

Japanese phone

Japanese landline or mobile (+81).

```
^(\\+81[\\s\\-]?|0)\\d{1,4}[\\s\\-]?\\d{1,4}[\\s\\-]?\\d{3,4}$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
(	start of a capturing group
\\+	literal “+”
81	literal text “81”
[\\s\\-]?	optional (zero or one): any of: whitespace, “-”
	OR — try the alternative
0	literal text “0”
)	end of group
\\d{1,4}	between 1 and 4 times: digit (0–9)
[\\s\\-]?	optional (zero or one): any of: whitespace, “-”
\\d{1,4}	between 1 and 4 times: digit (0–9)
[\\s\\-]?	optional (zero or one): any of: whitespace, “-”
\\d{3,4}	between 3 and 4 times: digit (0–9)
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript	<pre>const re = /^(\\+81[\\s\\-]? 0)\\d{1,4}[\\s\\-]?\\d{1,4}[\\s\\-]?\\d{3,4}\$/; re.test(value);</pre>
Python	<pre>import re re.match(r"^(\\+81[\\s\\-]? 0)\\d{1,4}[\\s\\-]?\\d{1,4}[\\s\\-]?\\d{3,4}\$", value)</pre>
Java	<pre>Pattern.compile("(\\+81[\\s\\-]? 0)\\d{1,4}[\\s\\-]?\\d{1,4}[\\s\\-]?\\d{3,4}\$").match er(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"^(\\+81[\\s\\-]? 0)\\d{1,4}[\\s\\-]?\\d{1,4}[\\s\\-]?\\d{3,4}\$");</pre>
Go	<pre>regexp.MustCompile(`^(\\+81[\\s\\-]? 0)\\d{1,4}[\\s\\-]?\\d{1,4}[\\s\\-]?\\d{3,4}\$`).MatchString(value)</pre>
Ruby	<pre>/^(\\+81[\\s\\-]? 0)\\d{1,4}[\\s\\-]?\\d{1,4}[\\s\\-]?\\d{3,4}\$/ .match?(value)</pre>
PHP	<pre>preg_match('~^(\\+81[\\s\\-]? 0)\\d{1,4}[\\s\\-]?\\d{1,4}[\\s\\-]?\\d{3,4}\$~', \$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 japanese phone against a source of truth (database, API, or checksum) where it matters.