POLITECNICO
MILANO 1863

Sphinx tutorial

Lorenzo Amici
Politecnico di Milano

22/10/2020

Introduction - Sphinx

Sphinx is a documentation generator, i.e. a tool that can translate plain text source files into various output formats, automatically producing cross-references, indices and other features.

Sphinx’s whole potential comes out when this tool is paired with the richness of its default plain-text markup format, reStructuredText, along with its significant extensibility capabilities.

The root directory of a Sphinx collection of plain-text documents is called the source directory. This directory also contains the Sphinx configuration file `conf.py`, where you can configure all aspects of how Sphinx reads your sources and builds your documentation.

Tools

The main tools we will use for this tutorial are:

- ✓ Anaconda: in order to create a virtual environment and use its command line
- ✓ A text editor (in my case Sublime Text): in order to write reStructuredText files
- ✓ Sphinx: in order to generate documentation from the reStructuredText files
- ✓ GitHub: world’s most common version control platform
- ✓ ReadTheDocs.org: in order to host the documentation online

Create a virtual environment - 1

In order to create a virtual environment in Anaconda, we need to open the Anaconda Prompt (one of the programs installed with the Anaconda package):

- ✓ Navigate into the folder for the project
- ✓ Run the command `conda create --name <environment name>`
- ✓ Press «y» to proceed
- ✓ This creates a new environment with the assigned name

Create a virtual environment - 2

We need to activate our environment: to do so, run the command `conda activate <environment name>`. The name in the parentheses should change from (base) to (<environment name>).

Now we have to install Sphinx in our environment:

- ✓ Run the command `conda install -n <environment name> sphinx`
- ✓ This will install all the necessary packages to use Sphinx, and Sphinx itself

Create a GitHub repo

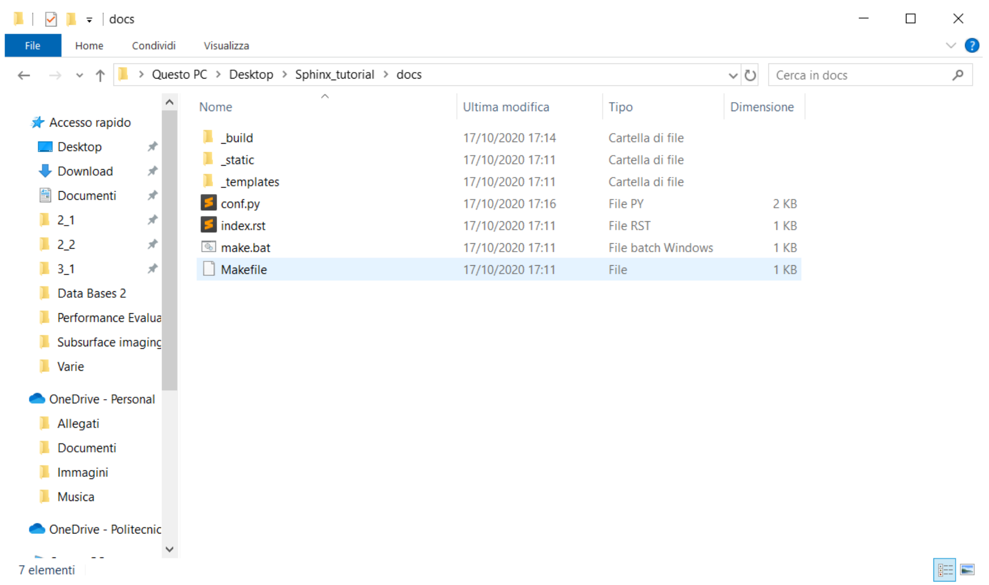
Since we will publish our code on GitHub, we need to create a GitHub repository:

- ✓ Go on your GitHub profile page
- ✓ Click on ‘Repositories’
- ✓ Click on ‘New’
- ✓ Choose a name and create the repository

- ✓ In Anaconda Prompt, run `git init`
- ✓ Then `git remote add origin https://github.com/<your GitHub name>/<your repository name>.git`
- ✓ In this way the local folder and the remote repository will be connected

First steps with Sphinx

Sphinx provides a useful command to get started with a prototype documentation: run `sphinx-quickstart` in the Anaconda Prompt and press ‘Enter’ to accept the default value of the questions the terminal will ask (you can name the project and the author as you wish). This will create a source directory with a `conf.py` file and a master document, `index.rst`.



