

ISBN-13

13-digit International Standard Book Number.

```
^(?=(?:\d*\d){13})97[89][\d-]+$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
(?=	start of a positive lookahead — followed by
(?:	start of a non-capturing group
\d*	zero or more: non-digit
\d	any digit (0–9)
{13}	end of group, exactly 13 times
)	end of group
97	literal text "97"
[89]	any of: "8", "9"
[\d-]+	one or more: any of: digits (0–9), "-"
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript	<pre>const re = /^(?=(?:\d*\d){13})97[89][\d-]+\$/; re.test(value);</pre>
Python	<pre>import re re.match(r"^(?=(?:\d*\d){13})97[89][\d-]+\$", value)</pre>
Java	<pre>Pattern.compile("^(?=(?:\d*\d){13})97[89][\d-]+\$").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"^(?=(?:\d*\d){13})97[89][\d-]+\$");</pre>
Go	<pre>regexp.MustCompile(`^(?=(?:\d*\d){13})97[89][\d-]+\$`).MatchString(value)</pre>
Ruby	<pre>/^(?=(?:\d*\d){13})97[89][\d-]+\$/ .match?(value)</pre>
PHP	<pre>preg_match('~^(?=(?:\d*\d){13})97[89][\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.
- **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 isbn-13 against a source of truth (database, API, or checksum) where it matters.