to be searched. It returns true if it finds a match, otherwise false.

Regex	Description	Matches (Examples)	Does Not Match
/abc/	Must contain substring abc anywhere	abc, xabc, 123abcxyz	ab, axyz
/abc/i	Case-insensitive match for abc	ABC, aBc, ABcxyz	axyz
/[abc]/	Must contain at least one of a, b, or c	a, cab, xyzab	xyz, ddf
/[^abc]/	Must contain at least one character **other than** a, b, or c	cat, the, 123d	a, abc, cbbca
/[0-9]/	Must contain at least one digit	a1, 123, the2	abc, @xyz
/[^0-9]/	Must contain at least one **non-digit**	a1, @12, xyz9	1, 123, 999
/(x|y)/	Must contain x or y	x, y, x1y2, asdx, y123	11, zzz
/^A/	Must start with A	Apple, A1, Awesome	a123, bA
/Bo*/	Must contain B followed by zero or more o	B, Bo, Boo, Born	bo, table

Regex	Description	Matches (Examples)	Does Not Match
/t$/	Must end with t	Bat, Best, 123t	top, table1
/^\d{10}$/	Exactly 10 digits	1234567890	1234, abc1234567
/^[A-Za-z]+$/	Only letters (uppercase/lowercase), at least one required	abc, XYZ, a	abc123, 1a, @b
/^\d{4}$/	Exactly 4 digits	1234, 9876	123, 12a4
/^[_@0-9]/	Username must **start** with _, @, or a digit	_abc, @user, 1name	user1, abc_1, hello123
/\W+/	Must contain at least one special character	abc@123, pass!word	password123, abcde
/^\w+([.-]?\w+)*@\w+([.]?\w+)*(\.\w{2,3})+$/	Email validation	test@gmail.com, abc.xy@domain.co.in	abc@com, @gmail.com
/^[0-9]$/	Exactly one digit (0–9)	0, 5, 9	12, a, 99
/^[A-Z]$/	Exactly one uppercase letter	A, Z, M	AB, a, 1

\	Marks the next character as either a special character or a literal. For example, n matches the character n, whereas \n matches a newline character. The sequence \\\\ matches \\ and \\(matches (, \\. Matches . in a string.
^	Matches the beginning of input.
$	Matches the end of input.
*****	Matches the preceding character zero or more times. For example, zo* matches either z or zoo.
+	Matches the preceding character one or more times. For example, zo+ matches zoo but not z.
?	Matches the preceding character zero or one time. For example, a?ve? matches the ve in never.

x|y	Matches either x or y. For example, z|wood matches z or wood. (z|w)oo matches zoo or wood.
{n}	n is a non-negative integer. Matches exactly n times. For example, o{2} does not match the o in Bob, but matches the first two os in foooood.
{n,}	In this expression, n is a non-negative integer. Matches the preceding character at least n times. For example, o{2,} does not match the o in Bob and matches all the os in foooood. The o{1,} expression is equivalent to o+ and o{0,} is equivalent to o*.
{n,m}	The m and n variables are non-negative integers. Matches the preceding character at least n and at most m times. For example, o{1,3} matches the first three os in fooooood. The o{0,1} expression is equivalent to o?.
[xyz]	A character set. Matches any one of the enclosed characters. For example, [abc] matches the a in plain.
[^xyz]	A negative character set. Matches any character that is not enclosed. For example, [^abc] matches the p in plain.
[a-z]	A range of characters. Matches any character in the specified range. For example, [a-z] matches any lowercase alphabetic character in the English alphabet.
[^m-z]	A negative range of characters. Matches any character that is not in the specified range. For example, [m-z] matches any character that is not in the range m through z.
\d	Matches a digit character. [0-9]
\D	Matches a non-digit character. [^0-9]
\w	Matches any word character including underscore. This expression is **equivalent to [A-Za-z0-9_].**
\W	Matches any non-word character. This expression is **equivalent to [^A-Za-z0-9_].**

? zero or one (optional), e.g., [+-]? matches an optional "+", "-", or an empty string.

Brackets [abc] specifies matches for the characters inside the brackets.

- [abc] Any of the characters a, b, or c
- [A-Z] Any character from uppercase A to uppercase Z
- [a-z] Any character from lowercase a to lowercase z
- [A-z] Any character from uppercase A to lowercase z

Brackets [^abc] specifies matches for any character NOT between the brackets.

- [^abc] Not any of the characters a, b, or c
- [^A-Z] Not any character from uppercase A to uppercase Z
- [^a-z] Not any character from lowercase a to lowercase z
- [^A-z] Not any character from uppercase A to lowercase z