# **Regular Expressions (Regex) Cheatsheet**

## **Character Classes**

* .: Any character except a newline
* \d: Any digit (0-9)
* \D: Any non-digit
* \w: Any word character (a-z, A-Z, 0-9, \_)
* \W: Any non-word character
* \s: Any whitespace character
* \S: Any non-whitespace character

## **Quantifiers**

* \*: 0 or more
* +: 1 or more
* ?: 0 or 1
* {n}: Exactly n times
* {n,}: At least n times
* {n, m}: Between n and m times

## **Anchors**

* ^: Start of a string
* $: End of a string
* \b: Word boundary

## **Groups and Alternation**

* ( ): Capture group
* |: Alternation (OR)

## **Character Escapes**

* \: Escape character

## **Special Characters**

* [ ]: Character class
* [^ ]: Negation within a character class
* .: Any character except a newline
* \*, +, ?, { }, (), |, ^, $: Special characters

## **Examples**

* ^\d{3}$: Matches a 3-digit number
* ^\w+@\w+\.\w+$: Matches an email address
* (red|blue) car: Matches "red car" or "blue car"

For more detailed information on regex patterns, you can refer to online resources or tutorials specific to regular expressions.

## **Lookahead and Lookbehind**

* (?<!: negative look behind
* (?<=: positive look behind
* (?=: positive look ahead
* (?!: negative look ahead

## **Check if a string *does not contain* specific\_string**

^(?!specific\_string).\*

In this pattern:

1. ^ asserts the start of the string.
2. (?!specific\_string) is the negative lookahead assertion. Replace "specific\_string" with the string you want to avoid matching.
3. .\* matches any character (.) zero or more times (\*), effectively matching any string.

So, the regex ^(?!specific\_string).\* will match any string except those that start with "specific\_string."