

UAE phone (+971)

UAE phone with +971 country code.

```
^\+971[\s\-]?\d{1,2}[\s\-]?\d{3}[\s\-]?\d{4}$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
\+	literal "+"
971	literal text "971"
[\s\-]?	optional (zero or one): any of: whitespace, "-"
\d{1,2}	between 1 and 2 times: digit (0–9)
[\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{4}	exactly 4 times: digit (0–9)
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript	<pre>const re = /\^+971[\s\-]?\d{1,2}[\s\-]?\d{3}[\s\-]?\d{4}\$/; re.test(value);</pre>
Python	<pre>import re re.match(r"^\+971[\s\-]?\d{1,2}[\s\-]?\d{3}[\s\-]?\d{4}\$", value)</pre>
Java	<pre>Pattern.compile("^\+971[\s\\-]?\d{1,2}[\s\\-]?\d{3}[\s\\-]?\d{4}\$").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"^\+971[\s\-]?\d{1,2}[\s\-]?\d{3}[\s\-]?\d{4}\$");</pre>
Go	<pre>regexp.MustCompile(`^\+971[\s\-]?\d{1,2}[\s\-]?\d{3}[\s\-]?\d{4}\$`).MatchString(value)</pre>
Ruby	<pre>/^\+971[\s\-]?\d{1,2}[\s\-]?\d{3}[\s\-]?\d{4}\$/ .match?(value)</pre>
PHP	<pre>preg_match('~^\+971[\s\-]?\d{1,2}[\s\-]?\d{3}[\s\-]?\d{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 uae phone (+971) against a source of truth (database, API, or checksum) where it matters.