Cron expression (5-field)

Standard 5-field cron: minute hour day month weekday.

```
^(\*|[0-5]?\d|\*\/\d+|[0-5]?\d-[0-5]?\d|[0-5]?\d(,[0-5]?\d)+)\s+(\*|1?
\d|2[0-3]|\*\/\d+|\d+-\d+|\d+(,\d+)+)\s+(\*|[1-9]|[12]\d|3[01]|\*\/\d
+|\d+-\d+|\?|L|W)\s+(\*|[1-9]|1[0-2]|\*\/\d+|JAN|FEB|MAR|APR|MAY|JUN|
JUL|AUG|SEP|OCT|NOV|DEC)\s+(\*|[0-6]|\*\/\d+|MON|TUE|WED|THU|FRI|SAT|
SUN|\?|L)$
```

Token-by-token breakdown

Token	Meaning
`^`	start of string (or line in multiline mode)
`(`	start of a capturing group
`\*`	literal “*”
`|`	OR — try the alternative
`[0-5]?`	optional (zero or one): any of: 0–5
`\d`	any digit (0–9)
`|`	OR — try the alternative
`\*`	literal “*”
`\/`	literal “/”
`\d+`	one or more: digit (0–9)
`|`	OR — try the alternative
`[0-5]?`	optional (zero or one): any of: 0–5
`\d`	any digit (0–9)
`-`	literal text “-”
`[0-5]?`	optional (zero or one): any of: 0–5
`\d`	any digit (0–9)
`|`	OR — try the alternative
`[0-5]?`	optional (zero or one): any of: 0–5
`\d`	any digit (0–9)
`(`	start of a capturing group
`,`	literal text “,”
`[0-5]?`	optional (zero or one): any of: 0–5
`\d`	any digit (0–9)
`)+`	end of group, one or more
`)`	end of group

© 2026 regexguide.com · a human-curated regex explainer & pattern library
Disclaimer: provided as-is, without warranty — test against your own data before production use.

Token	Meaning
`\s+`	one or more: whitespace
`(`	start of a capturing group
`\*`	literal “*”
`|`	OR — try the alternative
`1?`	optional (zero or one): the literal “1”
`\d`	any digit (0–9)
`|`	OR — try the alternative
`2`	literal text “2”
`[0-3]`	any of: 0–3
`|`	OR — try the alternative
`\*`	literal “*”
`\/`	literal “/”
`\d+`	one or more: digit (0–9)
`|`	OR — try the alternative
`\d+`	one or more: digit (0–9)
`-`	literal text “-”
`\d+`	one or more: digit (0–9)
`|`	OR — try the alternative
`\d+`	one or more: digit (0–9)
`(`	start of a capturing group
`,`	literal text “,”
`\d+`	one or more: digit (0–9)
`)+`	end of group, one or more
`)`	end of group
`\s+`	one or more: whitespace
`(`	start of a capturing group
`\*`	literal “*”
`|`	OR — try the alternative
`[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”
`|`	OR — try the alternative

© 2026 regexguide.com · a human-curated regex explainer & pattern library

Disclaimer: provided as-is, without warranty — test against your own data before production use.

Token	Meaning
`\*`	literal “*”
`\/`	literal “/”
`\d+`	one or more: digit (0–9)
`|`	OR — try the alternative
`\d+`	one or more: digit (0–9)
`-`	literal text “-”
`\d+`	one or more: digit (0–9)
`|`	OR — try the alternative
`\?`	literal “?”
`|`	OR — try the alternative
`L`	literal text “L”
`|`	OR — try the alternative
`W`	literal text “W”
`)`	end of group
`\s+`	one or more: whitespace
`(`	start of a capturing group
`\*`	literal “*”
`|`	OR — try the alternative
`[1-9]`	any of: 1–9
`|`	OR — try the alternative
`1`	literal text “1”
`[0-2]`	any of: 0–2
`|`	OR — try the alternative
`\*`	literal “*”
`\/`	literal “/”
`\d+`	one or more: digit (0–9)
`|`	OR — try the alternative
`JAN`	literal text “JAN”
`|`	OR — try the alternative
`FEB`	literal text “FEB”
`|`	OR — try the alternative
`MAR`	literal text “MAR”
`|`	OR — try the alternative
`APR`	literal text “APR”
`|`	OR — try the alternative
`MAY`	literal text “MAY”

© 2026 regexguide.com · a human-curated regex explainer & pattern library

Disclaimer: provided as-is, without warranty — test against your own data before production use.

