

Quoted string

Match either single- or double-quoted strings.

```
(["'"])(?: (?!\\) )\2.)*?\1
```

Token-by-token breakdown

Token	Meaning
(	start of a capturing group
["']	any of: '"', ''', ""
)	end of group
(?:	start of a non-capturing group
(?=	start of a positive lookahead — followed by
(	start of a capturing group
\\?	optional (zero or one): literal "\"
)	end of group
)	end of group
\2	backreference to group 2
.	any character (except newline)
)*?	end of group, zero or more (lazy — as few as possible)
\1	backreference to group 1

Quick usage in 7 languages

JavaScript	<pre>const re = /(["'])(?: (?!\\))\2.)*?\1/; re.test(value);</pre>
Python	<pre>import re re.match(r'(["'])(?: (?!\\))\2.)*?\1', value)</pre>
Java	<pre>Pattern.compile("(["'\\'])(?: (?!\\\\))\2.)*?\1").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"(["'\\'])(?: (?!\\))\2.)*?\1");</pre>
Go	<pre>regexp.MustCompile(`(["'])(?: (?!\\) )\2.)*?\1`).MatchString(value)</pre>
Ruby	<pre>/(["'])(?: (?!\\))\2.)*?\1/.match?(value)</pre>
PHP	<pre>preg_match('~(["'])(?: (?!\\))\2.)*?\1~', \$value);</pre>

Common pitfalls

- **Not anchored.** Without `^` and `$` this can match a substring anywhere in the input — add anchors if you need the whole value to conform.
- **Escape it correctly per language.** In Java and JavaScript strings each backslash must be doubled (`\\d`); in Python, Go, and C# use raw/verbatim strings so the backslashes survive.
- **Validate beyond format.** Matching the format doesn't guarantee the value is real. Confirm the quoted string against a source of truth (database, API, or checksum) where it matters.