



JoKneeMo

# Sphinx Watermark User Manual

Version

Version 2.0.0

# Sphinx Watermark

License **GPL v3**

 maintainability **A**

status **package or version not found**

**pypi** **package or version not found**

`sphinx-watermark` is an extension for Sphinx that enables watermarks for HTML output.

Full documentation: <https://jokneemo.github.io/sphinx-watermark>

# Why a Fork?

Forked from [kallimachos/sphinxmark](#)

This fork was created primarily to remove the dependency on bottle, and to support updates to Sphinx v7, Docutils, and Pillow.

Some themes perform differently in newer version of Docutils. The main issues faced are html elements are changed between div, section, article, etc.

Sphinxmark only supported div elements and the configuration changes I made to support it were too expansive for a simple pull request. See below for all of the enhancements added.

## What's Different?

- Removed bottle dependency
- HTML element selection
- Static png name to support spaces in text
- Collection of fonts
  - See the [fonts directory](#).
- Customizable border

## Installation

Install sphinx-watermark using pip:

```
$ pip3 install sphinx-watermark
```

**Note**

Sphinx Watermark uses the Pillow module for creating PNG files. If you encounter `C module is not installed` errors when Sphinx loads the sphinx-watermark extension, you may need to install some of the external libraries for Pillow.

## Usage

Configure sphinx-watermark in your `conf.py` file.

1. Add sphinx-watermark to the list of extensions:

```
extensions = ['sphinx_watermark']
```

2. If you are using a custom image file, specify its directory relative to the `conf.py` file. If no value is given, the path defaults to `_static`.

```
html_static_path = ['_static']
```

3. Enable and configure sphinx-watermark in `conf.py`. You only have to include the values you want to change from default.

Below is the bare-minimum config for an `Internal\nDraft` watermark:

```
watermark = {
    'enabled': True,
    'text': 'Internal\nDraft'
}
```

Below are the defaults for all options: (Notice that watermarks are disabled by default.)

```
watermark = {
    'enabled': False,
    'selector': {
        'type': 'div',
        'class': 'body'
    },
    'position': {
        'margin': None,
        'repeat': True,
    }
}
```

