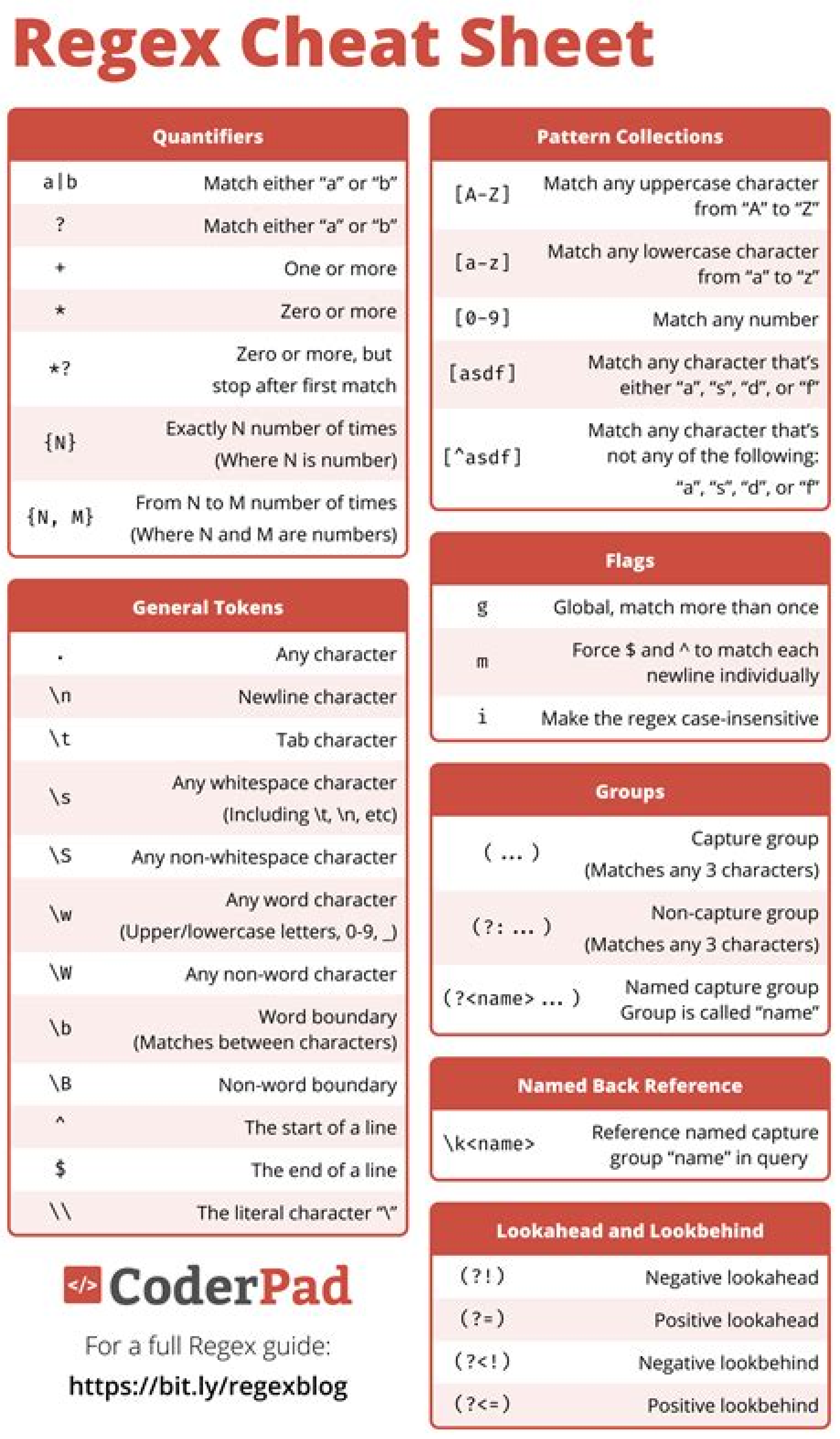
Javascript Regex Cheat Sheet



Regex Cheat Sheet

Quantifiers	
a|b	Match either "a" or "b"
?	Match either "a" or "b"
+	One or more
*	Zero or more
*?	Zero or more, but stop after first match
{N}	Exactly N number of times (Where N is number)
{N, M}	From N to M number of times (Where N and M are numbers)

