

CHEATSHEET: REGULAR EXPRESSIONS

WILDCARDS

Regex	Matches
.	Any character
\w	Any letter or digit
\d	Any digit
[]	Custom wildcard: [GTC] # Matches G,T, or C [A-Z] # Matches cap letters [1-5] # Matches numbers 1-5
\s	Any whitespace character
\n, \r	Newline or return character
\t	Tab

SPECIAL SYMBOLS AND QUANTIFIERS

Regex	Matches
^	Line beginning
\$	Line ending
\	Escape a special symbol: . # Match anything \. # Match a period \\ # Match a backslash
+	Match the previous character 1 or more times
*	Match the previous character 0 or more times
{}	Match the previous character a specified amount of times: a{4} # Match 4 a's a{2,6} # Match at least 2 but no more than 6 a's

CAPTURING TEXT WITH PARENTHESES

Search string	Regex	Result
filename-900.txt	Search: (\w+)-\d+\.txt Replace: \1\.txt	filename.txt
specimen123_XYZ_10	Search: (\w+)_w+(\d+) Replace: \1_2	specimen123_10

CHEATSHEET: REGULAR EXPRESSIONS

MODIFYING GREP USAGE

Argument	Purpose
grep -E (or use egrep)	Enable extended regular expression Note: Uses [] for character ranges instead of standard wildcards (e.g. [A-Za-z0-9] for \w)
grep -c	Count occurrences
grep -v	Show only lines that <i>don't</i> match
grep -i	Case insensitive search

PYTHON RE MODULE

Note that the re module captures using parentheses but uses `.group()` syntax, not `\1`, to reference captured groups.

Function	Purpose
<code>re.search(regex, string)</code>	Search a string for a particular regex
<code>re.split(regex, string)</code>	Split a string on a regular expression
<code>re.findall(regex, string)</code>	Return a list of all regex matches in string
<code>re.sub(regex, string, replacement)</code>	Replace occurrences of the <i>regex</i> in a string with the replacement string