Environment variable reference

$VAR or ${VAR} in shell scripts.

```
\$\{?([A-Z_][A-Z0-9_]*)\}?
```

Token-by-token breakdown

Token	Meaning
`\$`	literal “$”
`\{?`	optional (zero or one): literal “{”
`(`	start of a capturing group
`[A-Z_]`	any of: uppercase letters, “_”
`[A-Z0-9_]*`	zero or more: any of: uppercase letters, digits, “_”
`)`	end of group
`\}?`	optional (zero or one): literal “}”

Quick usage in 7 languages

JavaScript

```
const re = /\$\{?([A-Z_][A-Z0-9_]*)\}?/;
re.test(value);
```

Python

```
import re
re.match(r"\$\{?([A-Z_][A-Z0-9_]*)\}?", value)
```

Java

```
Pattern.compile("\\$\\{?([A-Z_][A-Z0-9_]*)\\}?").matcher(value).matches();
```

C# / .NET

```
Regex.IsMatch(value, @"\$\{?([A-Z_][A-Z0-9_]*)\}?");
```

Go

```
regexp.MustCompile(`\$\{?([A-Z_][A-Z0-9_]*)\}?`).MatchString(value)
```

Ruby

```
/\$\{?([A-Z_][A-Z0-9_]*)\}?/.match?(value)
```

PHP

```
preg_match('~\$\{?([A-Z_][A-Z0-9_]*)\}?~', $value);
```

Common pitfalls

- **Not anchored.** Without ^ and $ this can match a substring anywhere in the input — add anchors if you need the whole value to conform.
- **Uppercase only.** Ranges like [A-Z] won't match lowercase. Add a-z (or the i flag) if lowercase input should be accepted.
- **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 environment variable reference against a source of truth (database, API, or checksum) where it matters.

© 2026 regexguide.com · a human-curated regex explainer & pattern library

Disclaimer: provided as-is, without warranty — test against your own data before production use.