General Tokens	
.	Any character
\n	Newline character
\t	Tab character
\s	Any whitespace character (Including \t, \n, etc)
\S	Any non-whitespace character
\w	Any word character (Upper/lowercase letters, 0-9, _)
\W	Any non-word character
\b	Word boundary (Matches between characters)
\B	Non-word boundary
^	The start of a line
$	The end of a line
\\	The literal character "\"

Pattern Collections	
[A-Z]	Match any uppercase character from "A" to "Z"
[a-z]	Match any lowercase character from "a" to "z"
[0-9]	Match any number
[asdf]	Match any character that's either "a", "s", "d", or "f"
[^asdf]	Match any character that's not any of the following: "a", "s", "d", or "f"

Flags	
g	Global, match more than once
m	Force $ and ^ to match each newline individually
i	Make the regex case-insensitive

Groups	
(...)	Capture group (Matches any 3 characters)
(?: ...)	Non-capture group (Matches any 3 characters)
(?<name> ...)	Named capture group Group is called "name"

Named Back Reference	
\k<name>	Reference named capture group "name" in query

Lookahead and Lookbehind	
(?!)	Negative lookahead
(?=)	Positive lookahead
(?<!)	Negative lookbehind
(?<=)	Positive lookbehind

CoderPad

For a full Regex guide:
https://bit.ly/regexblog

JavaScript regex cheat sheet is an invaluable resource for developers looking to enhance their skills in pattern matching and string manipulation. Regular expressions (regex) are powerful tools that allow you to search, match, and manipulate strings in a flexible and efficient manner. This article will serve as a comprehensive guide to JavaScript regex, breaking down its syntax, common patterns, and practical usage tips.

What is Regex?

Regular expressions are sequences of characters that form a search pattern. They are used in programming languages, including JavaScript, to perform operations such as searching, replacing, or validating strings. Regex is essential for tasks like form validation, data scraping, and text processing.

Basic Syntax of JavaScript Regex

In JavaScript, regex can be created using two syntaxes:

1. Literal notation: `/pattern/flags`

2. Constructor notation: `new RegExp('pattern', 'flags')`

For example:
```javascript
const regex1 = /abc/i; // Using literal notation
const regex2 = new RegExp('abc', 'i'); // Using constructor notation
```

The `flags` can modify the behavior of the regex. Common flags include:
- g: Global search, finds all matches rather than stopping after the first match.
- i: Case-insensitive search.
- m: Multi-line search.

Common Regex Patterns

Understanding common regex patterns will make it easier to construct your own expressions. Here are some frequently used patterns:

1. Character Classes

Character classes allow you to define a set of characters to match:
- `[abc]`: Matches any one of the characters a, b, or c.
- `[^abc]`: Matches any character except a, b, or c.
- `[a-z]`: Matches any lowercase letter.
- `[A-Z]`: Matches any uppercase letter.
- `[0-9]`: Matches any digit.

2. Quantifiers

Quantifiers specify how many times a character or group should be matched:
- ``: Matches 0 or more times.
- `+`: Matches 1 or more times.
- `?`: Matches 0 or 1 time.
- `{n}`: Matches exactly n times.
- `{n,}`: Matches n or more times.
- `{n,m}`: Matches between n and m times.

3. Anchors

Anchors are used to specify positions in the string:
- `^`: Matches the beginning of a string.
- `$`: Matches the end of a string.

4. Special Characters

Certain characters have special meanings in regex:
- `.`: Matches any single character except line terminators.
- `\`: Escapes a special character.
- `\d`: Matches any digit (equivalent to `[0-9]`).
- `\D`: Matches any non-digit character.
- `\w`: Matches any word character (equivalent to `[a-zA-Z0-9_]`).
- `\W`: Matches any non-word character.
- `\s`: Matches any whitespace character (spaces, tabs, line breaks).
- `\S`: Matches any non-whitespace character.

Practical Examples of JavaScript Regex

Let’s explore some practical applications of regex in JavaScript.

