

build passing

supports-color

Detect whether a terminal supports color

Install

```
$ npm install supports-color
```

Usage

```
const supportsColor = require('supports-color');

if (supportsColor.stdout) {
  console.log('Terminal stdout supports color');
}

if (supportsColor.stdout.has256) {
  console.log('Terminal stdout supports 256 colors');
}

if (supportsColor.stderr.has16m) {
  console.log('Terminal stderr supports 16 million colors (truecolor)');
}
```

API

Returns an `Object` with a `stdout` and `stderr` property for testing either streams. Each property is an `Object`, or `false` if color is not supported.

The `stdout` / `stderr` objects specifies a level of support for color through a `.level` property and a corresponding flag:

- `.level = 1` and `.hasBasic = true` : Basic color support (16 colors)
- `.level = 2` and `.has256 = true` : 256 color support
- `.level = 3` and `.has16m = true` : Truecolor support (16 million colors)

Info

It obeys the `--color` and `--no-color` CLI flags.

For situations where using `--color` is not possible, use the environment variable `FORCE_COLOR=1` (level 1), `FORCE_COLOR=2` (level 2), or `FORCE_COLOR=3` (level 3) to forcefully enable color, or `FORCE_COLOR=0` to forcefully disable. The use of `FORCE_COLOR` overrides all other color support checks.

Explicit 256/Truecolor mode can be enabled using the `--color=256` and `--color=16m` flags, respectively.

Related

- [supports-color-cli](#) - CLI for this module
- [chalk](#) - Terminal string styling done right

Maintainers

- [Sindre Sorhus](#)
- [Josh Junon](#)

[Get professional support for this package with a Tidelift subscription](#)

Tidelift helps make open source sustainable for maintainers while giving companies assurances about security, maintenance, and licensing for their dependencies.
