

YouTube video ID

11-character YouTube video identifier.

```
^[A-Za-z0-9_-]{11}$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
[A-Za-z0-9_-]{11}	exactly 11 times: any of: uppercase letters, lowercase letters, digits, "_", "-"
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript	<pre>const re = /^[A-Za-z0-9_-]{11}\$/; re.test(value);</pre>
Python	<pre>import re re.match(r"^[A-Za-z0-9_-]{11}\$", value)</pre>
Java	<pre>Pattern.compile("^[A-Za-z0-9_-]{11}\$").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"^[A-Za-z0-9_-]{11}\$");</pre>
Go	<pre>regexp.MustCompile(`^[A-Za-z0-9_-]{11}\$`).MatchString(value)</pre>
Ruby	<pre>/^[A-Za-z0-9_-]{11}\$/ .match?(value)</pre>
PHP	<pre>preg_match('~^[A-Za-z0-9_-]{11}\$~', \$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.
- **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 youtube video id against a source of truth (database, API, or checksum) where it matters.