Token	Meaning
`|`	OR — try the alternative
`JUN`	literal text “JUN”
`|`	OR — try the alternative
`JUL`	literal text “JUL”
`|`	OR — try the alternative
`AUG`	literal text “AUG”
`|`	OR — try the alternative
`SEP`	literal text “SEP”
`|`	OR — try the alternative
`OCT`	literal text “OCT”
`|`	OR — try the alternative
`NOV`	literal text “NOV”
`|`	OR — try the alternative
`DEC`	literal text “DEC”
`)`	end of group
`\s+`	one or more: whitespace
`(`	start of a capturing group
`\*`	literal “*”
`|`	OR — try the alternative
`[0-6]`	any of: 0–6
`|`	OR — try the alternative
`\*`	literal “*”
`\/`	literal “/”
`\d+`	one or more: digit (0–9)
`|`	OR — try the alternative
`MON`	literal text “MON”
`|`	OR — try the alternative
`TUE`	literal text “TUE”
`|`	OR — try the alternative
`WED`	literal text “WED”
`|`	OR — try the alternative
`THU`	literal text “THU”
`|`	OR — try the alternative
`FRI`	literal text “FRI”
`|`	OR — try the alternative
`SAT`	literal text “SAT”

© 2026 regexguide.com · a human-curated regex explainer & pattern library

Disclaimer: provided as-is, without warranty — test against your own data before production use.

Token	Meaning
`|`	OR — try the alternative
`SUN`	literal text “SUN”
`|`	OR — try the alternative
`\?`	literal “?”
`|`	OR — try the alternative
`L`	literal text “L”
`)`	end of group
`$`	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript

```
const re = /^(\*|[0-5]?\d|\*\/\d+|[0-5]?\d-[0-5]?\d|[0-5]?\d(,[0-5]?\d)+)\s+(\*|1?\d|2[0-3]|\*\/\d+|\d+-\d+|\d+(,\d+)+)\s+(\*|[1-9]|[12]\d|3[01]|\*\/\d+|\d+-\d+|\?|L|W)\s+(\*|[1-9]|1[0-2]|\*\/\d+|JAN|FEB|MAR|APR|MAY|JUN|JUL|AUG|SEP|OCT|NOV|DEC)\s+(\*|[0-6]|\*\/\d+|MON|TUE|WED|THU|FRI|SAT|SUN|\?|L)$/;
re.test(value);
```

Python

```
import re
re.match(r"^(\*|[0-5]?\d|\*\/\d+|[0-5]?\d-[0-5]?\d|[0-5]?\d(,[0-5]?\d)+)\s+(\*|1?\d|2[0-3]|\*\/\d+|\d+-\d+|\d+(,\d+)+)\s+(\*|[1-9]|[12]\d|3[01]|\*\/\d+|\d+-\d+|\?|L|W)\s+(\*|[1-9]|1[0-2]|\*\/\d+|JAN|FEB|MAR|APR|MAY|JUN|JUL|AUG|SEP|OCT|NOV|DEC)\s+(\*|[0-6]|\*\/\d+|MON|TUE|WED|THU|FRI|SAT|SUN|\?|L)$", value)
```

Java

```
Pattern.compile("^(\\*|[0-5]?\\d|\\*\\/\\d+|[0-5]?\\d-[0-5]?\\d|[0-5]?\\d(,[0-5]?\\d)+)\\s+(\\*|1?\\d|2[0-3]|\\*\\/\\d+|\\d+-\\d+|\\d+(,\\d+)+)\\s+(\\*|[1-9]|[12]\\d|3[01]|\\*\\/\\d+|\\d+-\\d+|\\?|L|W)\\s+(\\*|[1-9]|1[0-2]|\\*\\/\\d+|JAN|FEB|MAR|APR|MAY|JUN|JUL|AUG|SEP|OCT|NOV|DEC)\\s+(\\*|[0-6]|\\*\\/\\d+|MON|TUE|WED|THU|FRI|SAT|SUN|\\?|L)$").matcher(value).matches();
```

