

Full name

Full name with Unicode letters, apostrophes, and hyphens.

```
^[A-Za-z\u00C0-\u017F][A-Za-z\u00C0-\u017F'\- ]+$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
[A-Za-z\u00C0-\u017F]	any of: uppercase letters, lowercase letters, "u", "0", "C", "O", "1", "7", "F"
[A-Za-z\u00C0-\u017F'\-]+	one or more: any of: uppercase letters, lowercase letters, "u", "0", "C", "O", "1", "7", "F", " ", "-", "'", " "
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript	<pre>const re = /^[A-Za-z\u00C0-\u017F][A-Za-z\u00C0-\u017F'\-]+\$/; re.test(value);</pre>
Python	<pre>import re re.match(r"^[A-Za-z\u00C0-\u017F][A-Za-z\u00C0-\u017F'\-]+\$", value)</pre>
Java	<pre>Pattern.compile("^[A-Za-z\u00C0-\u017F][A-Za-z\u00C0-\u017F'\-] +\$").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"^[A-Za-z\u00C0-\u017F][A-Za-z\u00C0-\u017F'\-]+\$");</pre>
Go	<pre>regexp.MustCompile(`^[A-Za-z\u00C0-\u017F][A-Za-z\u00C0-\u017F'\- ] +\$`).MatchString(value)</pre>
Ruby	<pre>/^[A-Za-z\u00C0-\u017F][A-Za-z\u00C0-\u017F'\-]+\$/ .match?(value)</pre>
PHP	<pre>preg_match('~^[A-Za-z\u00C0-\u017F][A-Za-z\u00C0-\u017F'\-]+\$~', \$value);</pre>

Common pitfalls

- **Anchored to the whole string.** This pattern uses ^ and \$, so it requires the entire input to match. To find it inside a longer text, drop the anchors and use the global (g) flag.
- **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 full name against a source of truth (database, API, or checksum) where it matters.