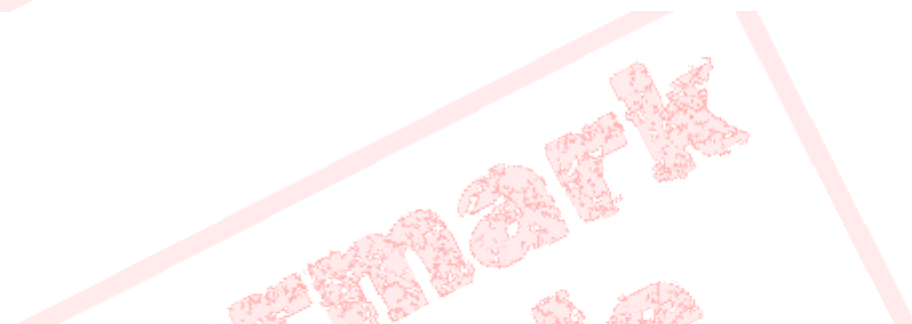
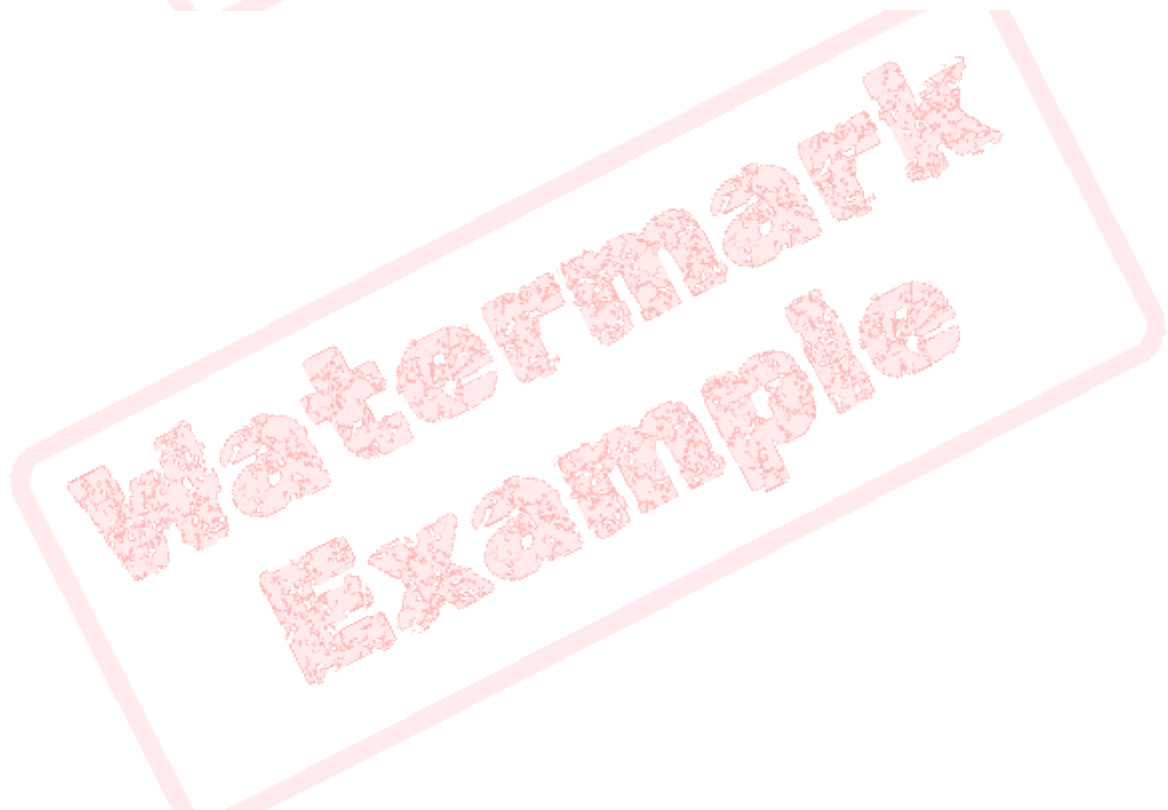
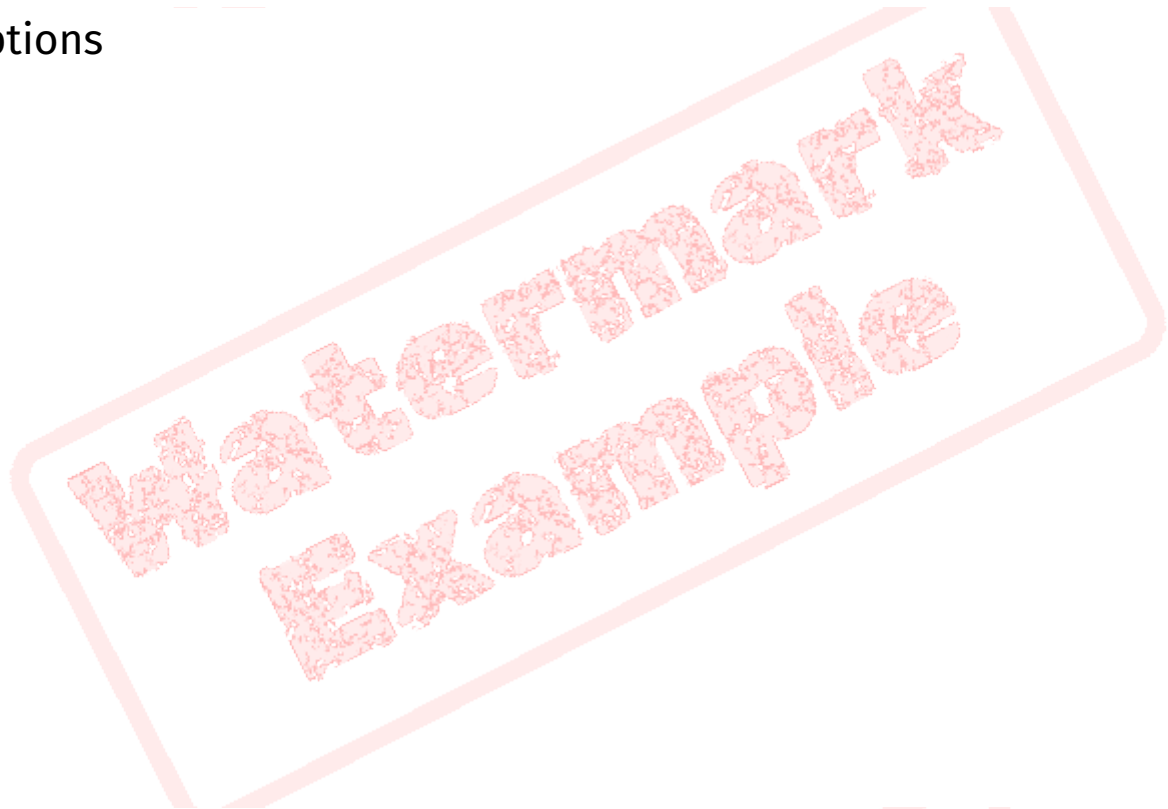
```
    'fixed': False
},
'image': None,
'text': {
    'content': None,
    'align': 'center',
    'font': 'RubikDistressed',
    'color': (255, 0, 0),
    'opacity': 40,
    'size': 100,
    'rotation': 0,
    'width': 816,
    'spacing': 400,
    'border': {
        'outline': (255, 0, 0),
        'fill': None,
        'width': 10,
        'padding': 30,
        'radius': 20,
    }
}
```