C# / .NET

```
Regex.IsMatch(value, @"^(\*|[0-5]?\d|\*\/\d+|[0-5]?\d-[0-5]?\d|[0-5]?\d(,[0-5]?\d)+)\s+(\*|1?\d|2[0-3]|\*\/\d+|\d+-\d+|\d+(,\d+)+)\s+(\*|[1-9]|[12]\d|3[01]|\*\/\d+|\d+-\d+|\?|L|W)\s+(\*|[1-9]|1[0-2]|\*\/\d+|JAN|FEB|MAR|APR|MAY|JUN|JUL|AUG|SEP|OCT|NOV|DEC)\s+(\*|[0-6]|\*\/\d+|MON|TUE|WED|THU|FRI|SAT|SUN|\?|L)$");
```

Go

```
regexp.MustCompile(`^(\*|[0-5]?\d|\*\/\d+|[0-5]?\d-[0-5]?\d|[0-5]?\d(,[0-5]?\d)+)\s+(\*|1?\d|2[0-3]|\*\/\d+|\d+-\d+|\d+(,\d+)+)\s+(\*|[1-9]|[12]\d|3[01]|\*\/\d+|\d+-\d+|\?|L|W)\s+(\*|[1-9]|1[0-2]|\*\/\d+|JAN|FEB|MAR|APR|MAY|JUN|JUL|AUG|SEP|OCT|NOV|DEC)\s+(\*|[0-6]|\*\/\d+|MON|TUE|WED|THU|FRI|SAT|SUN|\?|L)$`).MatchString(value)
```

Ruby

```
/^(\*|[0-5]?\d|\*\/\d+|[0-5]?\d-[0-5]?\d|[0-5]?\d(,[0-5]?\d)+)\s+(\*|1?\d|2[0-3]|\*\/\d+|\d+-\d+|\d+(,\d+)+)\s+(\*|[1-9]|[12]\d|3[01]|\*\/\d+|\d+-\d+|\?|L|W)\s+(\*|[1-9]|1[0-2]|\*\/\d+|JAN|FEB|MAR|APR|MAY|JUN|JUL|AUG|SEP|OCT|NOV|DEC)\s+(\*|[0-6]|\*\/\d+|MON|TUE|WED|THU|FRI|SAT|SUN|\?|L)$/.match?(value)
```

PHP

```
preg_match('~^(\*|[0-5]?\d|\*\/\d+|[0-5]?\d-[0-5]?\d|[0-5]?\d(,[0-5]?\d)+)\s+(\*|1?\d|2[0-3]|\*\/\d+|\d+-\d+|\d+(,\d+)+)\s+(\*|[1-9]|[12]\d|3[01]|\*\/\d+|\d+-\d+|\?|L|W)\s+(\*|[1-9]|1[0-2]|\*\/\d+|JAN|FEB|MAR|APR|MAY|JUN|JUL|AUG|SEP|OCT|NOV|DEC)\s+(\*|[0-6]|\*\/\d+|MON|TUE|WED|THU|FRI|SAT|SUN|\?|L)$~', $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.
- **Watch for backtracking.** This pattern nests quantifiers; on adversarial input that can cause exponential backtracking (ReDoS). Test with long non-matching strings, or use an RE2-based engine.
- **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.

© 2026 regexguide.com · a human-curated regex explainer & pattern library

Disclaimer: provided as-is, without warranty — test against your own data before production use.

• **Validate beyond format.** Matching the format doesn't guarantee the value is real. Confirm the cron expression (5-field) against a source of truth (database, API, or checksum) where it matters.

© 2026 regexguide.com · a human-curated regex explainer & pattern library

Disclaimer: provided as-is, without warranty — test against your own data before production use.