

Hashtag

Social media hashtag — # followed by letters/digits/underscore.

```
#[A-Za-z][A-Za-z0-9_]*
```

Token-by-token breakdown

Token	Meaning
#	literal text "#"
[A-Za-z]	any of: uppercase letters, lowercase letters
[A-Za-z0-9_]*	zero or more: any of: uppercase letters, lowercase letters, digits, "_"

Quick usage in 7 languages

JavaScript	<pre>const re = /[A-Za-z][A-Za-z0-9_]*/; re.test(value);</pre>
Python	<pre>import re re.match(r"#[A-Za-z][A-Za-z0-9_]*", value)</pre>
Java	<pre>Pattern.compile("#[A-Za-z][A-Za-z0-9_]*").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"#[A-Za-z][A-Za-z0-9_]*");</pre>
Go	<pre>regexp.MustCompile(`#[A-Za-z][A-Za-z0-9_]*`).MatchString(value)</pre>
Ruby	<pre>/#[A-Za-z][A-Za-z0-9_]*/.match?(value)</pre>
PHP	<pre>preg_match('~#[A-Za-z][A-Za-z0-9_]*~', \$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.
- **ASCII letters only.** [a-zA-Z] excludes accented and non-Latin letters (é, ü, ß, ñ, and non-Latin scripts). For international input use Unicode properties like \p{L} with the u flag.
- **Validate beyond format.** Matching the format doesn't guarantee the value is real. Confirm the hashtag against a source of truth (database, API, or checksum) where it matters.