

UK mobile number

UK mobile in 07XXX or +44 7XXX format.

```
^(\\+44\\s?|0)7\\d{3}\\s?\\d{6}$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
(	start of a capturing group
\\+	literal “+”
44	literal text “44”
\\s?	optional (zero or one): whitespace
	OR — try the alternative
0	literal text “0”
)	end of group
7	literal text “7”
\\d{3}	exactly 3 times: digit (0–9)
\\s?	optional (zero or one): whitespace
\\d{6}	exactly 6 times: digit (0–9)
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript	<pre>const re = /^(\\+44\\s? 0)7\\d{3}\\s?\\d{6}\$/; re.test(value);</pre>
Python	<pre>import re re.match(r"^(\\+44\\s? 0)7\\d{3}\\s?\\d{6}\$", value)</pre>
Java	<pre>Pattern.compile("(\\+44\\s? 0)7\\d{3}\\s?\\d{6}\$").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"^(\\+44\\s? 0)7\\d{3}\\s?\\d{6}\$");</pre>
Go	<pre>regexp.MustCompile(`^(\\+44\\s? 0)7\\d{3}\\s?\\d{6}\$`).MatchString(value)</pre>
Ruby	<pre>/^(\\+44\\s? 0)7\\d{3}\\s?\\d{6}\$/ .match?(value)</pre>
PHP	<pre>preg_match('~^(\\+44\\s? 0)7\\d{3}\\s?\\d{6}\$~', \$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 uk mobile number against a source of truth (database, API, or checksum) where it matters.