

---

# Sphinx Nameko Theme

*Release 0.0.1*

March 30, 2015



<b>1</b>	<b>Installation and setup</b>	<b>3</b>
<b>2</b>	<b>License</b>	<b>5</b>
<b>3</b>	<b>Changelog</b>	<b>7</b>
3.1	Version 0.0.1 . . . . .	7
<b>4</b>	<b>Theme style</b>	<b>9</b>
4.1	Headings . . . . .	9
4.2	Paragraphs . . . . .	9
4.3	Footnotes . . . . .	10
4.4	Tables . . . . .	10
4.5	Lists . . . . .	10
4.6	Topics . . . . .	11
4.7	Sidebars . . . . .	11
4.8	Admonitions . . . . .	11
4.9	Code . . . . .	12
	<b>Python Module Index</b>	<b>15</b>



A fork of [Sphinx Readable Theme](#) for use in [Nameko](#).



---

## Installation and setup

---

Install from PyPI:

```
$ pip install sphinx-nameko-theme
```

And add this to your `conf.py`:

```
import sphinx_nameko_theme
```

```
html_theme_path = [sphinx_nameko_theme.get_html_theme_path()]  
html_theme = 'nameko'
```





---

### License

---

Sphinx Nameko Theme is licensed under the MIT license.



---

## Changelog

---

### 3.1 Version 0.0.1

First release.



## 4.1 Headings

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

## 4.2 Paragraphs

Duis **aute irure dolor** in *reprehenderit* in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum <sup>1</sup> dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat <sup>2</sup> non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

---

<sup>1</sup> <http://www.google.com>

<sup>2</sup> <http://docutils.sourceforge.net/rst.html>

## 4.3 Footnotes

## 4.4 Tables

Header row, column 1	Header row, column 2
body row 1	Second column of row 1
body row 2	Second column of row 2
body row 3	Second line of paragraph
	Unordered list:
	<ul style="list-style-type: none"><li>• Second column of row 3</li><li>• Second item in bullet list (row 3, column 2)</li></ul>
	Row 4; column 1 will be empty

## 4.5 Lists

### 4.5.1 Unordered list

- Lorem ipsum
- Dolor sit amet
  - Dolor
  - Sit
  - Amet
- Consectetur adipiscing elit

### 4.5.2 Ordered list

1. Lorem ipsum
2. Dolor sit amet
  - (a) Dolor
  - (b) Sit
  - (c) Amet
3. Consectetur adipiscing elit

### 4.5.3 Definition Lists

**Lorem** Lorem ipsum dolor sit amet.

**Ipsum** Ipsum dolor amet sit.

**Dolor** [classifier] Dolor lorem ipsum.

**Sit amet** [classifier one][classifier two] Sit amet consectetur adipiscing elit.

## 4.6 Topics

### Lorem ipsum

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

## 4.7 Sidebars

### Sidebar title

Lorem ipsum dolor sit amet, nunc iaculis nulla molestie neque varius vulputate. Aenean hendrerit condimentum arcu ut porta.

Nullam ac velit lobortis, tincidunt est varius, condimentum lectus. In eu sapien nec lacus interdum ultricies. Aliquam in luctus metus. Nulla mattis, sem quis fringilla pharetra, dui eros consequat velit, at bibendum Mauris sed velit non nibh aliquet mattis.

Mauris sed velit non nibh aliquet mattis. Nullam ac velit lobortis, tincidunt est varius, condimentum lectus. In eu sapien nec lacus interdum ultricies. Aliquam in luctus metus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

