

Aadhaar (with spaces)

Aadhaar displayed as 1234 5678 9012.

```
^[2-9]\d{3}\s\d{4}\s\d{4}$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
[2-9]	any of: 2–9
\d{3}	exactly 3 times: digit (0–9)
\s	any whitespace
\d{4}	exactly 4 times: digit (0–9)
\s	any whitespace
\d{4}	exactly 4 times: digit (0–9)
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript	<pre>const re = /^[2-9]\d{3}\s\d{4}\s\d{4}\$/; re.test(value);</pre>
Python	<pre>import re re.match(r"^[2-9]\d{3}\s\d{4}\s\d{4}\$", value)</pre>
Java	<pre>Pattern.compile("^[2-9]\\d{3}\\s\\d{4}\\s\\d{4}\$").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"^[2-9]\d{3}\s\d{4}\s\d{4}\$");</pre>
Go	<pre>regexp.MustCompile(`^[2-9]\d{3}\s\d{4}\s\d{4}\$`).MatchString(value)</pre>
Ruby	<pre>/^[2-9]\d{3}\s\d{4}\s\d{4}\$/ .match?(value)</pre>
PHP	<pre>preg_match('~^[2-9]\d{3}\s\d{4}\s\d{4}\$~', \$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 aadhaar (with spaces) against a source of truth (database, API, or checksum) where it matters.