REGEX CHEAT SHEET
regexguide.com/patterns/port-number.html

Port number

TCP/UDP port — 0 to 65535.

```
^(6553[0-5]|655[0-2]\d|65[0-4]\d{2}|6[0-4]\d{3}|[1-5]\d{4}|[1-9]\d{0,3}|0)$
```

Token-by-token breakdown

Token	Meaning
`^`	start of string (or line in multiline mode)
`(`	start of a capturing group
`6553`	literal text “6553”
`[0-5]`	any of: 0–5
`|`	OR — try the alternative
`655`	literal text “655”
`[0-2]`	any of: 0–2
`\d`	any digit (0–9)
`|`	OR — try the alternative
`65`	literal text “65”
`[0-4]`	any of: 0–4
`\d{2}`	exactly 2 times: digit (0–9)
`|`	OR — try the alternative
`6`	literal text “6”
`[0-4]`	any of: 0–4
`\d{3}`	exactly 3 times: digit (0–9)
`|`	OR — try the alternative
`[1-5]`	any of: 1–5
`\d{4}`	exactly 4 times: digit (0–9)
`|`	OR — try the alternative
`[1-9]`	any of: 1–9
`\d{0,3}`	between 0 and 3 times: digit (0–9)
`|`	OR — try the alternative
`0`	literal text “0”
`)`	end of group
`$`	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript	`const re = /^(6553[0-5]|655[0-2]\d|65[0-4]\d{2}|6[0-4]\d{3}|[1-5]\d{4}|[1-9]\d{0,3}|0)$/;` `re.test(value);`
Python	`import re` `re.match(r"^(6553[0-5]|655[0-2]\d|65[0-4]\d{2}|6[0-4]\d{3}|[1-5]\d{4}|[1-9]\d{0,3}|0)$", value)`
Java	`Pattern.compile("^(6553[0-5]|655[0-2]\\d|65[0-4]\\d{2}|6[0-4]\\d{3}|[1-5]\\d{4}|[1-9]\\d{0,3}|0)$").matcher(value).matches();`
C# / .NET	`Regex.IsMatch(value, @"^(6553[0-5]|655[0-2]\d|65[0-4]\d{2}|6[0-4]\d{3}|[1-5]\d{4}|[1-9]\d{0,3}|0)$");`
Go	`regexp.MustCompile(`^(6553[0-5]|655[0-2]\d|65[0-4]\d{2}|6[0-4]\d{3}|[1-5]\d{4}|[1-9]\d{0,3}|0)$`).MatchString(value)`
Ruby	`/^(6553[0-5]|655[0-2]\d|65[0-4]\d{2}|6[0-4]\d{3}|[1-5]\d{4}|[1-9]\d{0,3}|0)$/.match?(value)`
PHP	`preg_match('~^(6553[0-5]|655[0-2]\d|65[0-4]\d{2}|6[0-4]\d{3}|[1-5]\d{4}|[1-9]\d{0,3}|0)$~', $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 port number against a source of truth (database, API, or checksum) where it matters.