4. Build your docs as normal. The defined watermark should appear behind the text.

**Note**

Some Sphinx themes place body content in different CSS divs. See the [sphinx-watermark documentation](#) for full configuration details.

## Options





**enabled (bool):**

- `True` - enable watermarks
- `False` - disable watermarks
- Default: `False`

```
watermark = {
    'enabled': False
}
```

**selector (dict or string):**

Short form is a string, setting the class name of a div element.

Long form is a dictionary, specifying an html element type and it's class.

**type (string):**

HTML element type to apply the watermark to.

- Default: `div`

**class (string):**

CSS class, applied to the type, where watermark is displayed

- Default: `body`
- Examples for common themes:

sphinx\_rtd\_theme ->

```
'class': 'document'
```

openstackdocstheme ->

```
'class': 'docs-body'
```

sphinx\_book\_theme ->

```
'class': 'row#main-content'
```

sphinx\_immaterial ->

```
'class': 'md-main__inner'
```

```
watermark = {
    'selector': {
        'type': 'div',
        'class': 'body'
    }
}
```

```
}  
}
```

position (dict):



margin (string):

- `left` - place watermark in the left





page margin

**repeat (bool):**

**image (string):**

- **right** - place watermark on right page margin

- Default: **None**

- **True** - image repeats down the page

- **False** - image appears once at top of page

- Default: **True**

- **True** - watermark does not scroll with content

- **False** - watermark scrolls with content

- When set to **True**, this option centers the watermark in the viewport,

not the element specified by **selector**.

