

Phone with extension

Phone number followed by ext./x and digits.

```
^.+(\s+(ext\.?|x|extension)\s*\d+)$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
.	one or more of any character
(	start of a capturing group
\s+	one or more: whitespace
(	start of a capturing group
ext	literal text "ext"
\.?	optional (zero or one): literal "."
	OR — try the alternative
x	literal text "x"
	OR — try the alternative
extension	literal text "extension"
)	end of group
\s*	zero or more: whitespace
\d+	one or more: digit (0–9)
)	end of group
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript	<pre>const re = /^.+(\s+(ext\.? x extension)\s*\d+)\$/; re.test(value);</pre>
Python	<pre>import re re.match(r"^.+(\s+(ext\.? x extension)\s*\d+)\$", value)</pre>
Java	<pre>Pattern.compile("^.+(\s+(ext\.? x extension)\s*\d+)\$").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"^.+(\s+(ext\.? x extension)\s*\d+)\$");</pre>
Go	<pre>regexp.MustCompile(`^.+(\s+(ext\.? x extension)\s*\d+)\$`).MatchString(value)</pre>
Ruby	<pre>/^.+(\s+(ext\.? x extension)\s*\d+)\$/ =~ value</pre>
PHP	<pre>preg_match('~^.+(\s+(ext\.? x extension)\s*\d+)\$~', \$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.

- **Greedy “.”**. A greedy `.` / `.+` can match more than intended. Use a lazy version `(.*?)` or a negated class `[^...]` to stop at the right place.
- **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 phone with extension against a source of truth (database, API, or checksum) where it matters.