

Visa card

Visa credit/debit card.

```
^4\d{12}(\d{3})?$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
4	literal text "4"
\d{12}	exactly 12 times: digit (0–9)
(	start of a capturing group
\d{3}	exactly 3 times: digit (0–9)
)?	end of group, optional (zero or one)
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript	<pre>const re = /^4\d{12}(\d{3})?\$/; re.test(value);</pre>
Python	<pre>import re re.match(r"^4\d{12}(\d{3})?\$", value)</pre>
Java	<pre>Pattern.compile("^4\\d{12}(\\d{3})?\$").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"^4\d{12}(\d{3})?\$");</pre>
Go	<pre>regexp.MustCompile(`^4\d{12}(\d{3})?\$`).MatchString(value)</pre>
Ruby	<pre>/^4\d{12}(\d{3})?\$/ .match?(value)</pre>
PHP	<pre>preg_match('~^4\d{12}(\d{3})?\$~', \$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.
- **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 visa card against a source of truth (database, API, or checksum) where it matters.