

Strong password

8+ chars, uppercase, lowercase, digit, symbol.

```
^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[^\w\s]).{8,}$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
(?=	start of a positive lookahead — followed by
.*	zero or more of any character
[a-z]	any of: lowercase letters
)	end of group
(?=	start of a positive lookahead — followed by
.*	zero or more of any character
[A-Z]	any of: uppercase letters
)	end of group
(?=	start of a positive lookahead — followed by
.*	zero or more of any character
\d	any digit (0–9)
)	end of group
(?=	start of a positive lookahead — followed by
.*	zero or more of any character
[^\w\s]	any character except word chars (A–Z, a–z, 0–9, _), whitespace
)	end of group
{8,}	8 or more times of any character
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript	<pre>const re = /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[^\w\s]).{8,}\$/; re.test(value);</pre>
Python	<pre>import re re.match(r"^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[^\w\s]).{8,}\$", value)</pre>
Java	<pre>Pattern.compile("^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[^\w\s]).{8,}\$").matcher(value). matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[^\w\s]).{8,}\$");</pre>
Go	<pre>regexp.MustCompile(`^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[^\w\s]).{8,}\$`).MatchString(va lue)</pre>

Ruby

```
/^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[^\w\s]).{8,}$/ .match?(value)
```

PHP

```
preg_match('~^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[^\w\s]).{8,}$~', $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.
- **Case sensitivity.** Letter ranges are case-sensitive — use the `i` flag if the input case can vary.
- **Greedy “.”.** A greedy `.` / `.*` can match more than intended. Use a lazy version (`.*?`) or a negated class (`[^...]`) to stop at the right place.
- **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 strong password against a source of truth (database, API, or checksum) where it matters.