Nome	Ultima modifica	Tipo	Dimensione
_build	17/10/2020 17:14	Cartella di file	
_static	17/10/2020 17:11	Cartella di file	
_templates	17/10/2020 17:11	Cartella di file	
conf.py	17/10/2020 17:16	File PY	2 KB
index.rst	17/10/2020 17:11	File RST	1 KB
make.bat	17/10/2020 17:11	File batch Windows	1 KB
Makefile	17/10/2020 17:11	File	1 KB

An example of source directory right after the `sphinx-quickstart` command

- The `conf.py` file is the main configuration file for our documentation. It is executed as a Python source file and it is used to assign configuration values (e.g. setting information about the documentation, importing extensions…)

- The `index.rst` file is intended to work as the first page shown in the html output and as a container for the ‘table of content tree’ or ‘toctree’, that is the way of rst to connect multiple files in a single hierarchy of documents

Pushing on GitHub

We can now start pushing our code on GitHub. To do so, run the following commands:

- ✓ `git add .`
- ✓ `git commit -m ”first commit”`
- ✓ `git push origin master`

Now our GitHub repository should contain the files created by the `sphinx-quickstart` command.

Writing and building the documentation

Let's start editing the documentation. Open the `index.rst` file in a text editor of your choice (I will use Sublime).

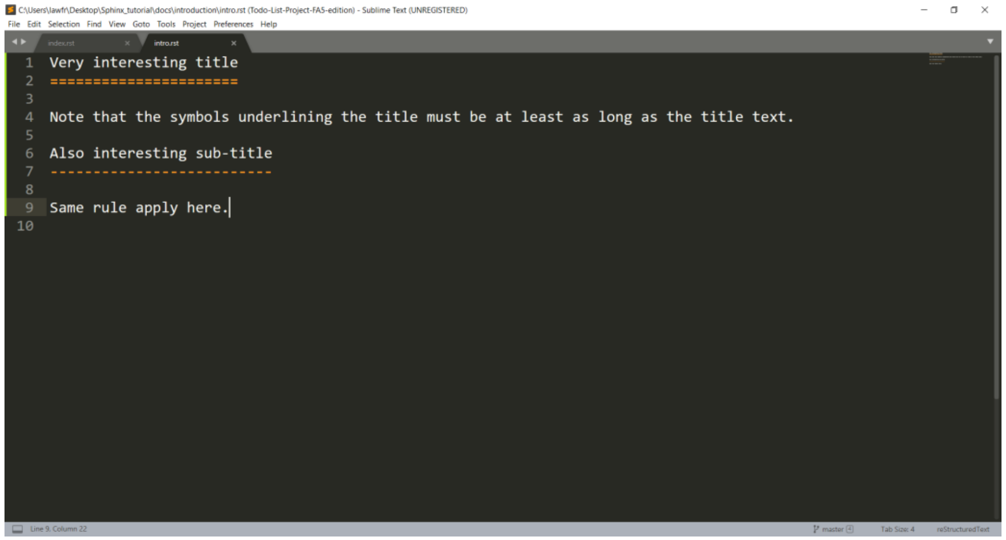
We can edit the content of the documentation by directly writing plain text in this file: try writing 'lorem' and hitting Tab. This will add a new paragraph with some Lorem Ipsum text in the first page of our documentation.

To see these changes in action, let's build the html version of our documentation: if we used the quickstart functionality, Sphinx allows us to do so simply executing the `make html` command. If we now navigate to _build\html, we can open the `index.html` file in the browser and see our documentation.

Writing the documentation - 2

We will now create a new section in our documentation. Inside your main documentation folder, create a new folder called ‘introduction’ (you can use the command `mkdir introduction`). Now navigate into that folder and create a file called `intro.rst`; this will be the file we will edit with the content to add.

Open the file in the text editor and start by adding a title: titles and sub-titles in rst are identified by a symbol underlining them. You can decide your own hierarchy, in my case the ‘=‘ identifies the titles and the ‘-’ identifies the sub-titles.

