Table 18.1: Special Sequence Characters in Regular Expressions

Character	Its description
\d	Represents any digit (0 to 9)
\D	Represents any non-digit
\s	Represents white space. Ex: \t\n\r\f\v
\S	Represents non-white space character
\w	Represents any alphanumeric (A to Z, a to z, 0 to 9)
\W	Represents non-alphanumeric
\b	Represents a space around words

Table 18.1: Special Sequence Characters in Regular Expressions

Character	Its description
\d	Represents any digit (0 to 9)
\D	Represents any non-digit
\s	Represents white space. Ex: \t\n\r\f\v
\S	Represents non-white space character
\w	Represents any alphanumeric (A to Z, a to z, 0 to 9)
\W	Represents non-alphanumeric
\b	Represents a space around words

Quantifiers in Regular Expressions

In regular expressions, some characters represent more than one character to be matched in the string. Such characters are called 'quantifiers'. For example, if we write '+' it represents 1 or more repetitions of the preceding character. Hence, if we write an expression as: r'\d+', this indicates that all numeric digits which occur for 1 or more times should be extracted. Table 18.2 shows quantifiers available in Python:

Table 18.2: Quantifiers Used in Regular Expressions

Character	Its description
+	1 or more repetitions of the preceding regexp
*	0 or more repetitions of the preceding regexp
?	0 or 1 repetitions of the preceding regexp
{m}	Exactly m occurrences

Table 18.3: Special Characters in Regular Expressions

Character	Its description
\	Escape special character nature
.	Matches any character except new line
^	Matches beginning of a string

Character	Its description
$	Matches ending of a string
[...]	Denotes a set of possible characters. Ex: [6b-d] matches any characters '6', 'b', 'c' or 'd'
[^...]	Matches every character except the ones inside brackets. Ex: [^a-c6] matches any character except 'a', 'b', 'c' or '6'
(...)	Matches the regular expression inside the parentheses and the result can be captured.
R | S	Matches either regex R or regex S