

Percentage (0-100%)

Percentage value with optional decimals and % sign.

```
^(100(\.0+)?|\d{1,2}(\.\d+)?)\s*%?$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
(	start of a capturing group
100	literal text "100"
(	start of a capturing group
\.	literal "."
0+	one or more: the literal "0"
)?	end of group, optional (zero or one)
	OR — try the alternative
\d{1,2}	between 1 and 2 times: digit (0–9)
(	start of a capturing group
\.	literal "."
\d+	one or more: digit (0–9)
)?	end of group, optional (zero or one)
)	end of group
\s*	zero or more: whitespace
%?	optional (zero or one): the literal "%"
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript	<pre>const re = /^(100(\.0+)? \d{1,2}(\.\d+)?)\s*%?\$/; re.test(value);</pre>
Python	<pre>import re re.match(r"^(100(\.0+)? \d{1,2}(\.\d+)?)\s*%?\$", value)</pre>
Java	<pre>Pattern.compile("^(100(\\.0+)? \\d{1,2}(\\.\\d+)?)\\s*%?\$").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"^(100(\.0+)? \d{1,2}(\.\d+)?)\s*%?\$");</pre>
Go	<pre>regexp.MustCompile(`^(100(\.0+)? \d{1,2}(\.\d+)?)\s*%?\$`).MatchString(value)</pre>
Ruby	<pre>/^(100(\.0+)? \d{1,2}(\.\d+)?)\s*%?\$/ .match?(value)</pre>
PHP	<pre>preg_match('~^(100(\.0+)? \d{1,2}(\.\d+)?)\s*%?\$~', \$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 percentage (0-100%) against a source of truth (database, API, or checksum) where it matters.