- image file name in **html\_static\_path** directory to use as

watermark

**text (dict or string):**

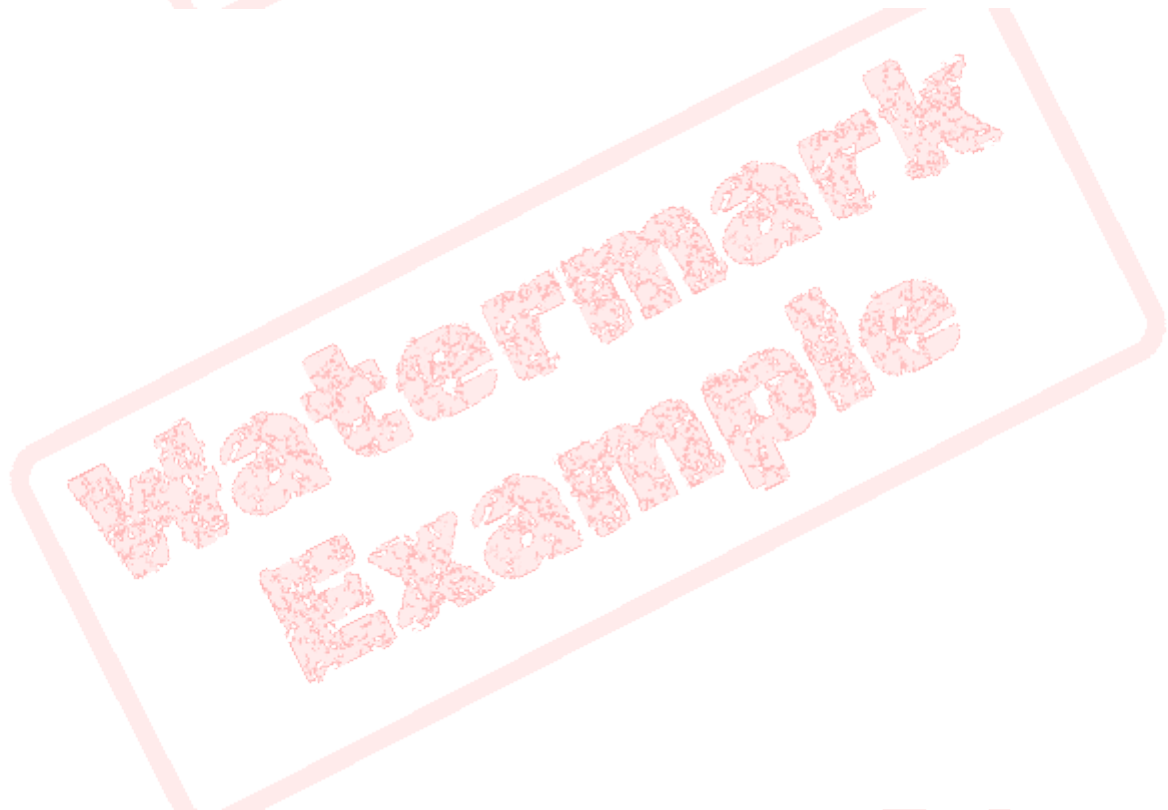
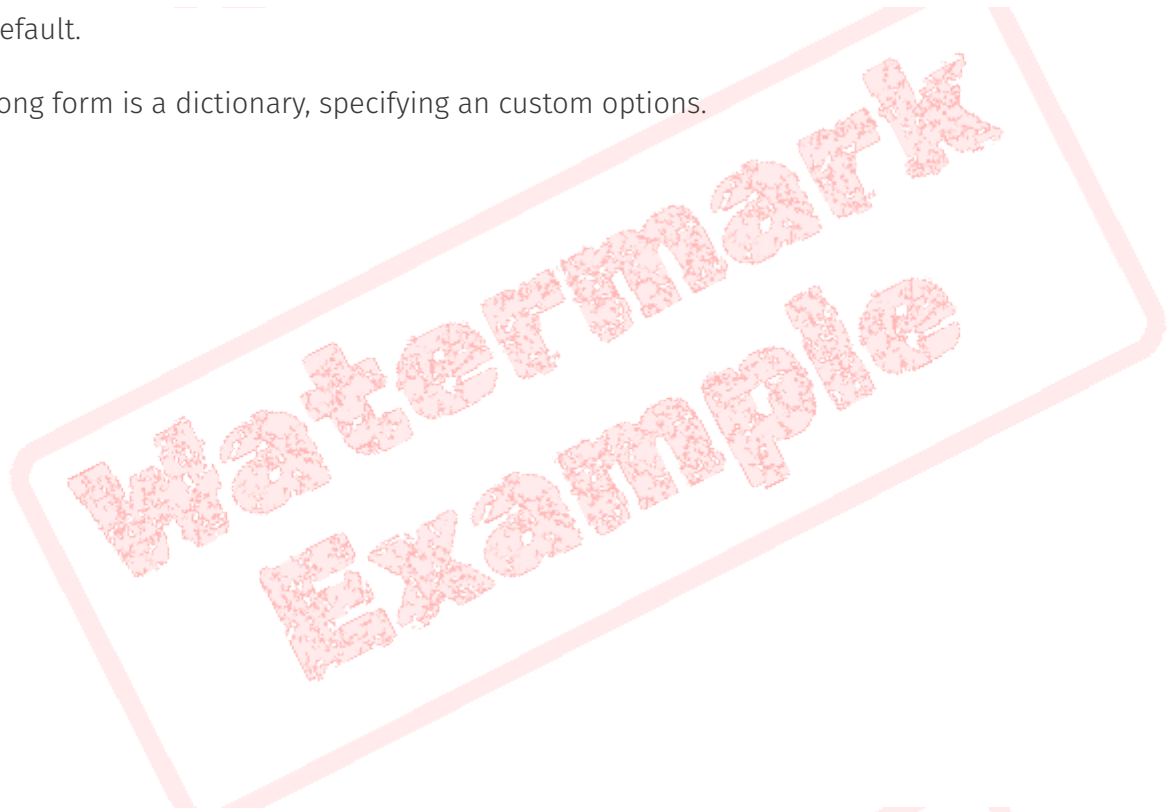
- Should not be used if text.content is set.
- Default: `None`

```
watermark = {  
    'image': None  
}
```



