

Russian phone (+7)

Russian phone with +7 prefix.

```
^(\\+7|8)[\\s\\-]?\\((?\\d{3})\\)?[\\s\\-]?\\d{3}[\\s\\-]?\\d{2}[\\s\\-]?\\d{2}$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
(	start of a capturing group
\\+	literal “+”
7	literal text “7”
	OR — try the alternative
8	literal text “8”
)	end of group
[\\s\\-]?	optional (zero or one): any of: whitespace, “-”
\\(?	optional (zero or one): literal “(”
\\d{3}	exactly 3 times: digit (0–9)
\\)?	optional (zero or one): literal “)”
[\\s\\-]?	optional (zero or one): any of: whitespace, “-”
\\d{3}	exactly 3 times: digit (0–9)
[\\s\\-]?	optional (zero or one): any of: whitespace, “-”
\\d{2}	exactly 2 times: digit (0–9)
[\\s\\-]?	optional (zero or one): any of: whitespace, “-”
\\d{2}	exactly 2 times: digit (0–9)
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript	<pre>const re = /^(\\+7 8)[\\s\\-]?\\((?\\d{3})\\)?[\\s\\-]?\\d{3}[\\s\\-]?\\d{2}[\\s\\-]?\\d{2}\$/; re.test(value);</pre>
Python	<pre>import re re.match(r"^(\\+7 8)[\\s\\-]?\\((?\\d{3})\\)?[\\s\\-]?\\d{3}[\\s\\-]?\\d{2}[\\s\\-]?\\d{2}\$", value)</pre>
Java	<pre>Pattern.compile("(\\+7 8)[\\s\\-]?\\((?\\d{3})\\)?[\\s\\-]?\\d{3}[\\s\\-]?\\d{2}[\\s\\-]?\\d{2}\$").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"^(\\+7 8)[\\s\\-]?\\((?\\d{3})\\)?[\\s\\-]?\\d{3}[\\s\\-]?\\d{2}[\\s\\-]?\\d{2}\$") ;</pre>
Go	<pre>regexp.MustCompile(`^(\\+7 8)[\\s\\-]?\\((?\\d{3})\\)?[\\s\\-]?\\d{3}[\\s\\-]?\\d{2}[\\s\\-]?\\d{2}\$`).MatchString(value)</pre>
Ruby	<pre>/^(\\+7 8)[\\s\\-]?\\((?\\d{3})\\)?[\\s\\-]?\\d{3}[\\s\\-]?\\d{2}[\\s\\-]?\\d{2}\$/ .match?(value)</pre>

PHP

```
preg_match('~^(\\+7|8)[\\s\\-]?\\(?\\d{3}\\)?[\\s\\-]?\\d{3}[\\s\\-]?\\d{2}[\\s\\-]?\\d{2}$~',  
$value);
```

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 russian phone (+7) against a source of truth (database, API, or checksum) where it matters.