

# Indian mobile number

10-digit Indian mobile, starts with 6, 7, 8, or 9.

```
^[6-9]\d{9}$
```

## Token-by-token breakdown

| Token | Meaning                                     |
|-------|---------------------------------------------|
| ^     | start of string (or line in multiline mode) |
| [6-9] | any of: 6–9                                 |
| \d{9} | exactly 9 times: digit (0–9)                |
| \$    | end of string (or line in multiline mode)   |

## Quick usage in 7 languages

|            |                                                                        |
|------------|------------------------------------------------------------------------|
| JavaScript | <pre>const re = /^[6-9]\d{9}\$/;<br/>re.test(value);</pre>             |
| Python     | <pre>import re<br/>re.match(r"^[6-9]\d{9}\$", value)</pre>             |
| Java       | <pre>Pattern.compile("^[6-9]\\d{9}\$").matcher(value).matches();</pre> |
| C# / .NET  | <pre>Regex.IsMatch(value, @"^[6-9]\d{9}\$");</pre>                     |
| Go         | <pre>regexp.MustCompile(`^[6-9]\d{9}\$`).MatchString(value)</pre>      |
| Ruby       | <pre>/^[6-9]\d{9}\$/ .match?(value)</pre>                              |
| PHP        | <pre>preg_match('~^[6-9]\d{9}\$~', \$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 indian mobile number against a source of truth (database, API, or checksum) where it matters.