default.

Long form is a dictionary, specifying an custom options.



**content (string):** Text to insert into the watermark.

Use `\n` for multiline text.

- Default: `None`

**align (string)\*\*:**

- `left` - Left justify the text.content
- `right` - Right justify the text.content
- `center` - Center justify the text.content
- Default: `center`

**font (string):**

Which font should be used to render the text.

- Anton
- Gluten
- KodeMono
- RobotoMono
- RubikDistressed
- ShantellSans
- Default: `RubikDistressed`

**color (int tuple):**

RGB Value in as a tuple of integers

- Default: `(255, 0, 0)` (Red)

**opacity (int):**

RGB Alpha, or how transparent should the watermark be

- Default: `40`

**size (int):**

Font size, in pixels, for the watermark text.content

- Default: `100`

**rotation (int):**

Degrees of rotation for the entire

watermark.

**width (int):**

- Default: `0`

Width, in pixels, of transparent box containing the watermark text.

If a large `text.size` is specified, or the `text.content` string is long, you may need to set this option to a number higher than the default for the entire string to display.

Note that the text will be cut off at the edge of the containing CSS div regardless of the width setting.

- Default: `816`

Spacing, in pixels, between text watermarks when repeated, and the text line height

- Default: `400`



image.

**outline**  
**(int**  
**tuple):** RGB Value in as a tuple  
of integers

- Default: `(255, 0, 0)` (Red)

