ISO 8601 week date

YYYY-Www-D format ISO week date.

```
^\d{4}-W(0[1-9]|[1-4]\d|5[0-3])-[1-7]$
```

Token-by-token breakdown

Token	Meaning
`^`	start of string (or line in multiline mode)
`\d{4}`	exactly 4 times: digit (0–9)
`-W`	literal text “-W”
`(`	start of a capturing group
`0`	literal text “0”
`[1-9]`	any of: 1–9
`|`	OR — try the alternative
`[1-4]`	any of: 1–4
`\d`	any digit (0–9)
`|`	OR — try the alternative
`5`	literal text “5”
`[0-3]`	any of: 0–3
`)`	end of group
`-`	literal text “-”
`[1-7]`	any of: 1–7
`$`	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript

```
const re = /^\d{4}-W(0[1-9]|[1-4]\d|5[0-3])-[1-7]$/;
re.test(value);
```

Python

```
import re
re.match(r"^\d{4}-W(0[1-9]|[1-4]\d|5[0-3])-[1-7]$", value)
```

Java

```
Pattern.compile("^\\d{4}-W(0[1-9]|[1-4]\\d|5[0-3])-[1-7]$").matcher(value).matches();
```

C# / .NET

```
Regex.IsMatch(value, @"^\d{4}-W(0[1-9]|[1-4]\d|5[0-3])-[1-7]$");
```

Go

```
regexp.MustCompile(`^\d{4}-W(0[1-9]|[1-4]\d|5[0-3])-[1-7]$`).MatchString(value)
```

Ruby

```
/^\d{4}-W(0[1-9]|[1-4]\d|5[0-3])-[1-7]$/.match?(value)
```

PHP

```
preg_match('~^\d{4}-W(0[1-9]|[1-4]\d|5[0-3])-[1-7]$~', $value);
```

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 week date against a source of truth (database, API, or checksum) where it matters.