

Line comment (// or #)

Single-line comment in C-family or scripting languages.

```
(\\\/|#) .* $
```

Token-by-token breakdown

Token	Meaning
(	start of a capturing group
\\	literal "/"
\/	literal "/"
	OR — try the alternative
#	literal text "#"
)	end of group
.*	zero or more of any character
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

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

Common pitfalls

- **Anchored to the whole string.** This pattern uses \$, so it requires the entire input to match. To find it inside a longer text, drop the anchors and use the global (g) flag.
- **Greedy “.*”.** A greedy .* /.+ can match more than intended. Use a lazy version (.*?) or a negated class ([^...]) to stop at the right place.
- **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 line comment (// or #) against a source of truth (database, API, or checksum) where it matters.