**fill** **(int**  
**tuple):** RGB Value in as a tuple  
of integers to fill the  
watermark background.

- Default: `None`

**width**  
**(int):** How wide, in pixels, the  
border should be.

- Default: `10`

**padding**  
**(int):** How much spacing, in  
pixels, should be  
between `text.content`  
and the border.

- Default: `30`

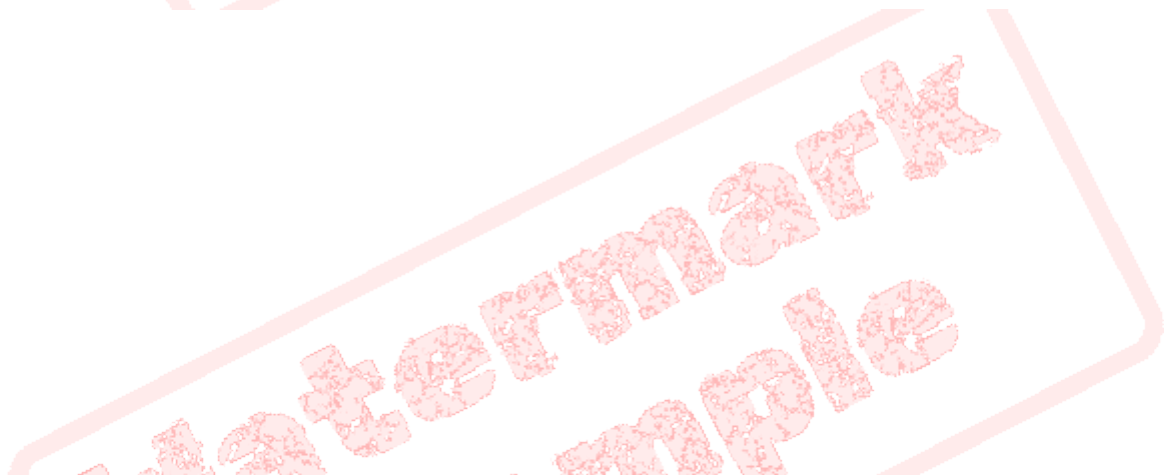
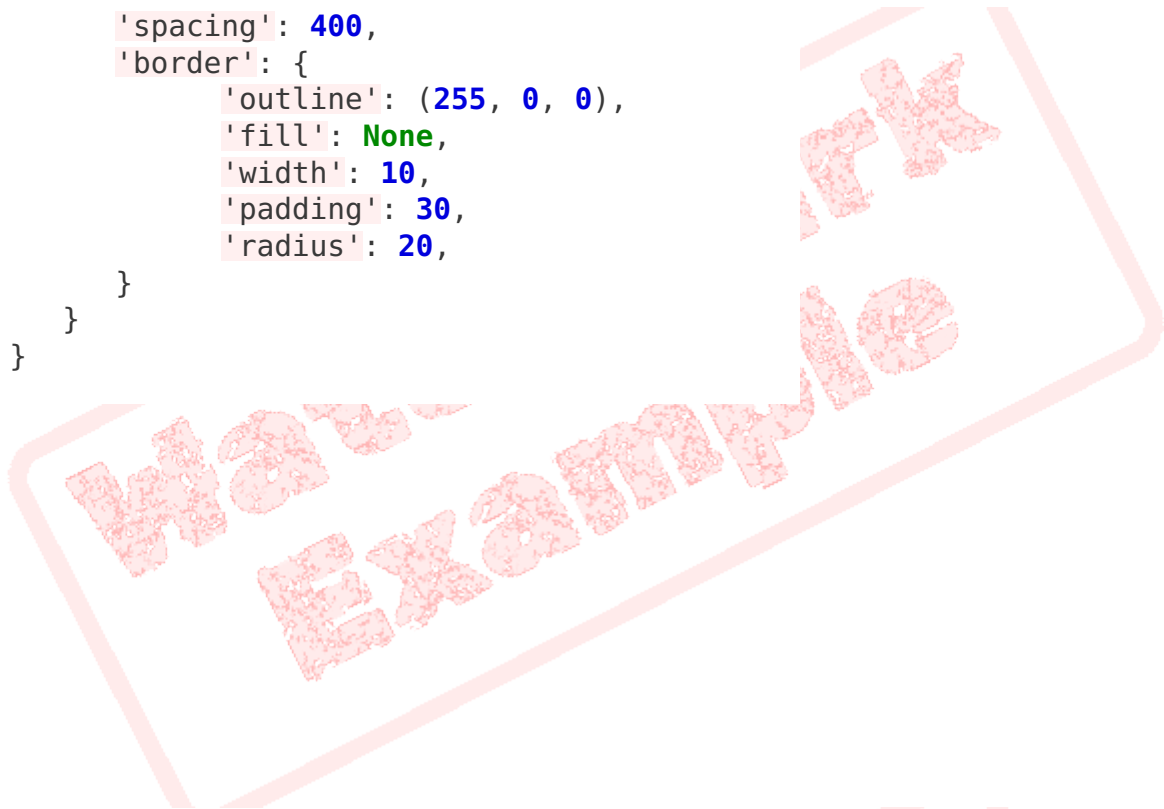
**radius**  
**(int):** How round, in pixels,  
the corners of the  
border should be.

