

# @Mention

@-mention in social media or chat — @ followed by username.

```
@[A-Za-z][A-Za-z0-9_]{1,29}
```

## Token-by-token breakdown

| Token              | Meaning                                                                           |
|--------------------|-----------------------------------------------------------------------------------|
| @                  | literal text "@"                                                                  |
| [A-Za-z]           | any of: uppercase letters, lowercase letters                                      |
| [A-Za-z0-9_]{1,29} | between 1 and 29 times: any of: uppercase letters, lowercase letters, digits, "_" |

## Quick usage in 7 languages

**JavaScript** `const re = /[A-Za-z][A-Za-z0-9_]{1,29}/;  
re.test(value);`

**Python** `import re  
re.match(r"@[A-Za-z][A-Za-z0-9_]{1,29}", value)`

**Java** `Pattern.compile("@[A-Za-z][A-Za-z0-9_]{1,29}").matcher(value).matches();`

**C# / .NET** `Regex.IsMatch(value, @"@[A-Za-z][A-Za-z0-9_]{1,29}");`

**Go** `regexp.MustCompile(`@[A-Za-z][A-Za-z0-9_]{1,29}`).MatchString(value)`

**Ruby** `/@[A-Za-z][A-Za-z0-9_]{1,29}/.match?(value)`

**PHP** `preg_match('~@[A-Za-z][A-Za-z0-9_]{1,29}~', $value);`

## 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 @mention against a source of truth (database, API, or checksum) where it matters.