**Regex cheat sheet:**

**Create a regular expression:**

* let re = /ab\*c/ - some pattern to match between the /slashes/.
* let re = new RegExp('ab\*c') - same pattern to match passed as a string when initializing RegExp instance.

**Character classes:**

* \d – any digit.
* \D – any character that is not a digit.
* \w – any alphanumeric character (including underscore \_).
* \W – any character that isn’t alphanumeric (and not underscore \_).
* . – (dot) matches any character until the \n (enter) or \r. Cancelled inside a group.
* \ - escaping character.

**Quantifiers:**

Quantifiers indicate numbers of characters or expressions to match.

* \* - matches 0 (zero) or more times.
* + - matches 1 or more times.
* ? – matches 0 or 1 times.
* {n} – matches exactly n times.

**Groups and ranges:**

* [a-zA-Z0-9] – matches any of the character that inside the square brackets. Matches a-z characters, A-Z and all the digits.
* [\w] = [A-Za-z0-9]
* [^a-g] – matches everything that isn’t inside the square brackets.
* x|y – matches x or y.

**Methods:**

* match(regex) – someString.match(regex) : returns an array with the first match (string) that contains the regex pattern in it. If no match found returns null. Will not return () groups with the g flag.
* matchAll(regex) – someString.matchAll(regex) : get all the matches. Need to use with the g flag. Returns an iterator so need to use … or a for loop.
* exec() – regex.exec(myString) : with every regex.exec call the regex object will be updated.

while (result = regex.exec(myString)) {

console.log(result, regex.lastIndex)}

* note that some string methods accept regular expressions (replace, replaceAll, search, split) as parameters.

**Modifier:**

Placed after the regex - /[^a-z]/g

* g – global match (finds all matches not only the first).
* i – case-insensitive matching.
* m – multiline matching.