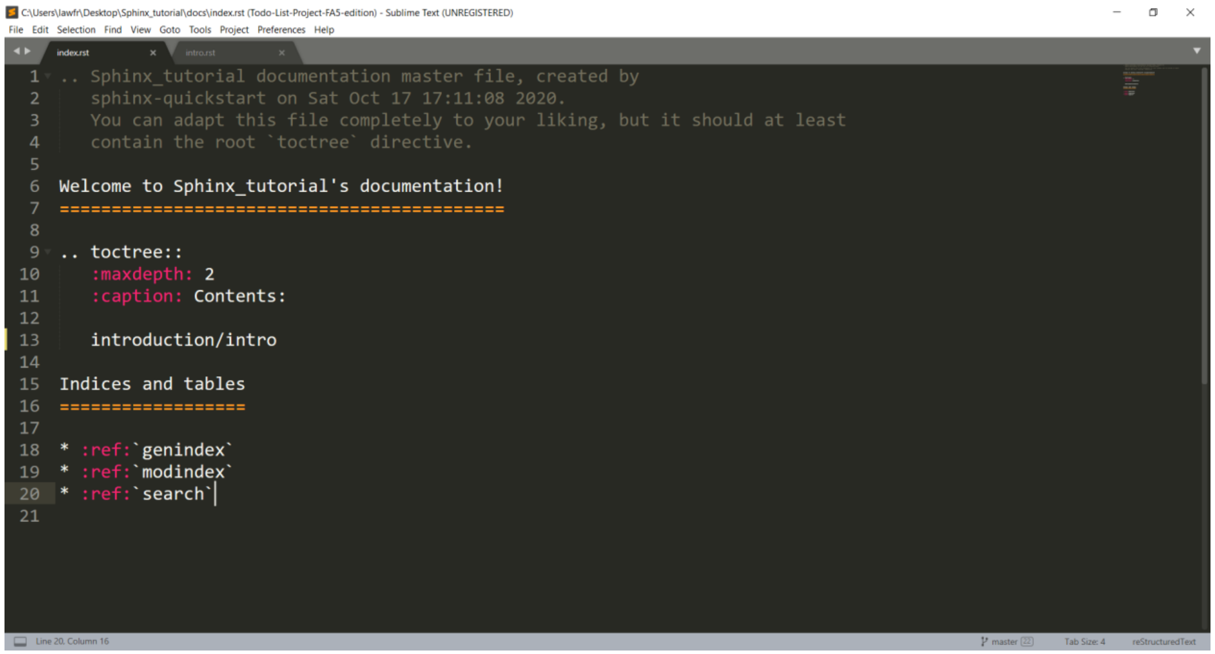

C:\Users\lawfr\Desktop\Sphinx_tutorial\docs\introduction\intro.rst (Todo-List-Project-FA5-edition) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
index.rst
intro.rst

```
Very interesting title
======================

Note that the symbols underlining the title must be at least as long as the title text.

Also interesting sub-title
--------------------------

Same rule apply here.
```

Line 9, Column 22
master
Tab Size: 4
reStructuredText

Documentation structure

In order to add the newly created section to the documentation, we need to edit the toctree in `index.rst`. We need to add the reference to the `intro.rst` file in the introduction folder:


C:\Users\lawfr\Desktop\Sphinx_tutorial\docs\index.rst (Todo-List-Project-FA5-edition) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
index.rst
intro.rst
Line 20, Column 16
master
Tab Size: 4
reStructuredText

```
.. Sphinx_tutorial documentation master file, created by
   sphinx-quickstart on Sat Oct 17 17:11:08 2020.
   You can adapt this file completely to your liking, but it should at least
   contain the root `toctree` directive.

Welcome to Sphinx_tutorial's documentation!
===========================================

.. toctree::
   :maxdepth: 2
   :caption: Contents:

   introduction/intro

Indices and tables
==================

* :ref:`genindex`
* :ref:`modindex`
* :ref:`search`
```

If we build again the html, it should contain an additional page with the content we just added.

