Basic Regular Expressions (BRE)

WE'LL COVER THE FOLLOWING

Metachar `.`

Matches any single character.

Example:

`x.z` matches `"xyz"`, etc., but within bracket expressions, the dot character matches a literal dot, e.g., `[x.y]` matches only `"x"`, `"."`, or `"y"`

Metachar `[ ]`

Matches a single character that is contained within the brackets.

Example:

`[xyz]` matches `"x"`, `"y"`, or `"z"`, where `[a-z]` specifies a range which matches any lowercase letter from `"a"` to `"z"`. Note that the `-` character is treated as a literal character if it is the last or the first e.g., `[-xyz]`

Metachar `[^ ]`

Matches a single character that is not contained within the brackets.

Example:

`[^xyz]` matches any character other than `"x"`, `"y"`, or `"z"`.

Metachar `^`

Matches the starting position within the string.

Example:

`^[xterm]` matches any string that starts with `xterm`.

Metachar `$`

Matches the ending position of the string or the position just before a string-ending newline.

Example:

`[mb]at$` matches `"mat"` and `"bat"`, but only at the end of the string or line.

Metachar `( )`

The string matched within the parentheses can be recalled later, also called a `block` or capturing `group`.

Example:

`([0-9]+)([a-z]+)` the first group matches atleast one digit and the second group atleast one alphabet.

Metachar `*`

Matches the preceding element zero or more times.

Example:

`xy*z` matches `"xy"`, `"xyz"`, `"xyyz"`, etc.

Metachar `{m,n}`

Matches the preceding element at least `m` and not more than `n` times.

Example:

Example:

`Y{2,3}` matches only `"YY"`, `"YYY".`