
numeral-system-py

Release 0.2.0

Oleg Strokachuk

Jan 07, 2020

CONTENTS:

1	numeral-system-py	3
1.1	Example of usage	3
2	Indices and tables	7
	Python Module Index	9
	Index	11

NUMERAL-SYSTEM-PY

Convert from one numeric system to other in python.

Supports following:

- roman numerals
- positional numerals (like binary, arabic (decimal), hexadecimal and others)

1.1 Example of usage

1.1.1 Roman

```
>>> import numeral_system
>>> numeral_system.roman.encode(7)
'VII'
>>> numeral_system.roman.decode('XXII')
22
```

1.1.2 Positional

```
>>> import numeral_system
>>> numeral_system.positional.encode(42, 2)
'101010'
>>> numeral_system.positional.encode(12, 3)
'110'
>>> numeral_system.positional.decode(101, 2)
5
>>> numeral_system.positional.decode('AF', 16)
175
```

Numeral system

Package which contains function for converting between different number system

Roman numeral system

This package contains functions for converting integer into roman and backward

`numeral_system.roman.decode(number)`

Convert string in roman representation to integer number

Parameters `number` (*str*) – roman number as string

Returns integer number

Return type `int`

`numeral_system.roman.encode(number)`

Convert integer from 1 to 3999 to roman number

Parameters `number` (*int*) – integer number

Returns roman number as string

Return type `str`

`numeral_system.roman.is_valid(number)`

Check if number is roman

Parameters `number` (*str*) – string to check

Returns True or False

Return type `bool`

Positional numeral system

Allows convert numbers in different positional numeral system. The most used are binary, octal, decimal and hexadecimal

`numeral_system.positional.decode(number, base, alphabet=('0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z'), sign_literal='-')`

Convert number from given base and alphabet to integer

Parameters

- **number** (*int* | *str*) – given number to convert
- **base** (*int* | *str for less base*) – base of given number
- **alphabet** (*tuple*) – alphabet of numeric system
- **sign_literal** (*str*) –

Returns converted number

Return type `str`

`numeral_system.positional.encode(number, base, alphabet=('0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z'), sign_literal='-')`

Convert integer number to number with given base and alphabet

Parameters

- **number** (*int* | *str*) – given number to convert
- **base** – base of given number
- **base** – int | str for less base
- **alphabet** (*tuple*) – alphabet of numeric system
- **sign_literal** (*str*) –

Returns converted number**Return type** *str*`numeral_system.positional.from_binary(number)`

To hexadecimal number representation from integer

Parameters **number** (*int* | *str*) –**Returns****Return type** *str*`numeral_system.positional.from_hex(number)`

From hexadecimal number representation to integer

Parameters **number** (*str*) –**Returns****Return type** *str*`numeral_system.positional.from_octal(number)`

From hexadecimal number representation to integer

Parameters **number** (*int* | *str*) –**Returns****Return type** *str*

`numeral_system.positional.is_valid(number, base, alphabet=('0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z'), sign_literal='-')`

Check if given number is valid in given base and alphabet

Parameters

- **number** (*int* | *str*) – given number to check
- **base** (*int* | *str for less base*) – base of given number
- **alphabet** (*tuple*) – alphabet of numeric system
- **sign_literal** (*str*) –

Returns True if given number is valid in positional numeral system with base**Return type** *bool*

`numeral_system.positional.max_base(alphabet=('0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z'))`

Get max base for given alphabet

Parameters **alphabet** (*tuple*) –

Returns

Return type int

`numeral_system.positional.to_binary(number)`

To binary number representation from integer

Parameters `number` (`int` | `str`) –

Returns

Return type str

`numeral_system.positional.to_hex(number)`

To hexadecimal number representation from integer

Parameters `number` (`str`) –

Returns

Return type str

`numeral_system.positional.to_octal(number)`

To hexadecimal number representation from integer

Parameters `number` (`int` | `str`) –

Returns

Return type str

Numeral system exceptions

All exceptions which can be thrown by this module during work.

exception `numeral_system.exceptions.IncorrectNumberRepresentationError`

Occurs when representation of number is incorrect according to numeric system

exception `numeral_system.exceptions.NumberOutOfRangeError`

Occurs when number can't be converted to necessary numeric system

exception `numeral_system.exceptions.NumericSystemException`

Base for all exceptions from this module.

exception `numeral_system.exceptions.WrongArgumentTypeError`

Occurs when argument type is wrong

exception `numeral_system.exceptions.WrongArgumentValueError`

Occurs when argument value is wrong

INDICES AND TABLES

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

PYTHON MODULE INDEX

n

- `numeral_system`, 4
- `numeral_system.exceptions`, 6
- `numeral_system.positional`, 4
- `numeral_system.roman`, 4

INDEX

D

`decode()` (*in module numeral_system.positional*), 4
`decode()` (*in module numeral_system.roman*), 4

E

`encode()` (*in module numeral_system.positional*), 4
`encode()` (*in module numeral_system.roman*), 4

F

`from_binary()` (*in module numeral_system.positional*), 5
`from_hex()` (*in module numeral_system.positional*), 5
`from_octal()` (*in module numeral_system.positional*), 5

I

`IncorrectNumberRepresentationError`, 6
`is_valid()` (*in module numeral_system.positional*), 5
`is_valid()` (*in module numeral_system.roman*), 4

M

`max_base()` (*in module numeral_system.positional*), 5

N

`NumberOutOfRangeException`, 6
`numeral_system` (*module*), 4
`numeral_system.exceptions` (*module*), 6
`numeral_system.positional` (*module*), 4
`numeral_system.roman` (*module*), 4
`NumericSystemException`, 6

T

`to_binary()` (*in module numeral_system.positional*), 6
`to_hex()` (*in module numeral_system.positional*), 6
`to_octal()` (*in module numeral_system.positional*), 6

W

`WrongArgumentTypeError`, 6
`WrongArgumentValueError`, 6