Adding features

The main features of the rst markup language that we can add are:

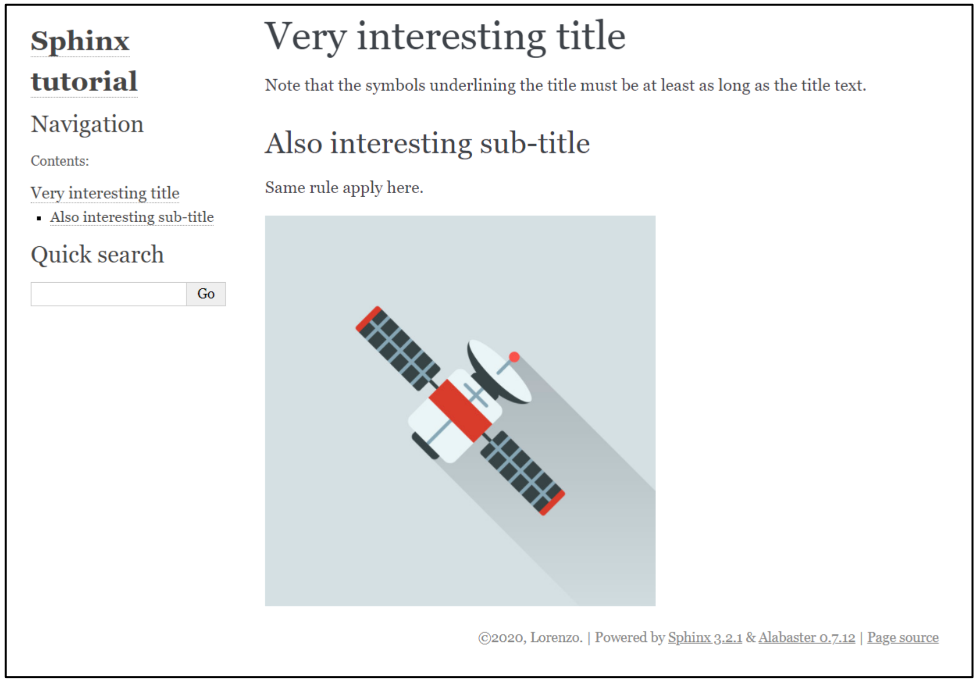
- ✓ Images
- ✓ Figures (images but with a caption)
- ✓ Bullet points
- ✓ Note boxes
- ✓ Internal and external links
- ✓ Text modification in the form of inline markup (bold, italic…)
- ✓ Raw html content

Adding features - images

To add an image in the documentation, we need to use the following syntax:

```
.. image:: path/to/image.png
   :width: 400px
   :alt: alternate text
   :scale: 50%
```

`.. image::` is called a directive in reStructuredText



Sphinx tutorial

Navigation

Contents:

Very interesting title

- Also interesting sub-title

Quick search

Very interesting title

Note that the symbols underlining the title must be at least as long as the title text.

Also interesting sub-title

Same rule apply here.

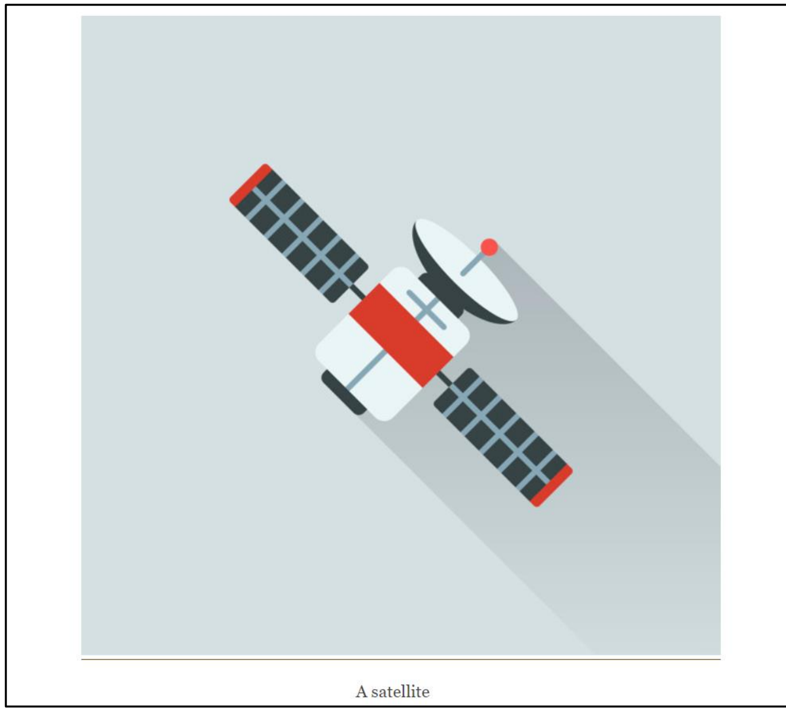
©2020, Lorenzo. | Powered by Sphinx 3.2.1 & Alabaster 0.7.12 | Page source

It's useful to create an images folder that contains all the images of our documentation.

