is-number

Returns true if the value is a finite number.

Please consider following this project's author, Jon Schlinkert, and consider starring the project to show your :heart: and support.

Install

Install with npm:

Why is this needed?

In JavaScript, it's not always as straightforward as it should be to reliably check if a value is a number. It's common for devs to use `+`, `-`, or `Number()` to cast a string value to a number (for example, when values are returned from user input, regex matches, parsers, etc). But there are many non-intuitive edge cases that yield unexpected results:

```
console.log(+[]); //=> 0
console.log(+''); //=> 0
console.log(+'   '); //=> 0
console.log(typeof NaN); //=> 'number'
```

This library offers a performant way to smooth out edge cases like these.

Usage

```
const isNumber = require('is-number');
```

See the tests for more examples.

true

```
isNumber(5e3);               // true
isNumber(0xff);              // true
isNumber(-1.1);              // true
isNumber(0);                 // true
isNumber(1);                 // true
isNumber(1.1);               // true
isNumber(10);                // true
isNumber(10.10);             // true
isNumber(100);               // true
isNumber('-1.1');            // true
isNumber('0');               // true
isNumber('012');             // true
isNumber('0xff');            // true
```


- isobject: Returns true if the value is an object and not an array or null. | homepage
- kind-of: Get the native type of a value. | homepage

Contributors

Commits	Contributor
49	jonschlinkert
5	charlike-old
1	benaadams
1	realityking

Author

Jon Schlinkert

- LinkedIn Profile
- GitHub Profile
- Twitter Profile

License

This file was generated by verb-generate-readme, v0.6.0, on June 15, 2018.

About

Contributing

Pull requests and stars are always welcome. For bugs and feature requests, please create an issue.

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, please don't edit the readme directly. Any changes to the readme must be made in the .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:

- is-plain-object: Returns true if an object was created by the `Object` constructor. | homepage
- is-primitive: Returns `true` if the value is a primitive. | homepage


```
isNumber('1');                  // true
isNumber('1.1');                // true
isNumber('10');                 // true
isNumber('10.10');              // true
isNumber('100');                // true
isNumber('5e3');                // true
isNumber(parseInt('012'));      // true
isNumber(parseFloat('012'));    // true
```

False

Everything else is false, as you would expect:

```
isNumber(Infinity);             // false
isNumber(NaN);                  // false
isNumber(null);                 // false
isNumber(undefined);            // false
isNumber('');                   // false
isNumber('   ');                // false
isNumber('foo');                // false
isNumber([1]);                  // false
isNumber([]);                   // false
isNumber(function () {});       // false
isNumber({});                   // false
```

Release history

7.0.0

- Refactor. Now uses `.isFinite` if it exists.
- Performance is about the same as v6.0 when the value is a string or number. But it's now 3x-4x faster when the value is not a string or number.

6.0.0

- Optimizations, thanks to @benaadams.

5.0.0

Breaking changes

- removed support for `instanceof Number` and `instanceof String`

Benchmarks

As with all benchmarks, take these with a grain of salt. See the benchmarks for more detail.

```
# all
v7.0 x 413,222 ops/sec ±2.02% (86 runs
sampled)
v6.0 x 111,061 ops/sec ±1.29% (85 runs
sampled)
parseFloat x 317,596 ops/sec ±1.36% (86
runs sampled)
fastest is 'v7.0'

# string
v7.0 x 3,054,496 ops/sec ±1.05% (89 runs
sampled)
v6.0 x 2,957,781 ops/sec ±0.98% (88 runs
sampled)
parseFloat x 3,071,060 ops/sec ±1.13%
(88 runs sampled)
fastest is 'parseFloat,v7.0'

# number
v7.0 x 3,146,895 ops/sec ±0.89% (89 runs
sampled)
v6.0 x 3,214,038 ops/sec ±1.07% (89 runs
sampled)
parseFloat x 3,077,588 ops/sec ±1.07%
(87 runs sampled)
fastest is 'v6.0'
```