- Default: `20`

```
watermark = {  
    'text': {  
        'content': None,  
        'align': 'center',  
        'font': 'RubikDistressed',  
        'color': (255, 0, 0),  
        'opacity': 40,  
        'size': 100,  
        'rotation': 0,  
        'width': 816,
```



```
'spacing': 400,  
'border': {  
    'outline': (255, 0, 0),  
    'fill': None,  
    'width': 10,  
    'padding': 30,  
    'radius': 20,  
}  
}
```



## Troubleshooting

You can enable debugging output for sphinx-watermark by raising the output verbosity level  $\geq 2$ . Pass the `-vv` option on the command line.

Example:

```
$ sphinx-build -vv -b html sourcedir buildir
```

## Release Notes

### Release 2.0.0

- Forked to v2.0.0 as the config syntax has fully changed
- New configuration syntax as a python dict
- Removed bottle dependency
- Added HTML element selection
- Use a static name for the watermark image to support spaces in text
- Added additional fonts with configuration
- Customizable border options
- Default image size to fit US Letter PDF pages

### Release 1.0.0

- Move to 1.0.0 stable release.
- [#39](#): Replace Travis-CI with GitHub Actions.

### Release 0.2.1

- [#38](#): use pathlib in place of os.path.
- [#37](#): add type annotations.

- #36: update option documentation to include `sphinx_book_theme`. Thanks rscohn2.
- #35: fix text watermark opacity issue. Thanks chad-iris.

## Release 0.2.0

- #34: add support for Sphinx 2.0. Remove support for Python 2.7.

## Release 0.1.19

- #29: fix isort test error.
- #28: remove invalid argument to `ImageDraw.text()` method.

## Release 0.1.18

- #26: enable `parallel_read_safe`.
- Remove unused `publish.sh` script.
- Tidy `tox.ini` file.

## Release 0.1.17

- #23: remove Pillow and Bottle version requirements.
- #22: add `sphinxmark_text_width` option and increase default width.
- Add Release Notes and restructure documentation.

## Code Listing

A Sphinx extension that enables watermarks for HTML output.

Project Source: <https://github.com/JoKneeMo/sphinx-watermark>

Copyright 2024 - JoKneeMo

Licensed under GNU General Public License version 3 (GPLv3).

Commit 0762fdef2eabead5edf99e393becc2cd5a926f11 and older are licensed under Apache License, Version 2.0 and copyrighten 2021 by Brian Moss

Original project: <https://github.com/kallimachos/sphinxmark>

`__init__.createimage(app: Sphinx, srcdir: Path, buildpath: Path) → str`

Create PNG image from string.

`__init__.deep_update(source, overrides)`

Update a nested dictionary or similar mapping. Modify `source` in place.

`__init__.generate_css(app: Sphinx, buildpath: str, imagefile: str) → str`

Create CSS file.

`__init__.generate_watermark(app: Sphinx, env: BuildEnvironment) → None`

Generate watermark

`__init__.get_image(app: Sphinx) → tuple`

Get image file.

`__init__.setup(app: Sphinx) → dict`

Configure setup for Sphinx extension.

#### Parameters:

`app` – Sphinx application context.