1. Validating Email Addresses

A common use case for regex is to validate email addresses. A simple regex pattern for this could be:
```javascript
const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
```
You can use this regex pattern to check if an email address is valid:
```javascript
const isValidEmail = (email) => emailRegex.test(email);
```

2. Extracting Numbers from a String

If you want to extract all numbers from a given string, you can use:
```javascript
const string = "There are 2 apples and 3 bananas.";
const numbers = string.match(/\d+/g); // ["2", "3"]
```

3. Replacing Text

You can use regex to replace certain patterns in a string. For example, to replace all instances of "cat" with "dog":
```javascript
const text = "The cat sat on the catwalk.";
const newText = text.replace(/cat/g, "dog");
```

Advanced Regex Techniques

Once you are comfortable with the basics, you can explore advanced regex techniques.

1. Lookaheads and Lookbehinds

Lookaheads and lookbehinds are zero-width assertions that allow you to match a string based on what follows or precedes it:
- Lookahead: `(?=...)` ensures that what follows is a certain pattern.
- Lookbehind: `(?<=...)` ensures that what precedes is a certain pattern.

Example:
```javascript
const lookaheadRegex = /\d(?= years)/; // Matches a digit followed by " years"
```

2. Non-capturing Groups

If you want to group parts of your regex without capturing them, use `(?:...)`. This is useful when you need to apply quantifiers but do not need to retrieve the matched text:
```javascript
const regex = /(?:abc)+/; // Matches one or more occurrences of "abc"
```

Tips for Using JavaScript Regex

To make the most of regex, consider these tips:

- Test your regex patterns using online tools like regex101.com to ensure they work as expected.
- Use comments to document complex regex patterns for better readability.
- Be cautious with greedy vs. lazy quantifiers. Use `?` or `+?` for lazy matching when needed.
- Always escape special characters when you want to match them literally.

Conclusion

In summary, a **JavaScript regex cheat sheet** is a powerful tool that every developer should keep handy. Understanding the syntax, common patterns, and practical applications of regex can significantly enhance your ability to manipulate strings in JavaScript. By mastering regex, you'll be equipped to handle a variety of tasks, from simple string searches to complex data validation. Whether you're a beginner or an experienced developer, honing your regex skills will pay dividends in your coding journey.

Frequently Asked Questions

What is a JavaScript regex cheat sheet?

A JavaScript regex cheat sheet is a quick reference guide that summarizes the syntax and usage of regular expressions in JavaScript, helping developers quickly recall patterns, flags, and methods.

What are the basic components of a regex pattern in JavaScript?

The basic components include literals (characters), meta-characters (like ., , ?, +, etc.), character classes (like [a-z]), anchors (^ for start, $ for end), and quantifiers (like {n}, {n,}, {n,m}).

How do you create a regular expression in JavaScript?

You can create a regex in JavaScript using two methods: using the RegExp constructor (e.g., new RegExp('pattern')) or by using regex literals enclosed in slashes (e.g., /pattern/).

What are some common flags used in JavaScript regex?

Common flags include 'g' for global search, 'i' for case-insensitive search, and 'm' for multi-line search.

How can you test a regex pattern against a string in JavaScript?

You can use the .test() method to test a regex pattern against a string, which returns true or false, or the .exec() method to retrieve matched results.

What is the difference between greedy and lazy quantifiers in regex?

Greedy quantifiers (like) match as much text as possible, while lazy quantifiers (like ?) match as little text as necessary to satisfy the pattern.

How can you escape special characters in a regex pattern?

You can escape special characters in a regex pattern by prefixing them with a backslash (e.g., \. to match a dot, \$ to match a dollar sign).

What is a character class in regex, and how do you use it?

A character class is a set of characters enclosed in square brackets (e.g., [abc]) that matches any single character from the set. You can also use ranges like [a-z] to match any lowercase letter.

Where can I find comprehensive JavaScript regex cheat sheets online?

You can find comprehensive JavaScript regex cheat sheets on websites like MDN Web Docs, Regex101, or various programming blogs that offer downloadable PDFs.

Javascript Regex Cheat Sheet

javascript - When should I use ?? (nullish coalescing) vs || (logical ...
The nullish coalescing operator (??) in JavaScript only considers null or undefined as "nullish" values. If the left-hand side is any other value, even falsy values like "" (empty string), 0, or ...

