Regex Cheatsheet

A detailed cheat sheet featuring numerous regex examples and explanations on how regex works. Explore literal matching, metacharacters, groups, lookahead/lookbehind, alternation, and practical patterns for emails, URLs, dates, and code extraction.

Basic Regex Patterns

Literal & Simple Patterns

`/cat/`
Matches the literal string “cat”.
Example: In “concatenate”, finds “cat” at position 0.

`/dog/`
Matches the literal string “dog”.
Example: In “dogma”, finds “dog” at the start.

`/123/`
Matches the literal number sequence “123”.
Example: In “abc123xyz”, finds “123”.

`/hello/`
Matches the literal word “hello”.
Example: In “well, hello there”, finds “hello”.

`world`
Matches the literal word “world”.
Example: In “hello world”, finds “world”.

`a.b`
The dot (.) matches any character (except newline), so it finds patterns like “acb” or “a-b”.

Anchors & Boundaries

`^Hello`
Matches “Hello” only at the beginning of a string.
Example: In “Hello there”, matches at index 0.

`world$`
Matches “world” only at the end of a string.
Example: In “Hello world”, matches at the end.

`^\d+`
Matches one or more digits at the start of a string.
Example: In “123abc”, matches “123”.

`/\w+$/`
Matches one or more word characters at the end of a string.
Example: In “foo bar”, matches “bar”.

`/^start/`
Matches the literal word “start” at the beginning.

`/end$/`
Matches the literal word “end” at the end.

Character Classes

`/[abc]/`
Matches any one of the characters: a, b, or c.

`/[A-Z]/`
Matches any uppercase letter.

`/[0-9]/`
Matches any digit from 0 to 9.

`/[a-zA-Z]/`
Matches any letter, regardless of case.

`/[^0-9]/`
Matches any character that is not a digit.

`/[\s]/`
Matches any whitespace character (space, tab, newline).

Quantifiers

`/a*/`
Matches zero or more occurrences of “a”.
Example: Matches “”, “a”, “aa”, etc.

`/a+/`
Matches one or more occurrences of “a”.
Example: Matches “a”, “aa”, but not “”.

`/a?/`
Matches zero or one occurrence of “a”.
Example: Matches “” or “a”.

`/a{3}/`
Matches exactly three “a” characters.

`/a{2,}/`
Matches two or more occurrences of “a”.

`/a{2,4}/`
Matches between two and four occurrences of “a”.

Advanced Regex Techniques

Grouping & Capturing

`/(cat)/`
Captures the sequence “cat” into group 1.
Useful for backreferences.

`/(dog|cat)/`
Matches either “dog” or “cat” and captures the match.

`/(\d{3})-(\d{2})-(\d{4})/`
Matches a pattern like a Social Security Number with three capturing groups.

`/(?P<area>\d{3})/`
Uses a named capturing group “area” to capture three digits.

`/(?:abc)/`
A non-capturing group for the literal “abc”; grouping without storing the match.

`\1`
Backreference to the first captured group. Ensures the same text is repeated.

Lookahead & Lookbehind

`/(?=abc)/`	Positive lookahead: Asserts that “abc” follows without consuming characters.
`/(?!abc)/`	Negative lookahead: Ensures that “abc” does not follow the current position.
`/(?<=abc)def/`	Positive lookbehind: Matches “def” only if preceded by “abc”.
`/(?<!abc)def/`	Negative lookbehind: Matches “def” only if not preceded by “abc”.
`/foo(?=bar)/`	Matches “foo” only if it is immediately followed by “bar”.
`/bar(?<!foo)/`	Matches “bar” only if it is not immediately preceded by “foo”.

Alternation & Escaping

`/cat|dog/`	Matches either “cat” or “dog” using alternation.
`/a\+b/`	Escapes the plus sign to match the literal string “a+b”.
`/\[abc\]/`	Escapes square brackets to match the literal string “[abc]”.
`/\./`	Escapes the dot to match a literal period.
`/\\/`	Matches a single backslash character.
`/(a|b)c/`	Matches either “ac” or “bc” by grouping alternatives.

Practical Regex Examples

Email & URL Patterns

Email Pattern 1	`/^[\w.-]+@[\w.-]+\.[A-Za-z]{2,6}$/` Matches a standard email address format.
Email Pattern 2	`/^[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+$/` A robust pattern for validating emails.
URL Pattern 1	`/^(https?://)?(www\.)?[\w-]+\.[a-z]{2,}$/` Matches basic URLs with optional http/https and www.
URL Pattern 2	`/^(ftp|http|https):\/\/[\w.-]+(?:\.[\w\.-]+)+[\w\-\._~:/?#[\]@!$&'()*+,;=.]+$/` A more comprehensive URL matching pattern.
Simple URL	`/^(www\.)?[\w-]+\.[a-z]{2,}$/` Matches URLs without protocol.
Email with mailto	`/(mailto:)?[\w.-]+@[\w.-]+\.[A-Za-z]{2,6}/` Matches an email address with an optional “mailto:” prefix.

Date & Time Formats

ISO Date	`/^\d{4}-\d{2}-\d{2}$/` Matches dates in YYYY-MM-DD format.
US Date	`/^\d{2}/\d{2}/\d{4}$/` Matches dates in MM/DD/YYYY format.
European Date	`/^\d{2}-\d{2}-\d{4}$/` Matches dates in DD-MM-YYYY format.
24-Hour Time	`/^\d{2}:\d{2}:\d{2}$/` Matches time in HH:MM:SS format.
12-Hour Time	`/^\d{1,2}:\d{2}(?:AM|PM)$/` Matches time in 12-hour format with AM/PM.
Day of Week	`/^(Mon|Tue|Wed|Thu|Fri|Sat|Sun)$/` Matches abbreviated days of the week.

Code Snippets & Extraction

Python Comment	`/^\s*#.*$/` Matches a Python comment line starting with ‘#’.
HTML Tag	`/<\/?[a-zA-Z]+(\s+[a-zA-Z]+="[^"]*")*\s*>/` Matches a simple HTML tag with optional attributes.
JS Console Log	`/console\.log\(.*\);/` Matches a JavaScript console.log statement.
Python Function	`/def\s+\w+\(.*\):/` Matches a Python function definition.
Java Main Method	`/public\s+static\s+void\s+main\(.*\)/` Matches a Java main method signature.
TODO/FIXME Comment	`/(TODO|FIXME):.*/` Matches lines with TODO or FIXME comments.
JS Variable Declaration	`/\b(?:var|let|const)\s+\w+\b/` Matches variable declarations in JavaScript.