

Markdown link

Markdown link in [text](url) format.

```
\[([^\]]+)\]\([^\)]+\)
```

Token-by-token breakdown

Token	Meaning
\[	literal "["
(	start of a capturing group
[^\]]+	one or more: any character except "]"
)	end of group
\]	literal "]"
\(	literal "("
(	start of a capturing group
[^\)]+	one or more: any character except ")"
)	end of group
\)	literal ")"

Quick usage in 7 languages

JavaScript `const re = /\[([^\]]+)\]\([^\)]+\)/;
re.test(value);`

Python `import re
re.match(r"\[([^\]]+)\]\([^\)]+\)", value)`

Java `Pattern.compile("\\[([^\]]+)\]\([^\)]+\)").matcher(value).matches();`

C# / .NET `Regex.IsMatch(value, @"\[([^\]]+)\]\([^\)]+\)");`

Go `regexp.MustCompile(`\[([^\]]+)\]\([^\)]+\)`).MatchString(value)`

Ruby `/\[([^\]]+)\]\([^\)]+\)/.match?(value)`

PHP `preg_match('~\[([^\]]+)\]\([^\)]+\)~', $value);`

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.
- **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 markdown link against a source of truth (database, API, or checksum) where it matters.