

ISO 8601 date (YYYY-MM-DD)

The international date format.

```
^\d{4}-(0[1-9]|1[0-2])-(0[1-9]|[12]\d|3[01])$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
\d{4}	exactly 4 times: digit (0–9)
-	literal text “-”
(	start of a capturing group
0	literal text “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 “-”
(	start of a capturing group
0	literal text “0”
[1-9]	any of: 1–9
	OR — try the alternative
[12]	any of: “1”, “2”
\d	any digit (0–9)
	OR — try the alternative
3	literal text “3”
[01]	any of: “0”, “1”
)	end of group
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript	<pre>const re = /^\d{4}-(0[1-9] 1[0-2])-(0[1-9] [12]\d 3[01])\$/; re.test(value);</pre>
Python	<pre>import re re.match(r"^\d{4}-(0[1-9] 1[0-2])-(0[1-9] [12]\d 3[01])\$", value)</pre>
Java	<pre>Pattern.compile("^\d{4}-(0[1-9] 1[0-2])-(0[1-9] [12]\d 3[01])\$").matcher(value).match es();</pre>

C# / .NET	<code>Regex.IsMatch(value, @"^d{4}-(0[1-9] 1[0-2])-(0[1-9] [12]\d 3[01])\$");</code>
Go	<code>regexp.MustCompile(`^d{4}-(0[1-9] 1[0-2])-(0[1-9] [12]\d 3[01])\$`).MatchString(value)</code>
Ruby	<code>/^d{4}-(0[1-9] 1[0-2])-(0[1-9] [12]\d 3[01])\$/ .match?(value)</code>
PHP	<code>preg_match('~^d{4}-(0[1-9] 1[0-2])-(0[1-9] [12]\d 3[01])\$~', \$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 iso 8601 date (yyyy-mm-dd) against a source of truth (database, API, or checksum) where it matters.