

Block comment (/ * */)

C-style multi-line comment.

```
\/\*[\s\S]*?\*\/
```

Token-by-token breakdown

Token	Meaning
\/	literal "/"
*	literal "*"
[\s\S]*?	zero or more (lazy — as few as possible): any of: whitespace, non-whitespace
*	literal "*"
\/	literal "/"

Quick usage in 7 languages

JavaScript	<pre>const re = /*[\s\S]*?*\/; re.test(value);</pre>
Python	<pre>import re re.match(r"\/*[\s\S]*?*\/", value)</pre>
Java	<pre>Pattern.compile("\/*[\s\S]*?*\/").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"\/*[\s\S]*?*\/");</pre>
Go	<pre>regexp.MustCompile(`\/\*[\s\S]*?\*\/`).MatchString(value)</pre>
Ruby	<pre>\/*[\s\S]*?*\/.match?(value)</pre>
PHP	<pre>preg_match('~\/*[\s\S]*?*\/~', \$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 block comment (`/ * */`) against a source of truth (database, API, or checksum) where it matters.