Adding features - figures

Figures in rst are images with a caption. To add a figure in the documentation, we need to use the following syntax:

```
.. figure:: path/to/image.png
   :width: 400px
   :alt: alternate text
   :scale: 50%

   Here goes the caption of the figure
```

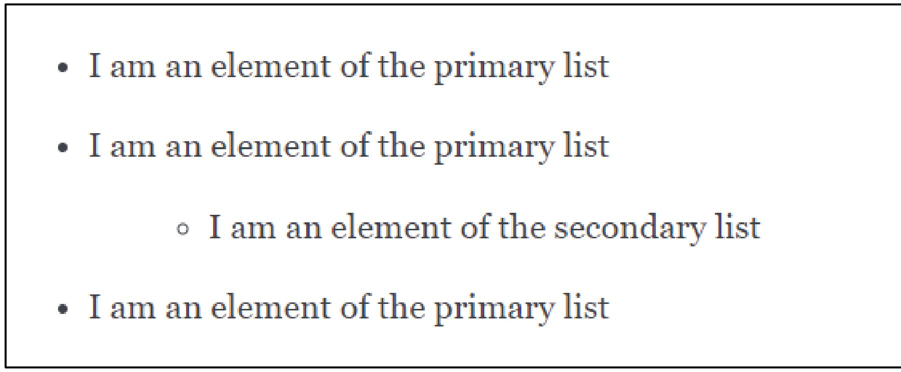

A satellite

Adding features – bullet points

To create bullet point lists, you can use a variety of symbols: ‘+’, ‘*’, ‘-’ ecc…

Nested bullet points are obtained indenting the code in the proper way:

```
+ I am an element of the primary list
+ I am an element of the primary list

   - I am an element of the secondary list

+ I am an element of the primary list
```

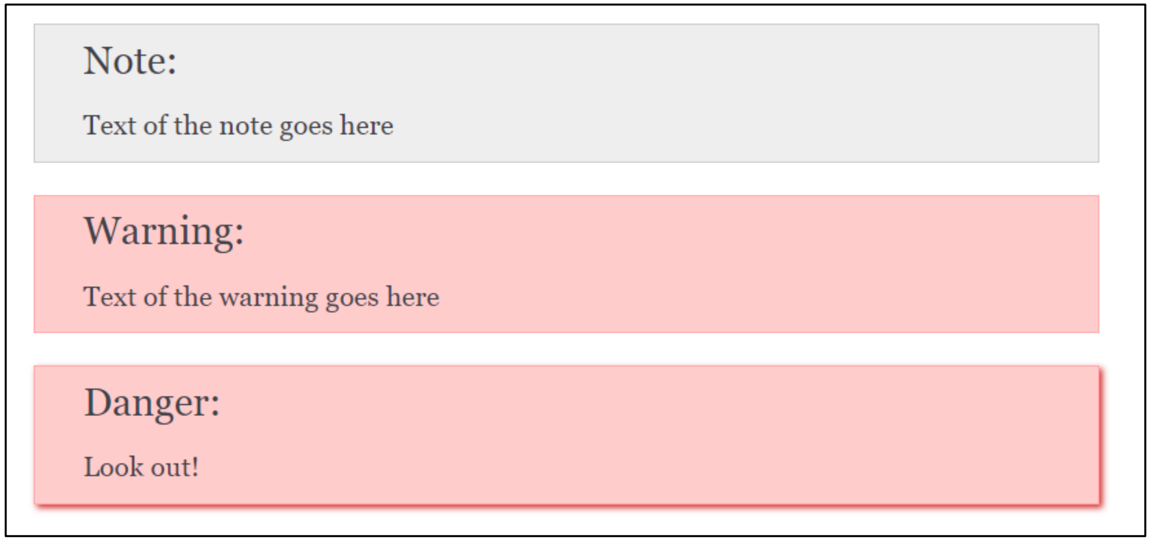


- I am an element of the primary list
- I am an element of the primary list
    - I am an element of the secondary list
- I am an element of the primary list

Adding features – annotations

There are many types of annotation boxes: they mainly differ for color. Here are some examples:

```
.. note:: Text of the note goes here

.. warning:: Text of the warning goes here

.. danger:: Look out!
```



Note:

Text of the note goes here

Warning:

Text of the warning goes here

Danger:

Look out!

