

# Positive integer

Non-negative whole number, no leading zeros.

```
^(0|[1-9]\d*)$
```

## Token-by-token breakdown

| Token | Meaning                                     |
|-------|---------------------------------------------|
| ^     | start of string (or line in multiline mode) |
| (     | start of a capturing group                  |
| 0     | literal text "0"                            |
|       | OR — try the alternative                    |
| [1-9] | any of: 1–9                                 |
| \d*   | zero or more: digit (0–9)                   |
| )     | end of group                                |
| \$    | end of string (or line in multiline mode)   |

## Quick usage in 7 languages

|                   |                                                                          |
|-------------------|--------------------------------------------------------------------------|
| <b>JavaScript</b> | <pre>const re = /^(0 [1-9]\d*)\$/;<br/>re.test(value);</pre>             |
| <b>Python</b>     | <pre>import re<br/>re.match(r"^(0 [1-9]\d*)\$", value)</pre>             |
| <b>Java</b>       | <pre>Pattern.compile("^(0 [1-9]\\d*)\$").matcher(value).matches();</pre> |
| <b>C# / .NET</b>  | <pre>Regex.IsMatch(value, @"^(0 [1-9]\d*)\$");</pre>                     |
| <b>Go</b>         | <pre>regexp.MustCompile(`^(0 [1-9]\d*)\$`).MatchString(value)</pre>      |
| <b>Ruby</b>       | <pre>/^(0 [1-9]\d*)\$/ .match?(value)</pre>                              |
| <b>PHP</b>        | <pre>preg_match('~^(0 [1-9]\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 positive integer against a source of truth (database, API, or checksum) where it matters.