

## ansi-regex

Regular expression for matching ANSI escape codes

### Install

```
$ npm install ansi-regex
```

### Usage

```
const ansiRegex = require('ansi-regex');

ansiRegex().test('\u001B[4mcake\u001B[0m');
//=> true

ansiRegex().test('cake');
//=> false

'\u001B[4mcake\u001B[0m'.match(ansiRegex());
//=> ['\u001B[4m', '\u001B[0m']

'\u001B[4mcake\u001B[0m'.match(ansiRegex({onlyFirst: true}));
//=> ['\u001B[4m']

'\u001B]8;;https://github.com\u0007click\u001B]8;;\u0007'.match(ansiRegex());
//=> ['\u001B]8;;https://github.com\u0007', '\u001B]8;;\u0007']
```

### API

#### ansiRegex(options?)

Returns a regex for matching ANSI escape codes.

**options**   Type: object

**onlyFirst**   Type: boolean Default: false (*Matches any ANSI escape codes in a string*)

Match only the first ANSI escape.

### FAQ

#### Why do you test for codes not in the ECMA 48 standard?

Some of the codes we run as a test are codes that we acquired finding various lists of non-standard or manufacturer specific codes. We test for both standard and non-standard codes, as most of them follow the same or similar format

and can be safely matched in strings without the risk of removing actual string content. There are a few non-standard control codes that do not follow the traditional format (i.e. they end in numbers) thus forcing us to exclude them from the test because we cannot reliably match them.

On the historical side, those ECMA standards were established in the early 90's whereas the VT100, for example, was designed in the mid/late 70's. At that point in time, control codes were still pretty ungoverned and engineers used them for a multitude of things, namely to activate hardware ports that may have been proprietary. Somewhere else you see a similar 'anarchy' of codes is in the x86 architecture for processors; there are a ton of "interrupts" that can mean different things on certain brands of processors, most of which have been phased out.

## Maintainers

- Sindre Sorhus
- Josh Junon

---

**[https://tidelift.com/subscription/pkg/npm-ansi-regex?utm\\_source=npm-ansi-regex&utm\\_medium=referral&utm\\_campaign=package\\_maintenance](https://tidelift.com/subscription/pkg/npm-ansi-regex?utm_source=npm-ansi-regex&utm_medium=referral&utm_campaign=package_maintenance)**

**Tidelift helps make open source sustainable for maintainers while giving companies**