

German phone (+49)

German phone with +49 country code.

```
^\+49[\s\-]?\d{2,4}[\s\-]?\d{4,11}$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
\+	literal "+"
49	literal text "49"
[\s\-]?	optional (zero or one): any of: whitespace, "-"
\d{2,4}	between 2 and 4 times: digit (0–9)
[\s\-]?	optional (zero or one): any of: whitespace, "-"
\d{4,11}	between 4 and 11 times: digit (0–9)
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript	<pre>const re = /\+49[\s\-]?\d{2,4}[\s\-]?\d{4,11}\$/; re.test(value);</pre>
Python	<pre>import re re.match(r"\+49[\s\-]?\d{2,4}[\s\-]?\d{4,11}\$", value)</pre>
Java	<pre>Pattern.compile("^\\+49[\\s\\-]?\\d{2,4}[\\s\\-]?\\d{4,11}\$").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"^\\+49[\\s\\-]?\\d{2,4}[\\s\\-]?\\d{4,11}\$");</pre>
Go	<pre>regexp.MustCompile(`^\\+49[\\s\\-]?\\d{2,4}[\\s\\-]?\\d{4,11}\$`).MatchString(value)</pre>
Ruby	<pre>/^\\+49[\\s\\-]?\\d{2,4}[\\s\\-]?\\d{4,11}\$/ .match?(value)</pre>
PHP	<pre>preg_match('~^\\+49[\\s\\-]?\\d{2,4}[\\s\\-]?\\d{4,11}\$~', \$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 german phone (+49) against a source of truth (database, API, or checksum) where it matters.