

12-hour time (with AM/PM)

12-hour time with AM/PM marker.

```
^(0?[1-9]|1[0-2]):[0-5]\d\s?(AM|PM|am|pm)$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
(	start of a capturing group
0?	optional (zero or one): the literal "0"
[1-9]	any of: 1–9
	OR — try the alternative
1	literal text "1"
[0-2]	any of: 0–2
)	end of group
:	literal text ":"
[0-5]	any of: 0–5
\d	any digit (0–9)
\s?	optional (zero or one): whitespace
(	start of a capturing group
AM	literal text "AM"
	OR — try the alternative
PM	literal text "PM"
	OR — try the alternative
am	literal text "am"
	OR — try the alternative
pm	literal text "pm"
)	end of group
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript	<pre>const re = /^(0?[1-9] 1[0-2]):[0-5]\d\s?(AM PM am pm)\$/; re.test(value);</pre>
Python	<pre>import re re.match(r"^(0?[1-9] 1[0-2]):[0-5]\d\s?(AM PM am pm)\$", value)</pre>
Java	<pre>Pattern.compile("^(0?[1-9] 1[0-2]):[0-5]\\d\\s?(AM PM am pm)\$").matcher(value).matches() ;</pre>

C# / .NET	<code>Regex.IsMatch(value, @"^(0?[1-9] 1[0-2]):[0-5]\d\s?(AM PM am pm)\$");</code>
Go	<code>regexp.MustCompile(`^(0?[1-9] 1[0-2]):[0-5]\d\s?(AM PM am pm)\$`).MatchString(value)</code>
Ruby	<code>/^(0?[1-9] 1[0-2]):[0-5]\d\s?(AM PM am pm)\$/ .match?(value)</code>
PHP	<code>preg_match('~^(0?[1-9] 1[0-2]):[0-5]\d\s?(AM PM am pm)\$~', \$value);</code>

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 12-hour time (with `am/pm`) against a source of truth (database, API, or checksum) where it matters.