
pynumeral Documentation

Release 0.2.0

Guillaume Thomas

Nov 15, 2019

Contents:

1	Usage	3
1.1	Formatting	3
2	Indices and tables	7

`pynumeral` is a python implementation of number formatting using `numeraljs` format. This project was motivated when we had to render numbers backend-side (which is written in python) while keep using formats defined by users. It has been coded regardless of the javascript implementation. It has been tested on a fairly large amount of formats but there are not any guarantees that it covers all the cases. Since unit testing is pretty simple, contributions are more than welcome.

CHAPTER 1

Usage

`pynumeral` provides one api: `format`. Below you can find many examples. For more informations about format tokens, please read [numeraljs documentation](#).

```
import pynumeral

pynumeral.format(3.141592653589793, "0.00")
# 3.14
```

1.1 Formatting

1.1.1 Numbers

```
import pynumeral

pynumeral.format(10000, None)
# "10000"
pynumeral.format(10000, "0,0.0000")
# "10,000.0000"
pynumeral.format(10000.23, "0,0")
# "10,000"
pynumeral.format(10000.23, "+0,0")
# "+10,000"
pynumeral.format(-10000, "0,0.0")
# "-10,000.0"
pynumeral.format(10000.1234, "0.000")
# "10000.123"
pynumeral.format(100.1234, "00000")
# "00100"
pynumeral.format(1000.1234, "000000,0")
```

(continues on next page)

(continued from previous page)

```
# "001,000"
pynumeral.format(10000.1234, "0[.]00000")
# "10000.12340"
pynumeral.format(-10000, "(0,0.0000)")
# "(10,000.0000)"
pynumeral.format(-0.23, ".00")
# "-.23"
pynumeral.format(-0.23, "(.00)")
# "(.23)"
pynumeral.format(0.23, "0.00000")
# "0.23000"
pynumeral.format(1123456789, "0,0e+0")
# "1e+09"
pynumeral.format(12398734.202, "0.00e+0")
# "1.24e+07"
pynumeral.format(0.000123987, "0.000e+0")
# "1.240e-04"
```

1.1.2 Humanize

```
import pynumeral

pynumeral.format(1230974, "0.0a")
# "1.2m"
pynumeral.format(1230974000, "0.0a")
# "1.2b"
pynumeral.format(1230974000000, "0.0a")
# "1.2t"
pynumeral.format(1460, "0 a")
# "1 k"
pynumeral.format(-104000, "0a")
# "-104k"
pynumeral.format(1, "0o")
# "1st"
pynumeral.format(2, "0o")
# "2nd"
pynumeral.format(3, "0o")
# "3rd"
pynumeral.format(100, "0o")
# "100th"
```

1.1.3 Currencies

```
import pynumeral

pynumeral.format(1000.234, "$0,0.00")
# "$1,000.23"
pynumeral.format(1000.2, "0,0[.]00 $")
# "1,000.20 $"
pynumeral.format(-1000.234, "($0,0)")
# "($1,000)"
pynumeral.format(1230974, "($ 0.00 a)")
# "$ 1.23 m"
```


1.1.4 Bytes

```
import pynumeral

pynumeral.format(100, "0b")
# "100B"
pynumeral.format(1024, "0b")
# "1KB"
pynumeral.format(2, "0 ib")
# "2 B"
pynumeral.format(2048, "0 ib")
# "2 KiB"
pynumeral.format(2048000, "0 ib")
# "2 MiB"
pynumeral.format(2048000000, "0 ib")
# "2 GiB"
pynumeral.format(2048000000000, "0 ib")
# "2 TiB"
pynumeral.format(3072, "0.0 b")
# "3.1 KB"
pynumeral.format(3072000, "0.0 b")
# "3.1 MB"
pynumeral.format(3072000000000, "0.0 b")
# "3.1 TB"
pynumeral.format(7884486213, "0.00b")
# "7.88GB"
pynumeral.format(3467479682787, "0.000 ib")
# "3.154 TiB"
```

1.1.5 Percentage

```
import pynumeral

pynumeral.format(1, "0%")
# "100%"
pynumeral.format(0.974878234, "0.000%")
# "97.488%"
pynumeral.format(-0.43, "0 %")
# "-43 %"
pynumeral.format(0.43, "(0.000 %) ")
# "43.000 %"
```

1.1.6 Duration

```
import pynumeral

pynumeral.format(25, "00:00:00")
# "0:00:25"
pynumeral.format(238, "00:00:00")
# "0:03:58"
pynumeral.format(63846, "00:00:00")
# "17:44:06"
pynumeral.format(63846, "HH:MM:SS")
```

(continues on next page)

(continued from previous page)

```
# "17:44:06"  
pynumeral.format(63846, "00:00")  
# "17:44"
```

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`