Which equals operator (== vs ===) should be used in JavaScript ...
Dec 11, 2008 · I'm using JSLint to go through JavaScript, and it's returning many suggestions to replace == (two equals signs) with === (three equals signs) when doing things like comparing ...

What does the !! (double exclamation mark) operator do in ...
Novice JavaScript developers need to know that the "not not" operator is using implicitly the original loose comparison method instead of the exact === or !== operators and also the ...

What is the purpose of the dollar sign in JavaScript?
Mar 29, 2022 · Javascript does have types; and in any case, how is the dollar sign even related to that? It's just a character that happens to be a legal identifier in Javascript.

How do you use the ? : (conditional) operator in JavaScript?
Jun 7, 2011 · The conditional (ternary) operator is the only JavaScript operator that takes three operands. This operator is frequently used as a shortcut for the if statement.

How to use OR condition in a JavaScript IF statement?
Mar 2, 2010 · How to use OR condition in a JavaScript IF statement? Asked 15 years, 5 months ago Modified 2 years, 6 months ago Viewed 874k times

javascript - What does [object Object] mean? - Stack Overflow
In JavaScript there are 7 primitive types: undefined, null, boolean, string, number, bigint and symbol. Everything else is an object. The primitive types boolean, string and number can be ...

What does ${} (dollar sign and curly braces) mean in a string in ...
Mar 7, 2016 · What does $ {} (dollar sign and curly braces) mean in a string in JavaScript? Asked 9 years, 4 months ago Modified 1 year, 7 months ago Viewed 418k times

Wait 5 seconds before executing next line - Stack Overflow
This function below doesn’t work like I want it to; being a JS novice I can’t figure out why. I need it to wait 5 seconds before checking whether the newState is -1. Currently, it doesn’t wait, i...

How to close current tab in a browser window? - Stack Overflow
Jan 16, 2010 · Find solutions to close a browser tab using JavaScript on Stack Overflow. Explore methods and limitations for different browsers.

javascript - When should I use ?? (nullish coalescing) vs || (logical ...
The nullish coalescing operator (??) in JavaScript only considers null or undefined as "nullish" values. If the left-hand side is any other value, even falsy values like "" (empty string), 0, or ...

Which equals operator (== vs ===) should be used in JavaScript ...
Dec 11, 2008 · I'm using JSLint to go through JavaScript, and it's returning many suggestions to replace == (two equals signs) with === (three equals signs) when doing things like comparing ...

What does the !! (double exclamation mark) operator do in …
Novice JavaScript developers need to know that the "not not" operator is using implicitly the original loose comparison method instead of the exact === or !== operators and also the …

What is the purpose of the dollar sign in JavaScript?
Mar 29, 2022 · Javascript does have types; and in any case, how is the dollar sign even related to that? It's just a character that happens to be a legal identifier in Javascript.

How do you use the ? : (conditional) operator in JavaScript?
Jun 7, 2011 · The conditional (ternary) operator is the only JavaScript operator that takes three operands. This operator is frequently used as a shortcut for the if statement.

How to use OR condition in a JavaScript IF statement?
Mar 2, 2010 · How to use OR condition in a JavaScript IF statement? Asked 15 years, 5 months ago Modified 2 years, 6 months ago Viewed 874k times

javascript - What does [object Object] mean? - Stack Overflow
In JavaScript there are 7 primitive types: undefined, null, boolean, string, number, bigint and symbol. Everything else is an object. The primitive types boolean, string and number can be …

What does ${} (dollar sign and curly braces) mean in a string in …
Mar 7, 2016 · What does $ {} (dollar sign and curly braces) mean in a string in JavaScript? Asked 9 years, 4 months ago Modified 1 year, 7 months ago Viewed 418k times

Wait 5 seconds before executing next line - Stack Overflow
This function below doesn't work like I want it to; being a JS novice I can't figure out why. I need it to wait 5 seconds before checking whether the newState is -1. Currently, it doesn't wait, i...

How to close current tab in a browser window? - Stack Overflow
Jan 16, 2010 · Find solutions to close a browser tab using JavaScript on Stack Overflow. Explore methods and limitations for different browsers.

Master JavaScript with our comprehensive regex cheat sheet! Discover essential patterns and tips to enhance your coding skills. Learn more and boost your efficiency!

Back to Home