

SHA-1 hash

40-character SHA-1 hash.

```
^[0-9a-fA-F]{40}$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
[0-9a-fA-F]{40}	exactly 40 times: any of: digits, a–f, A–F
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript	<pre>const re = /^[0-9a-fA-F]{40}\$/; re.test(value);</pre>
Python	<pre>import re re.match(r"^[0-9a-fA-F]{40}\$", value)</pre>
Java	<pre>Pattern.compile("^[0-9a-fA-F]{40}\$").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"^[0-9a-fA-F]{40}\$");</pre>
Go	<pre>regexp.MustCompile(`^[0-9a-fA-F]{40}\$`).MatchString(value)</pre>
Ruby	<pre>/^[0-9a-fA-F]{40}\$/ .match?(value)</pre>
PHP	<pre>preg_match('~^[0-9a-fA-F]{40}\$~', \$value);</pre>

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.
- **Validate beyond format.** Matching the format doesn't guarantee the value is real. Confirm the sha-1 hash against a source of truth (database, API, or checksum) where it matters.