

Twitter/X handle

Twitter @handle — 1-15 alphanumeric/underscore.

```
@([A-Za-z0-9_]{1,15})\b
```

Token-by-token breakdown

Token	Meaning
@	literal text "@"
(	start of a capturing group
[A-Za-z0-9_]{1,15}	between 1 and 15 times: any of: uppercase letters, lowercase letters, digits, "_"
)	end of group
\b	word boundary

Quick usage in 7 languages

JavaScript	<pre>const re = /@([A-Za-z0-9_]{1,15})\b/; re.test(value);</pre>
Python	<pre>import re re.match(r"@([A-Za-z0-9_]{1,15})\b", value)</pre>
Java	<pre>Pattern.compile("@([A-Za-z0-9_]{1,15})\b").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, "@([A-Za-z0-9_]{1,15})\b");</pre>
Go	<pre>regexp.MustCompile("@([A-Za-z0-9_]{1,15})\b").MatchString(value)</pre>
Ruby	<pre>/@([A-Za-z0-9_]{1,15})\b/.match?(value)</pre>
PHP	<pre>preg_match('~@([A-Za-z0-9_]{1,15})\b~', \$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.
- **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 twitter/x handle against a source of truth (database, API, or checksum) where it matters.