Adding features – links

We can add both external and internal links. The first ones are included in this way:

```
`external link <http://www.geolab.polimi.it/>`_
```

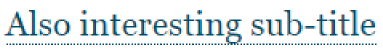
The internal links connect parts of our documentation with one another. We need first to assign a name to the part of the documentation to which we want to redirect:

```
.. _subtitle:
```

Then we can link any other part of the documentation with that section:

```
:ref:`subtitle`
```

$\longrightarrow$

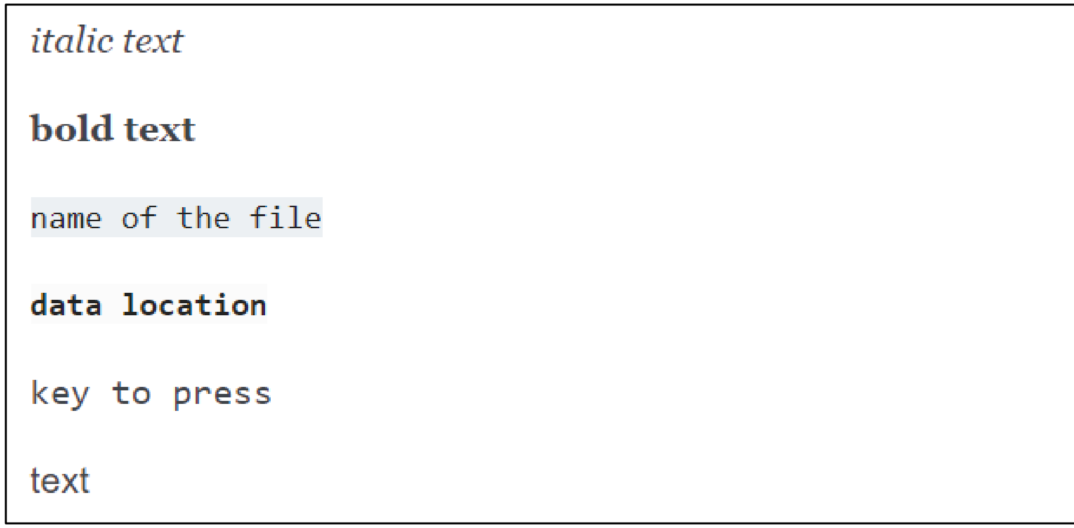


Also interesting sub-title

Adding features – inline markup

We can modify the text in many ways using inline markups:

- ✓ `*italic*` *italic*
- ✓ `**bold**` **bold**
- ✓ `` :file:`name of the file` ``
- ✓ `` :data:`data location` ``
- ✓ `` :kbd:`key to press` ``
- ✓ `` :guilabel:`text` ``


italic text

bold text

`name of the file`

`data location`

`key to press`

text

In this case :guilabel: seems not doing anything but we will see the result when we change theme

Adding features – html content

We can also add raw html content to our documentation. In this case we will add a GoogleMaps map:

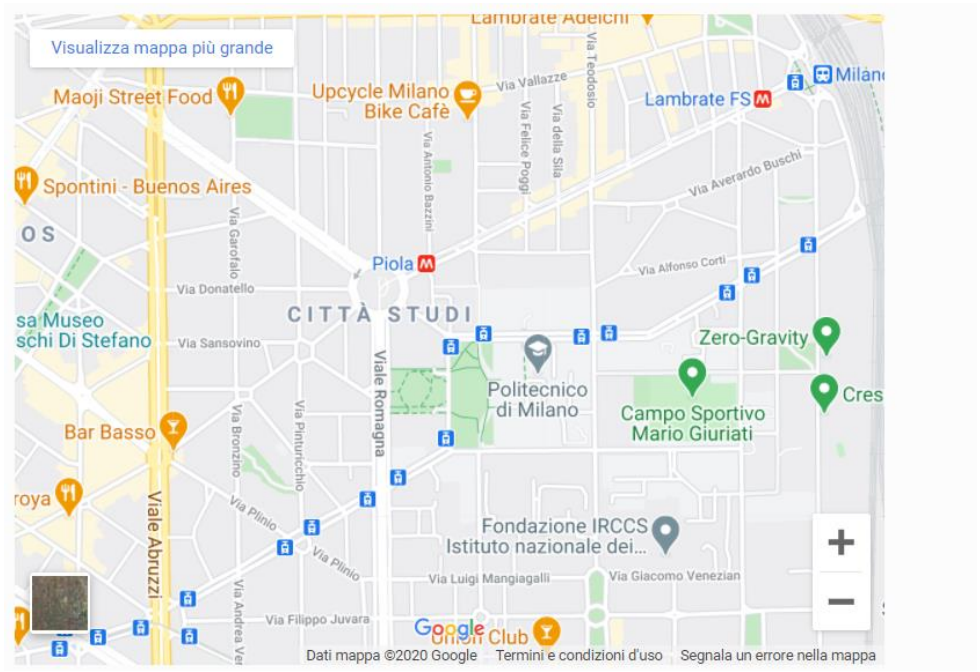
✓ add in the code:

```
.. raw:: html

    <iframe
    src="https://www.google.com/maps/embed?pb=!1m14!1m12!1m3!1d6085.36089
    2068745!2d9.22657130605 4593!3d45.479461266190164!2m3!1f0!2f0!3f0!3m2!
    1i1024!2i768!4f13.1!5e0!3m2!1sit!2sit!4v1603108894032!5m2!1sit!2sit"
    width="600" height="450" frameborder="0" style="border:0;«
    allowfullscreen="" aria-hidden="false" tabindex="0"></iframe>
```

