

Numeral.js v. 2.0.6

A javascript library for formatting and manipulating numbers.

[GitHub](#)[.ZIP](#)[.TGZ](#)

Use it

In the Browser

```
<script src="numeral.min.js"></script>
```

or include from cdnjs.com

```
<script src="//cdnjs.cloudflare.com/ajax/libs/numeral.js/2.0.6/numeral.min.js"></script>
```

In Node.js

```
npm install numeral
```

```
var numeral = require('numeral');
```

Create

Create an instance of a numeral. Numeral takes numbers or strings that it tries to convert into a number.

```
var myNumeral = numeral(1000);

var value = myNumeral.value();
// 1000

var myNumeral2 = numeral('1,000');

var value2 = myNumeral2.value();
// 1000
```

Input	Value
numeral(974)	974
numeral(0.12345)	0.12345
numeral('10,000.12')	10000.12
numeral('23rd')	23
numeral('\$10,000.00')	10000
numeral('100B')	100
numeral('3.467TB')	3467000000000
numeral('-76%')	-0.76
numeral('2:23:57')	NaN

Format

Numbers can be formatted to look like currency, percentages, times, or even plain old numbers with decimal places, thousands, and abbreviations. And you can always [create a custom format](#).

```
var string = numeral(1000).format('0,0');  
// '1,000'
```

Numbers

Number	Format	String
10000	'0,0.0000'	10,000.0000
10000.23	'0,0'	10,000
10000.23	' +0,0'	+10,000
-10000	'0,0.0'	-10,000.0
10000.1234	'0.000'	10000.123
100.1234	'00000'	00100

Number	Format	String
1000.1234	'000000,0'	001,000
10	'000.00'	010.00
10000.1234	'0[.]000000'	10000.12340
-10000	'(0,0.0000)'	(10,000.0000)
-0.23	'00'	-.23
-0.23	'(.00)'	(.23)
0.23	'0.000000'	0.23000
0.23	'0.0[0000]'	0.23
1230974	'0.0a'	1.2m
1460	'0 a'	1 k
-104000	'0a'	-104k
1	'0o'	1st
100	'0o'	100th

Currency

Number	Format	String
1000.234	'\$0,0.00'	\$1,000.23
1000.2	'0,0[.]00 \$'	1,000.20 \$
1001	'\$ 0,0[.]00'	\$ 1,001
-1000.234	'(\$0,0)'	(\$1,000)
-1000.234	'\$0.00'	-\$1000.23
1230974	'(\$ 0.00 a)'	\$ 1.23 m

Bytes

Number	Format	String
100	'Ob'	100B
1024	'Ob'	1KB
2048	'O ib'	2 KiB
3072	'0.0 b'	3.1 KB
7884486213	'0.00b'	7.88GB
3467479682787	'0.000 ib'	3.154 TiB

Percentages

Number	Format	String
1	'0%'	100%
0.974878234	'0.000%'	97.488%
-0.43	'0 %'	-43 %
0.43	'(0.000 %)'	43.000 %

Time

Number	Format	String
25	'00:00:00'	0:00:25
238	'00:00:00'	0:03:58
63846	'00:00:00'	17:44:06

Exponential

Number	Format	String
1123456789	'0,0e+0'	1e+9
12398734.202	'0.00e+0'	1.24e+7

Number	Format	String
0.000123987	'0.000e+0'	1.240e-4

Functions

Value

The value is always available.

```
var number = numeral(1000);

var string = number.format('0,0');
// '1,000'

var value = number.value();
// 1000
```

Manipulate

Not that you will use these often, but they're there when you need them.

```
var number = numeral(1000);

var added = number.add(10);
// 1010
```

Before	Function	After
1000	.add(100)	1100
1100	.subtract(100)	1000
1000	.multiply(100)	100000
100000	.divide(100)	1000

Set

Set the value of your numeral object.

```
var number = numeral();

number.set(1000);
```

```
var value = number.value();  
// 1000
```

Difference

Find the difference between your numeral object and a value

```
var number = numeral(1000),  
    value = 100;  
  
var difference = number.difference(value);  
// 900
```

Clone

Go ahead and clone any numeral object while you're at it.

```
var a = numeral(1000);  
var b = numeral(a);  
var c = a.clone();  
  
var aVal = a.set(2000).value();  
// 2000  
  
var bVal = b.value();  
// 1000  
  
var cVal = c.add(10).value();  
// 1010
```

Settings

Default Formatting

Set a default format so you can use .format() without a string. The default format to '0,0'

```
var number = numeral(1000);  
  
number.format();  
// '1,000'  
  
numeral.defaultFormat('$0,0.00');  
  
number.format();  
// '$1,000.00'
```

Custom Zero and Null Formatting

Set a custom output when formatting numerals with a value of 0 or null

```
var number = numeral(0);
var nullNumber = numeral(null);

numeral.zeroFormat('N/A');
numeral.nullFormat('N/A');

var zero = number.format('0.0')
// 'N/A'

var na = nullNumber.format('0.0')
// 'N/A'
```

Locales

Let's make this useable all over the place!

```
// load a locale
numeral.register('locale', 'fr', {
  delimiters: {
    thousands: ' ',
    decimal: ',',
  },
  abbreviations: {
    thousand: 'k',
    million: 'm',
    billion: 'b',
    trillion: 't'
  },
  ordinal : function (number) {
    return number === 1 ? 'er' : 'ème';
  },
  currency: {
    symbol: '€'
  }
});

// switch between locales
numeral.locale('fr');
```

As I am not fluent in every locale on the planet, please feel free to create locale files of your own by submitting a pull request. Don't forget to create both the locale file (example: locales/fr.js) and the locale test (example: tests/locales/fr.js). Thanks for helping out.

Formats

Adding your own custom formats is as easy as adding a locale.

```

// load a format
numeral.register('format', 'percentage', {
  regexps: {
    format: /(%)/,
    unformat: /(%) /
  },
  format: function(value, format, roundingFunction) {
    var space = numeral._.includes(format, ' %') ? ' ' : '',
        output;

    value = value * 100;

    // check for space before %
    format = format.replace(/\s?\%/ , '');

    output = numeral._.numberToFormat(value, format, roundingFunction);

    if (numeral._.includes(output, ' ')) {
      output = output.split(' ');

      output.splice(-1, 0, space + '%');

      output = output.join(' ');
    } else {
      output = output + space + '%';
    }

    return output;
  },
  unformat: function(string) {
    return numeral._.stringToNumber(string) * 0.01;
  }
});

// use your custom format
numeral().format('0%');

```

Acknowledgements

Numeral.js, while less complex, was inspired by and heavily borrowed from [Moment.js](#)

Project by [adamwdraper](#)

