

US Social Security Number

SSN with known-invalid ranges excluded.

```
^(?!000|666|9\d{2})\d{3}-(?!00)\d{2}-(?!0000)\d{4}$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
(?!	start of a negative lookahead — NOT followed by
000	literal text "000"
	OR — try the alternative
666	literal text "666"
	OR — try the alternative
9	literal text "9"
\d{2}	exactly 2 times: digit (0–9)
)	end of group
\d{3}	exactly 3 times: digit (0–9)
-	literal text "-"
(?!	start of a negative lookahead — NOT followed by
00	literal text "00"
)	end of group
\d{2}	exactly 2 times: digit (0–9)
-	literal text "-"
(?!	start of a negative lookahead — NOT followed by
0000	literal text "0000"
)	end of group
\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 = /^(?!000 666 9\d{2})\d{3}-(?!00)\d{2}-(?!0000)\d{4}\$/; re.test(value);</pre>
Python	<pre>import re re.match(r"^(?!000 666 9\d{2})\d{3}-(?!00)\d{2}-(?!0000)\d{4}\$", value)</pre>
Java	<pre>Pattern.compile("^(?!000 666 9\d{2})\d{3}-(?!00)\d{2}-(?!0000)\d{4}\$").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"^(?!000 666 9\d{2})\d{3}-(?!00)\d{2}-(?!0000)\d{4}\$");</pre>

Go	<code>regexp.MustCompile(`^(?!000 666 9\d{2})\d{3}-(?!00)\d{2}-(?!0000)\d{4}\$`).MatchString(value)</code>
Ruby	<code>/^(?!000 666 9\d{2})\d{3}-(?!00)\d{2}-(?!0000)\d{4}\$/ .match?(value)</code>
PHP	<code>preg_match('~^(?!000 666 9\d{2})\d{3}-(?!00)\d{2}-(?!0000)\d{4}\$~', \$value);</code>

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 us social security number against a source of truth (database, API, or checksum) where it matters.