

Git SHA (full)

Full 40-character SHA-1 commit hash.

```
\b[0-9a-f]{40}\b
```

Token-by-token breakdown

Token	Meaning
\b	word boundary
[0-9a-f]{40}	exactly 40 times: any of: digits, a–f
\b	word boundary

Quick usage in 7 languages

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

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.
- **Case sensitivity.** Letter ranges are case-sensitive — use the `i` flag if the input case can vary.
- **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 git sha (full) against a source of truth (database, API, or checksum) where it matters.