This iframe was obtained in GoogleMaps and can be embedded in our web documentation

Adding features – html content

The result should look like this:

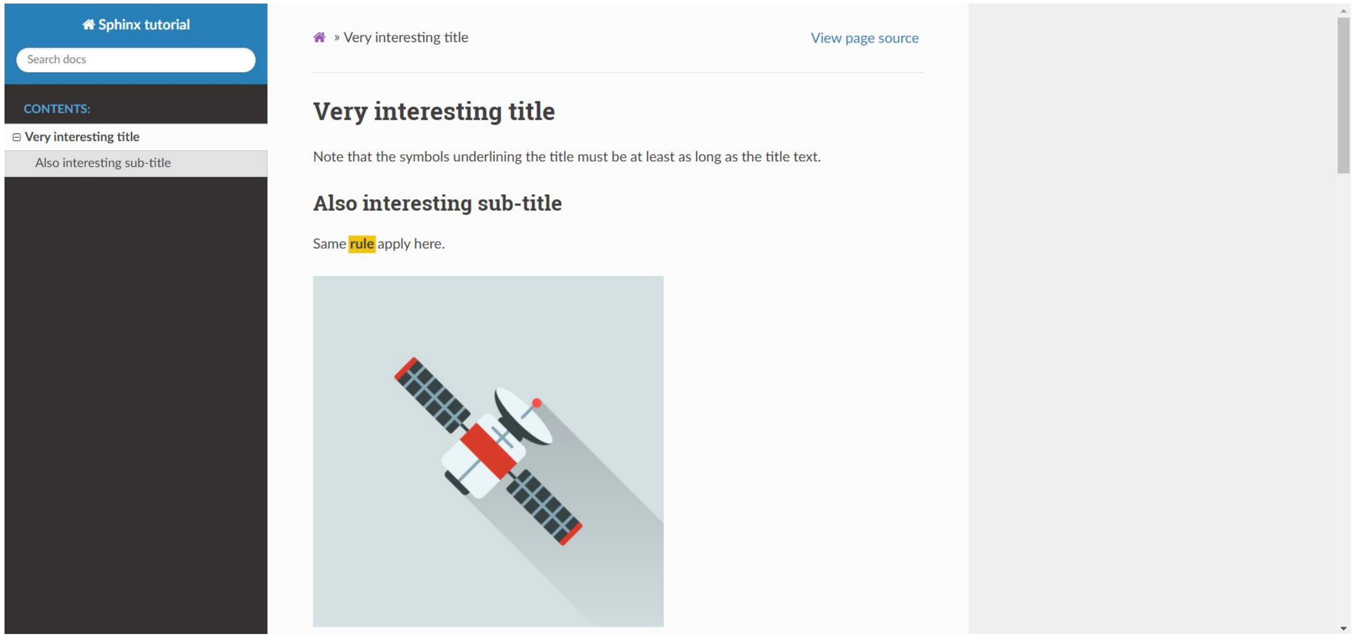


The iframe can also be modified directly in the rst file (for example we could make the canvas bigger or smaller)

Changing theme

We will now edit the `conf.py` file in order to change the theme of our documentation. Let's first download the theme provided by ReadTheDocs, simply running in the Anaconda Prompt `pip install sphinx_rtd_theme`. Once it's downloaded, let's open the `conf.py` file in a text editor and change the `html_theme` variable to `'sphinx_rtd_theme'`.

