In summary, the basic wildcard and repetition operators are:

* period (.)—wildcard; matches a single character.
* question mark (\? or ?)—matches 0 or 1 of the previous character, grouping, or wildcard. (This operator differs depending on whether you are using basic or extended regular expressions.)
* asterisk(\*)—matches zero or more of the previous character, grouping, or wildcard.
* plus(\+ or +)—matches one or more of the previous character, grouping, or wildcard. (This operator differs depending on whether you are using basic or extended regular expressions.

**Predefined Character Classes**

Most regular expression languages support some form of predefined character classes. When used between brackets, these define commonly used sets of characters. The most broadly supported set of predefined character classes are the POSIX character classes:

* [:alnum:]—all alphanumeric characters (a-z, A-Z, and 0-9).
* [:alpha:]—all alphabetic characters (a-z, A-Z).
* [:blank:]—all whitespace within a line (spaces or tabs).
* [:cntrl:]—all control characters (ASCII 0-31).
* [:digit:]—all numbers.
* [:graph:]—all alphanumeric or punctuation characters.
* [:lower:]—all lowercase letters (a-z).
* [:print:]—all printable characters (opposite of [:cntrl:], same as the union of [:graph:] and [:space:]).
* [:punct:]—all punctuation characters
* [:space:]—all whitespace characters (space, tab, newline, carriage return, form feed, and vertical tab). (See note below about compatibility.)
* [:upper:]—all uppercase letters.
* [:xdigit:]—all hexadecimal digits (0-9, a-f, A-F).

The basic modifiers are:

* /g—replace globally. Without this flag, a substitution command replaces only the first matching occurrence per line. With this flag, a substitution command also replaces subsequent matches.
* /i—use case insensitive matching (Perl extension; equivalent to grep -i).
* /m—multiline matching (Perl extension). the $ and ^ anchors should match at newline boundaries in addition to matching at the beginning an end of the string as a whole. The dot (.) does not match newline characters.
* /o—compile once (Perl extension). In Perl, if a regular expression includes a variable as part of the pattern, the regular expression engine must recompile the expression every time it is used because the variable contents might have changed.

If you know that the contents will not change after they are set the first time, the /o flag disables recompilation of the expression. For regular expressions that do not contain variables, this switch has no effect.

* /s—single-line matching (Perl extension). The $ and ^ anchors should not match at newline boundaries. With this modifier, they only match at the very beginning and end of the string as a whole. The dot (.) matches newline characters just like any other character.
* /x—extend readability (Perl extension). This mode causes matching to ignore all whitespace between tokens in the expression unless quoted or wrapped in brackets (in most languages) and to treat a hash mark (#) as the start of a single-line comment.