

Regular Expressions Cheatsheet

A regular expression is a coded string of characters that allows one to find and replace patterns in text document.

Common Symbols

1. Most characters match themselves.
2. **^** and **\$** refer to the beginning and end of a line.
3. **.** symbolizes any character. If you actually want to match a ".", use **\.**
4. ***** means 0 or more of something
5. **+** means 1 or more of something
6. Sets of characters can be enclosed in **[]**.
 1. **[xyz]** matches any of the characters **x**, **y**, or **z**
 2. **[a-z]** matches any of the characters from **a** to **z**
 3. **[^xyz]** matches anything *but* the characters **x**, **y**, or **z**
7. Non-printing characters include **\n**, **\t** (newline and tab)
8. Special characters include
 1. **\s** (whitespace characters like spaces and tabs)
 2. **\w** (word characters, including letters and numbers)
 3. **\d** (numbers, digits)
9. Curly braces indicate how many of a pattern are being matched.
 1. **\s{2}** matches exactly 2 whitespace characters
 2. **\s{2,}** matches 2 or more whitespace characters
 3. **\s{2,4}** matches 2-4 whitespace characters
10. Parentheses **()** can be used to form a series of logical groups. These can then be referred to as **\1**, **\2**, **\3** in replace patterns. (Note that VS Code uses **\$1**, **\$2**, etc.)

Example Patterns and what they will find (match)

1. **.*** matches any sequence of 0 or more characters (any text)
2. **\s+** matches all spaces (and tabs)
3. **\w+** matches a word with 1 or more characters
4. **^\d.*\$** matches any line that begins with a number
5. **\d+\.\d+** matches a number with a decimal point in it
6. **\w[\w]+\.** matches a sentence

Example Uses

Find: ,
Replace: \t

Result: Replaces all the commas with tabs.

Find: _ _ (Each underline is actually a space that otherwise wouldn't be visible here.)

Replace: _ (The underline represents a single space character.)

Result: Converts two spaces to a single space.

Find: \n\n

Replace: \n

Result: Takes a double-spaced document and makes it single-spaced.

Find: (\w+) (\w+)

Replace: \$2 \$1

Result: Takes two words and reverses them.

Find: \w+\$

Replace:

Result: Removes the last word on the line.

Find: ^

Replace: **STUDENT:**

Result: Puts "STUDENT:" at the beginning of a line.

Find: **PHONE:** \(\\d\\d\\d\\d) \\d\\d\\d-\\d\\d\\d\\d

Replace:

Result: Removes phone numbers from each line. Note that parentheses () have to be escaped as \\(and \\) because otherwise, they'll be interpreted as grouping indicators.

Changing Case

Different text editors have different strategies for manipulating upper- and lowercase letters. In VS Code, from menu bar select View > Command Palette... > and enter **>Transform** to see a list of operations you can perform on selected text.

Undo!

If you make a mistake, *Command-z* on Apple computers or *control-z* on Windows machines will undo the previous operation so you can try again.

Workflow

You'll often need to perform a series of Find/Replace commands in order to arrive at the final version of what you're looking for.