

IPv4 address

IPv4 address with octet range validation.

```
^((25[0-5]|2[0-4]\d|[01]?[0-9]\d?)\.){3}(25[0-5]|2[0-4]\d|[01]?[0-9]\d?)$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
(	start of a capturing group
(	start of a capturing group
25	literal text "25"
[0-5]	any of: 0–5
	OR — try the alternative
2	literal text "2"
[0-4]	any of: 0–4
\d	any digit (0–9)
	OR — try the alternative
[01]?	optional (zero or one): any of: "0", "1"
\d	any digit (0–9)
\d?	optional (zero or one): digit (0–9)
)	end of group
\.	literal "."
) {3}	end of group, exactly 3 times
(	start of a capturing group
25	literal text "25"
[0-5]	any of: 0–5
	OR — try the alternative
2	literal text "2"
[0-4]	any of: 0–4
\d	any digit (0–9)
	OR — try the alternative
[01]?	optional (zero or one): any of: "0", "1"
\d	any digit (0–9)
\d?	optional (zero or one): digit (0–9)
)	end of group

Token	Meaning
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript	<pre>const re = /^((25[0-5] 2[0-4]\d [01]?\d\d?)\.){3}(25[0-5] 2[0-4]\d [01]?\d\d?)\$/; re.test(value);</pre>
Python	<pre>import re re.match(r"^((25[0-5] 2[0-4]\d [01]?\d\d?)\.){3}(25[0-5] 2[0-4]\d [01]?\d\d?)\$", value)</pre>
Java	<pre>Pattern.compile("^((25[0-5] 2[0-4]\\d [01]?\\d\\d?)\\.){3}(25[0-5] 2[0-4]\\d [01]?\\d\\d?\\d?)\$").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"^((25[0-5] 2[0-4]\d [01]?\d\d?)\.){3}(25[0-5] 2[0-4]\d [01]?\d\d?)");</pre>
Go	<pre>regexp.MustCompile(`^((25[0-5] 2[0-4]\d [01]?\d\d?)\.){3}(25[0-5] 2[0-4]\d [01]?\d\d?)\$`).MatchString(value)</pre>
Ruby	<pre>/^((25[0-5] 2[0-4]\d [01]?\d\d?)\.){3}(25[0-5] 2[0-4]\d [01]?\d\d?)\$/ .match?(value)</pre>
PHP	<pre>preg_match('~^((25[0-5] 2[0-4]\d [01]?\d\d?)\.){3}(25[0-5] 2[0-4]\d [01]?\d\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 ipv4 address against a source of truth (database, API, or checksum) where it matters.