## 4.8 Admonitions

### Admonition title

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

**Attention:** Lorem ipsum dolor sit amet, consectetur adipisicing elit.

**Caution:** Lorem ipsum dolor sit amet, consectetur adipisicing elit.

**Danger:** Lorem ipsum dolor sit amet, consectetur adipisicing elit.

**Error:** Lorem ipsum dolor sit amet, consectetur adipisicing elit.

**Hint:** Lorem ipsum dolor sit amet, consectetur adipisicing elit.

**Important:** Lorem ipsum dolor sit amet, consectetur adipisicing elit.

**Note:** Lorem ipsum dolor sit amet, consectetur adipisicing elit.

**See also:**

**Module `zipfile`** Documentation of the `zipfile` standard module.

**GNU tar manual, Basic Tar Format** Documentation for tar archive files, including GNU tar extensions.

---

**Tip:** Lorem ipsum dolor sit amet, consectetur adipisicing elit.

---

**Warning:** Lorem ipsum dolor sit amet, consectetur adipisicing elit.

## 4.9 Code

```
"""An example module docstring to show Pygments style."""

# Some comment.

import datetime
from functools import partial

number = 123
word = 'foo'

class ExampleClass(object):

    """An example class docstring to show Pygments style."""

    def __init__(self, arg1, arg2=None, *args, **kwargs):
        self.attr1 = attr1
        self.attr2 = attr2 or datetime.datetime.now()
        for arg in args:
            print('Argument: {}'.format(arg))
        for k, v in kwargs.iteritems():
            print('Keyword argument named {}: {}'.format(k, v))

    def call_method(self, arg):
        """An example method docstring."""
        if not isinstance(arg, int):
            raise ValueError('Only ints allowed.')
        self.attr1 = arg

    @property
    def example_property(self):
        """An example property docstring."""
        return self.attr1 * 2

def example_function(arg1, arg2=None, *args, **kwargs):
    """An example function docstring to show Pygments style."""
    raise NotImplementedError()
```

### 4.9.1 Autodoc

An example module to show autodoc style.



Contains an example constant, `Storage` class for storing objects and helper function `store_integers()` for storing only integers.

**INT\_CONSTANT = 1**

Example integer constant.

**STR\_CONSTANT = 'string'**

Example integer constant.

**class Storage** (*items=None*)

Bases: `object`

A class for storing objects.

This is an example class to show autodoc style.

It stores a list of objects and saves date of last appended item. Example usage:

```
>>> storage = Storage(['foo', 'bar'])
>>> storage.items
['foo', 'bar']
>>> storage.last_updated
datetime.datetime(2013, 8, 15, 13, 41, 38, 515797)
>>> storage.add_item('baz')
>>> storage.items
['foo', 'bar', 'baz']
>>> storage.last_updated
datetime.datetime(2013, 8, 15, 13, 41, 40, 595544)
```

**Parameters** *items* – Optional list of items to start with.

**items = None**

List of items, add new item using `add_item()`.

**last\_updated = None**

`datetime.datetime` of last item update, will be set to `datetime.datetime.now()` on object instantiation.

**add\_item** (*item*)

Append item to the list.

`last_updated` will be set to `datetime.datetime.now()`.

**Parameters** *item* – Something to append to *items*.

**store\_integers** (*items, allow\_zero=True*)

Store integers from the given list in a storage.

This is an example function to show autodoc style.

Return `Storage` instance with integers from the given list.

Examples:

```
>>> storage = store_integers([1, 'foo', 2, 'bar', 0])
>>> storage.items
[1, 2, 0]
>>> storage = store_integers([1, 'foo', 2, 'bar', 0], allow_zero=False)
>>> storage.items
[1, 2]
```

**Parameters**

- **items** – List of objects of any type, only `int` instances will be stored.
- **allow\_zero** – Boolean – if `False`, 0 integers will be skipped. Defaults to `True`.

**e**

example, [12](#)



## A

`add_item()` (Storage method), 13

Admonitions, 11

Autodoc, 12

## C

Code, 12

## E

`example` (module), 12

## H

Headings, 9

## I

`INT_CONSTANT` (in module `example`), 13

`items` (Storage attribute), 13

## L

`last_updated` (Storage attribute), 13

Lists, 10

## P

Paragraphs, 9

## S

`Storage` (class in `example`), 13

`store_integers()` (in module `example`), 13

`STR_CONSTANT` (in module `example`), 13

## T

Tables, 10

Topics, 10