

# NtDocutils

0.1.0

👤 Miguel Angel Rivera Notararigo

✉ [ntrrgx@gmail.com](mailto:ntrrgx@gmail.com)

🔗 0.1.0

🕒 2017-04-06 18:35:00 -04:00

© MIT


# NtDocutils

**NtDocutils** is a **Material Design Lite** theme for **Docutils** (maybe a little more 😊). This site was built with it, so you may see what you can do, but if you want to see all the styles, you should go to the [demo site](#).

## Contents

- Features
- Installation
  - From PyPI
- Usage
  - OS filtering
  - Attachments
  - Theme customization
  - Command line reference
- Uninstallation
- Attributions and references

## Features

- **All features** from **Docutils** for the `rst2html5.py` writer.
- Special roles for emojis and keyboard keys (see [this section](#)).
- Filter content by OS and distributions (see [this section](#)).
- Download the attachments (see [this section](#)), print the article and more from the  button.
- Theme customization (see [this section](#)).

# Installation

---

NtDocutils requires:

- **Python** 3.4 or above
- **Docutils** 0.13.1 (*autoinstalled*)
- **Pygments** 2.2.0 (*autoinstalled*)

## From PyPI

```
pip install NtDocutils==0.1.0
```

# Usage

---

Basically, you have to do two things:

1. Create a `.rst` file:

`example.rst`:

```
1 =====
2 My Article
3 =====
4
5 :Author: Vultur Gryphus
6 :Contact: info@vultur.org.ve
7
8 Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
9 tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
10 quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
11 consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse
12 cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non
13 proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
```

2. Process your file:

```
ntdocutils example.rst example.html
```

And that's it, you already have some like this:



The following sections cover the usage of some utilities to improve your article and at the end you can see the [command line reference](#).

## OS filtering

You can create content for every OS just adding the `os os-{{ OS name }}` classes in the elements you want filter, **NtDocutils** will create the buttons dynamically at the left bottom corner and set the first OS that it finds as default.  
E.g:

```
1 .. class:: os os-debian
2
3 **Text for Debian**
4
5 .. class:: os os-windows
6
7 **Text for Windows**
```


Will result in:

**Text for Windows**

And creates the following buttons:




# Attachments

You can set a list of useful files in the  button by adding the following lines in your article:

```
1 .. raw:: html
2
3     <script>
4         ATTACHMENTS = [
5             {
6                 "url": "NtDocutils.rst",
7                 "name": "NtDocutils 0.1.0.rst",
8                 "icon": "code"
9             }
10        ];
11    </script>
```

## Syntax



**ATTACHMENTS** is a global array, it store the files list showed in  button, each file must be defined as an object with the following properties:

**url**(String)

URL to the file, could be an external link.

**name**(String)

**Optional.** Name displayed at the list, also overwrites the attachment name.

**icon**(String)

**Optional.** **Material icon** displayed at the list, by default **NtDocutils** uses an icon related to the file extension.

```
1 .. raw:: html
2
3     <script>
4         ATTACHMENTS = [
5             {
6                 "url": URL,
7                 "name": DISPLAY_NAME,
8                 "icon": ICON_NAME
9             },
10            ...
11            {
12                "url": URL_N,
13                "name": DISPLAY_NAME_N,
14                "icon": ICON_NAME_N
15            }
16        ];
17    </script>
```

# Theme customization

You can use the **customize tool** from the **MDL** site to get a custom `.css` with your preferred colors, after that, you must setup some styles by creating a file with the following **template**:

`customize.css`

```
1 /* Ribbon */
2
3     .ribbon {
4         background-color: {{ Primary color }};
5     }
6
7 /* ... */

83 /* ... */
84
85 /* Links */
86
87     /*a {
88         color: {{ Accent color }};
89     }*/
90
91 /* ... */
```

The recommended color for the ribbon background (line 4) is the primary color from the theme, you can get this value searching the property `color` at the rule `.mdl-button.mdl-button--colored` in the file you downloaded from **MDL** (`material.min.css`). The links (line 88) uses the accent color from the theme, but in some cases this make them a little unreadable, so you could uncomment it and use the primary color. You can feel free editing the others rules, but usually they will be fine with that values. When you are ready, you have to run **NtDocutils** with the following option:

```
ntdocutils --stylesheet=path/to/material.min.css,path/to/customize.css \
source.rst destination.html
```

## Command line reference

```
ntdocutils [-h] [-V] [-S SERVER] SOURCE DESTINATION
```

`-h, --help`

Shows the help message.

`-V, --version`

Shows the **NtDocutils** version.

`-S SERVER, --server SERVER`

Server where assets are hosted. If `file` is passed as value, it will activate the offline mode, this will create a directory with the theme name in the `DESTINATION` parent folder and stores the necessary assets in there.

## Note



All the options from the `rst2html.py` writer are available.

# Uninstallation

Should be enough with this 😊:

```
pip uninstall NtDocutils
```

# Attributions and references

---

**NtDocutils** uses:

- **reStructuredText** and **Docutils**
- **Material Design Lite**
- **Roboto**
- **Material icons**

**Docutils Team.** *reStructuredText*. <http://docutils.sourceforge.net/rst.html>

**Mozilla Developer Network.** *JavaScript | MDN*. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>