
IEC60063 Documentation

Release 0.1

Chintalagiri Shashank

Mar 26, 2021

Contents

1	Usage Example	1
2	Installation	2
3	Source Downloads and Documentation	2
4	License	2
4.1	IEC60063 API	2
5	Indices and tables	4
	Python Module Index	5
	Index	6

This python module provides a generator for the IEC60063 Preferred Values. These are the ‘standard’ values for various types of passive electrical components, such as resistors, capacitors, etc.

For more information about IEC60063, take a look at this [Wikipedia page](#).

1 Usage Example

```
>>> import iec60063
>>> values = iec60063.gen_vals('resistor', 'E12', '1E', '10K')
>>> for value in values:
>>>     print value
```

Note: Though this module provides the preferred values, there is no guarantee that the values generated will actually be something that you can find in a market near you. If this is something you intend to use to calculate values for components that you will physically need, you should first try to determine what ranges of values are easily available from wherever you normally obtain them for each series and component type.

2 Installation

This package can be installed from pypi using pip:

```
$ pip install iec60063
```

Or using easy_install:

```
$ easy_install iec60063
```

3 Source Downloads and Documentation

The simplest way to obtain the source for this package is to clone the git repository:

```
https://github.com/chintal/iec60063
```

You can install it as usual, with:

```
python setup.py install
```

The latest version of the documentation can be found at [ReadTheDocs](#).

4 License

iec60063 is distributed under the LGPLv3 license.

4.1 IEC60063 API

`iec60063.E192 = [Decimal('1.00'), Decimal('1.01'), Decimal('1.02'), Decimal('1.04'), Decimal('1.06'), Decimal('1.08'), Decimal('1.10'), Decimal('1.12'), Decimal('1.15'), Decimal('1.18'), Decimal('1.20'), Decimal('1.25'), Decimal('1.30'), Decimal('1.35'), Decimal('1.40'), Decimal('1.50'), Decimal('1.60'), Decimal('1.70'), Decimal('1.80'), Decimal('2.00'), Decimal('2.20'), Decimal('2.50'), Decimal('2.80'), Decimal('3.15'), Decimal('3.55'), Decimal('4.00'), Decimal('4.50'), Decimal('5.00'), Decimal('5.60'), Decimal('6.30'), Decimal('7.10'), Decimal('8.00'), Decimal('9.00'), Decimal('10.00')]`
The E192 Number Series

`iec60063.E96 = [Decimal('1.00'), Decimal('1.02'), Decimal('1.05'), Decimal('1.07'), Decimal('1.10'), Decimal('1.13'), Decimal('1.16'), Decimal('1.19'), Decimal('1.23'), Decimal('1.27'), Decimal('1.31'), Decimal('1.36'), Decimal('1.41'), Decimal('1.47'), Decimal('1.53'), Decimal('1.60'), Decimal('1.68'), Decimal('1.76'), Decimal('1.85'), Decimal('1.94'), Decimal('2.03'), Decimal('2.14'), Decimal('2.24'), Decimal('2.35'), Decimal('2.47'), Decimal('2.60'), Decimal('2.72'), Decimal('2.85'), Decimal('3.00'), Decimal('3.15'), Decimal('3.30'), Decimal('3.45'), Decimal('3.60'), Decimal('3.75'), Decimal('3.90'), Decimal('4.05'), Decimal('4.20'), Decimal('4.35'), Decimal('4.50'), Decimal('4.75'), Decimal('5.00')]`
The E96 Number Series

`iec60063.E48 = [Decimal('1.00'), Decimal('1.05'), Decimal('1.10'), Decimal('1.15'), Decimal('1.20'), Decimal('1.25'), Decimal('1.30'), Decimal('1.35'), Decimal('1.40'), Decimal('1.50'), Decimal('1.60'), Decimal('1.70'), Decimal('1.80'), Decimal('2.00'), Decimal('2.20'), Decimal('2.50'), Decimal('2.80'), Decimal('3.15'), Decimal('3.55'), Decimal('4.00'), Decimal('4.50'), Decimal('5.00')]`
The E48 Number Series

