# Bytes utility

[NPM Version](https://npmjs.org/package/bytes) [NPM Downloads](https://npmjs.org/package/bytes) [Build Status](https://github.com/visionmedia/bytes.js/actions?query=workflow%3Aci) [Test Coverage](https://coveralls.io/r/visionmedia/bytes.js?branch=master)

Utility to parse a string bytes (ex: 1TB) to bytes (1099511627776) and vice-versa.

## Installation

This is a [Node.js](https://nodejs.org/en/) module available through the [npm registry](https://www.npmjs.com/). Installation is done using the [npm install command](https://docs.npmjs.com/getting-started/installing-npm-packages-locally):

$ npm install bytes

## Usage

var bytes = require('bytes');

#### bytes(number｜string value, [options]): number｜string｜null

Default export function. Delegates to either bytes.format or bytes.parse based on the type of value.

**Arguments**

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| value | number｜string | Number value to format or string value to parse |
| options | Object | Conversion options for format |

**Returns**

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| results | string｜number｜null | Return null upon error. Numeric value in bytes, or string value otherwise. |

**Example**

bytes(1024);

// output: '1KB'

bytes('1KB');

// output: 1024

#### bytes.format(number value, [options]): string｜null

Format the given value in bytes into a string. If the value is negative, it is kept as such. If it is a float, it is rounded.

**Arguments**

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| value | number | Value in bytes |
| options | Object | Conversion options |

**Options**

| **Property** | **Type** | **Description** |
| --- | --- | --- |
| decimalPlaces | number｜null | Maximum number of decimal places to include in output. Default value to 2. |
| fixedDecimals | boolean｜null | Whether to always display the maximum number of decimal places. Default value to false |
| thousandsSeparator | string｜null | Example of values: ' ', ',' and '.'... Default value to ''. |
| unit | string｜null | The unit in which the result will be returned (B/KB/MB/GB/TB). Default value to '' (which means auto detect). |
| unitSeparator | string｜null | Separator to use between number and unit. Default value to ''. |

**Returns**

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| results | string｜null | Return null upon error. String value otherwise. |

**Example**

bytes.format(1024);

// output: '1KB'

bytes.format(1000);

// output: '1000B'

bytes.format(1000, {thousandsSeparator: ' '});

// output: '1 000B'

bytes.format(1024 \* 1.7, {decimalPlaces: 0});

// output: '2KB'

bytes.format(1024, {unitSeparator: ' '});

// output: '1 KB'

#### bytes.parse(string｜number value): number｜null

Parse the string value into an integer in bytes. If no unit is given, or value is a number, it is assumed the value is in bytes.

Supported units and abbreviations are as follows and are case-insensitive:

* b for bytes
* kb for kilobytes
* mb for megabytes
* gb for gigabytes
* tb for terabytes
* pb for petabytes

The units are in powers of two, not ten. This means 1kb = 1024b according to this parser.

**Arguments**

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| value | string｜number | String to parse, or number in bytes. |

**Returns**

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| results | number｜null | Return null upon error. Value in bytes otherwise. |

**Example**

bytes.parse('1KB');

// output: 1024

bytes.parse('1024');

// output: 1024

bytes.parse(1024);

// output: 1024

## License

[MIT](http://license)