

Shell variable assignment

Bash variable assignment — NAME=value.

```
^[A-Za-z_][A-Za-z0-9_]*=
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
[A-Za-z_]	any of: uppercase letters, lowercase letters, “_”
[A-Za-z0-9_]*	zero or more: any of: uppercase letters, lowercase letters, digits, “_”
=	literal text “=”

Quick usage in 7 languages

JavaScript	<pre>const re = /^[A-Za-z_][A-Za-z0-9_]*=/; re.test(value);</pre>
Python	<pre>import re re.match(r"^[A-Za-z_][A-Za-z0-9_]*=", value)</pre>
Java	<pre>Pattern.compile("^[A-Za-z_][A-Za-z0-9_]*=").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"^[A-Za-z_][A-Za-z0-9_]*=");</pre>
Go	<pre>regexp.MustCompile(`^[A-Za-z_][A-Za-z0-9_]*=`).MatchString(value)</pre>
Ruby	<pre>/^[A-Za-z_][A-Za-z0-9_]*=/.match?(value)</pre>
PHP	<pre>preg_match('~^[A-Za-z_][A-Za-z0-9_]*=~', \$value);</pre>

Common pitfalls

- **Anchored to the whole string.** This pattern uses ^, so it requires the entire input to match. To find it inside a longer text, drop the anchors and use the global (g) flag.
- **ASCII letters only.** [a-zA-Z] excludes accented and non-Latin letters (é, ü, ß, ñ, and non-Latin scripts). For international input use Unicode properties like \p{L} with the u flag.
- **Validate beyond format.** Matching the format doesn't guarantee the value is real. Confirm the shell variable assignment against a source of truth (database, API, or checksum) where it matters.