REGEX CHEAT SHEET

regexguide.com/patterns/constant-case.html

CONSTANT_CASE

All-uppercase with underscores.

```
^[A-Z][A-Z0-9]*(_[A-Z0-9]+)*$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
[A-Z]	any of: uppercase letters
[A-Z0-9]*	zero or more: any of: uppercase letters, digits
(	start of a capturing group
_	literal text “_”
[A-Z0-9]+	one or more: any of: uppercase letters, digits
)*	end of group, zero or more
$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript

```
const re = /^[A-Z][A-Z0-9]*(_[A-Z0-9]+)*$/;
re.test(value);
```

Python

```
import re
re.match(r"^[A-Z][A-Z0-9]*(_[A-Z0-9]+)*$", value)
```

Java

```
Pattern.compile("^[A-Z][A-Z0-9]*(_[A-Z0-9]+)*$").matcher(value).matches();
```

C# / .NET

```
Regex.IsMatch(value, @"^[A-Z][A-Z0-9]*(_[A-Z0-9]+)*$");
```

Go

```
regexp.MustCompile(`^[A-Z][A-Z0-9]*(_[A-Z0-9]+)*$`).MatchString(value)
```

Ruby

```
/^[A-Z][A-Z0-9]*(_[A-Z0-9]+)*$/.match?(value)
```

PHP

```
preg_match('~^[A-Z][A-Z0-9]*(_[A-Z0-9]+)*$~', $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.
- **Uppercase only.** Ranges like [A-Z] won't match lowercase. Add a-z (or the i flag) if lowercase input should be accepted.
- **Watch for backtracking.** This pattern nests quantifiers; on adversarial input that can cause exponential backtracking (ReDoS). Test with long non-matching strings, or use an RE2-based engine.
- **Validate beyond format.** Matching the format doesn't guarantee the value is real. Confirm the constant_case against a source of truth (database, API, or checksum) where it matters.