# is-glob [NPM version](https://www.npmjs.com/package/is-glob) [NPM monthly downloads](https://npmjs.org/package/is-glob) [NPM total downloads](https://npmjs.org/package/is-glob) [Build Status](https://github.com/micromatch/is-glob/actions)

Returns true if the given string looks like a glob pattern or an extglob pattern. This makes it easy to create code that only uses external modules like node-glob when necessary, resulting in much faster code execution and initialization time, and a better user experience.

Please consider following this project's author, [Jon Schlinkert](https://github.com/jonschlinkert), and consider starring the project to show your :heart: and support.

## Install

Install with [npm](https://www.npmjs.com/):

$ npm install --save is-glob

You might also be interested in [is-valid-glob](https://github.com/jonschlinkert/is-valid-glob) and [has-glob](https://github.com/jonschlinkert/has-glob).

## Usage

var isGlob = require('is-glob');

### Default behavior

**True**

Patterns that have glob characters or regex patterns will return true:

isGlob('!foo.js');

isGlob('\*.js');

isGlob('\*\*/abc.js');

isGlob('abc/\*.js');

isGlob('abc/(aaa|bbb).js');

isGlob('abc/[a-z].js');

isGlob('abc/{a,b}.js');

//=> true

Extglobs

isGlob('abc/@(a).js');

isGlob('abc/!(a).js');

isGlob('abc/+(a).js');

isGlob('abc/\*(a).js');

isGlob('abc/?(a).js');

//=> true

**False**

Escaped globs or extglobs return false:

isGlob('abc/\\@(a).js');

isGlob('abc/\\!(a).js');

isGlob('abc/\\+(a).js');

isGlob('abc/\\\*(a).js');

isGlob('abc/\\?(a).js');

isGlob('\\!foo.js');

isGlob('\\\*.js');

isGlob('\\\*\\\*/abc.js');

isGlob('abc/\\\*.js');

isGlob('abc/\\(aaa|bbb).js');

isGlob('abc/\\[a-z].js');

isGlob('abc/\\{a,b}.js');

//=> false

Patterns that do not have glob patterns return false:

isGlob('abc.js');

isGlob('abc/def/ghi.js');

isGlob('foo.js');

isGlob('abc/@.js');

isGlob('abc/+.js');

isGlob('abc/?.js');

isGlob();

isGlob(null);

//=> false

Arrays are also false (If you want to check if an array has a glob pattern, use [has-glob](https://github.com/jonschlinkert/has-glob)):

isGlob(['\*\*/\*.js']);

isGlob(['foo.js']);

//=> false

### Option strict

When options.strict === false the behavior is less strict in determining if a pattern is a glob. Meaning that some patterns that would return false may return true. This is done so that matching libraries like [micromatch](https://github.com/micromatch/micromatch) have a chance at determining if the pattern is a glob or not.

**True**

Patterns that have glob characters or regex patterns will return true:

isGlob('!foo.js', {strict: false});

isGlob('\*.js', {strict: false});

isGlob('\*\*/abc.js', {strict: false});

isGlob('abc/\*.js', {strict: false});

isGlob('abc/(aaa|bbb).js', {strict: false});

isGlob('abc/[a-z].js', {strict: false});

isGlob('abc/{a,b}.js', {strict: false});

//=> true

Extglobs

isGlob('abc/@(a).js', {strict: false});

isGlob('abc/!(a).js', {strict: false});

isGlob('abc/+(a).js', {strict: false});

isGlob('abc/\*(a).js', {strict: false});

isGlob('abc/?(a).js', {strict: false});

//=> true

**False**

Escaped globs or extglobs return false:

isGlob('\\!foo.js', {strict: false});

isGlob('\\\*.js', {strict: false});

isGlob('\\\*\\\*/abc.js', {strict: false});

isGlob('abc/\\\*.js', {strict: false});

isGlob('abc/\\(aaa|bbb).js', {strict: false});

isGlob('abc/\\[a-z].js', {strict: false});

isGlob('abc/\\{a,b}.js', {strict: false});

//=> false

## About

**Contributing** Pull requests and stars are always welcome. For bugs and feature requests, [please create an issue](http://../../issues/new).

**Running Tests** Running and reviewing unit tests is a great way to get familiarized with a library and its API. You can install dependencies and run tests with the following command:

$ npm install && npm test

**Building docs** *(This project's readme.md is generated by* [*verb*](https://github.com/verbose/verb-generate-readme)*, please don't edit the readme directly. Any changes to the readme must be made in the* [*.verb.md*](http://.verb.md) *readme template.)*

To generate the readme, run the following command:

$ npm install -g verbose/verb#dev verb-generate-readme && verb

### Related projects

You might also be interested in these projects:

* [assemble](https://www.npmjs.com/package/assemble): Get the rocks out of your socks! Assemble makes you fast at creating web projects… [more](https://github.com/assemble/assemble) | [homepage](https://github.com/assemble/assemble)
* [base](https://www.npmjs.com/package/base): Framework for rapidly creating high quality, server-side node.js applications, using plugins like building blocks | [homepage](https://github.com/node-base/base)
* [update](https://www.npmjs.com/package/update): Be scalable! Update is a new, open source developer framework and CLI for automating updates… [more](https://github.com/update/update) | [homepage](https://github.com/update/update)
* [verb](https://www.npmjs.com/package/verb): Documentation generator for GitHub projects. Verb is extremely powerful, easy to use, and is used… [more](https://github.com/verbose/verb) | [homepage](https://github.com/verbose/verb)

### Contributors

| **Commits** | **Contributor** |
| --- | --- |
| 47 | [jonschlinkert](https://github.com/jonschlinkert) |
| 5 | [doowb](https://github.com/doowb) |
| 1 | [phated](https://github.com/phated) |
| 1 | [danhper](https://github.com/danhper) |
| 1 | [paulmillr](https://github.com/paulmillr) |

### Author

**Jon Schlinkert**

* [GitHub Profile](https://github.com/jonschlinkert)
* [Twitter Profile](https://twitter.com/jonschlinkert)
* [LinkedIn Profile](https://linkedin.com/in/jonschlinkert)

### License

Copyright © 2019, [Jon Schlinkert](https://github.com/jonschlinkert). Released under the [MIT License](http://license).

*This file was generated by* [*verb-generate-readme*](https://github.com/verbose/verb-generate-readme)*, v0.8.0, on March 27, 2019.*