`iec60063.E24 = [Decimal('1.0'), Decimal('1.1'), Decimal('1.2'), Decimal('1.3'), Decimal('1.5'), Decimal('1.6'), Decimal('1.8'), Decimal('2.0'), Decimal('2.2'), Decimal('2.5'), Decimal('2.8'), Decimal('3.15'), Decimal('3.6'), Decimal('4.0'), Decimal('4.5'), Decimal('5.0')]`
The E24 Number Series

`iec60063.E12 = [Decimal('1.0'), Decimal('1.2'), Decimal('1.5'), Decimal('1.8'), Decimal('2.0'), Decimal('2.2'), Decimal('2.5'), Decimal('2.8'), Decimal('3.15'), Decimal('3.6'), Decimal('4.0'), Decimal('4.5'), Decimal('5.0')]`
The E12 Number Series

`iec60063.E6 = [Decimal('1.0'), Decimal('1.5'), Decimal('2.2'), Decimal('3.3'), Decimal('4.7'), Decimal('7.1'), Decimal('10.0')]`
The E6 Number Series

`iec60063.E3 = [Decimal('1.0'), Decimal('2.2'), Decimal('4.7')]`
The E3 Number Series

`iec60063.cap_ostrs = ['fF', 'pF', 'nF', 'uF', 'mF', 'F']`
Order Strings for Capacitors

`iec60063.res_ostrs = ['m', 'E', 'K', 'M', 'G']`
Order Strings for Resistors

`iec60063.zen_ostrs = ['V']`

Order Strings for Zener Diodes

`iec60063.ind_ostrs = ['nH', 'uH', 'mH']`

Order Strings for Inductors

`iec60063.num_ostrs = ['']`

Order Strings for Numbers

`iec60063.get_ostr(stype=None)`

Given a device type, returns the order strings to be used for that type.

Supports the following types:

- resistor
- capacitor
- zener
- inductor
- number (Default)

Parameters *stype* – The type of component you want the order strings for.

Returns list

`iec60063.get_series(seriesst)`

Given a specific series name, returns the Number series.

Supports the following series:

- E192
- E96
- E48
- E24
- E12
- E6
- E3

Parameters *seriesst* – The IEC60063 series you want.

Returns list

`iec60063.gen_vals(series, ostrs, start=None, end=None)`

Generate values for a specific type over a specific series within an optional range.

The *series* parameter would typically be a string, naming one of the standard IEC60063 series. It could also be an arbitrary list of numbers (`decimal.Decimal` is recommended).

Similarly, the *ostrs* parameter would be a string naming one of the component types supported. You can provide a list of order strings of your own choosing instead.

For each order string in *ostrs*, if a list is provided, or the list of strings returned by `get_ostr()`, this will generate all values of the series between 1 and 999, both included, subject to start / end conditions that may be specified using those parameters.

If the start and / or end is not specified, this function will return all the possible values, given the order strings it has available.

Parameters

- **series** (*str*) – Which IEC60063 series to use. See [get_series\(\)](#).
- **ostrs** (*str*) – What order strings to append to the numbers. See [get_ostr\(\)](#).
- **start** (*str*) – The value to start from. This value must be a part of the series.
- **end** (*str*) – The value to end at. This value must be a part of the series.

Returns A generator which produces all the values within the range.

Return type generator

5 Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Python Module Index

i

iec60063, [2](#)

Index

C

cap_ostrs (*in module iec60063*), 2

E

E12 (*in module iec60063*), 2

E192 (*in module iec60063*), 2

E24 (*in module iec60063*), 2

E3 (*in module iec60063*), 2

E48 (*in module iec60063*), 2

E6 (*in module iec60063*), 2

E96 (*in module iec60063*), 2

G

gen_vals() (*in module iec60063*), 3

get_ostr() (*in module iec60063*), 3

get_series() (*in module iec60063*), 3

I

iec60063 (*module*), 2

ind_ostrs (*in module iec60063*), 3

N

num_ostrs (*in module iec60063*), 3

R

res_ostrs (*in module iec60063*), 2

Z

zen_ostrs (*in module iec60063*), 3