If we build the html version of our documentation, we will see the changes.



Very interesting title

Note that the symbols underlining the title must be at least as long as the title text.

Also interesting sub-title

Same rule apply here.

Hosting - 1

We are ready to host our documentation online, in order to made it accessible by anybody; we will use the service provided by ReadTheDocs.

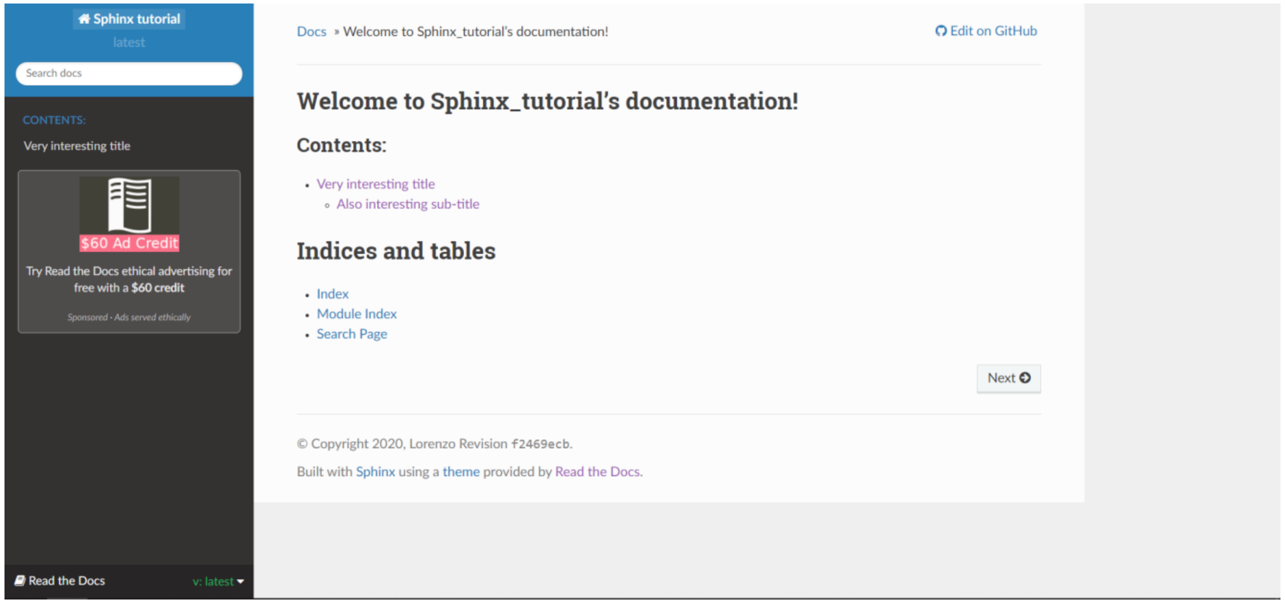
The first thing we need to do is add to the `conf.py` file the specification about the name of the master document (e.g. the one containing the main toctree, in our case index.rst), if not already present. Search your `conf.py` file for the `master_doc` variable: if you find it, check that the assigned value is `‘index’`. If you don’t find it, simply add the line `master_doc = ‘index’`.

Now we can push again our code on GitHub. Here are the usual steps:

```
git add .
git commit –m ‘‘added documentation content’’
git push origin master
```

Hosting - 2

Now visit https://readthedocs.org/ and sign in with your GitHub profile. Go to your profile page, click on 'Import a Project' and select 'Import manually'. In the new window we can choose the project's name, and then we need to fill the Repository URL field with `https://github.com/<your GitHub name>/<repository name>.git`. Select 'Next' and then 'Build version': this will start building the documentation out of the GitHub code we pushed. Once the process is finished, we can click on 'View docs' and go to the online and openly reachable page of our documentation.



Docs » Welcome to Sphinx_tutorial's documentation!

Welcome to Sphinx_tutorial's documentation!

Contents:

- Very interesting title
  - Also interesting sub-title

Indices and tables

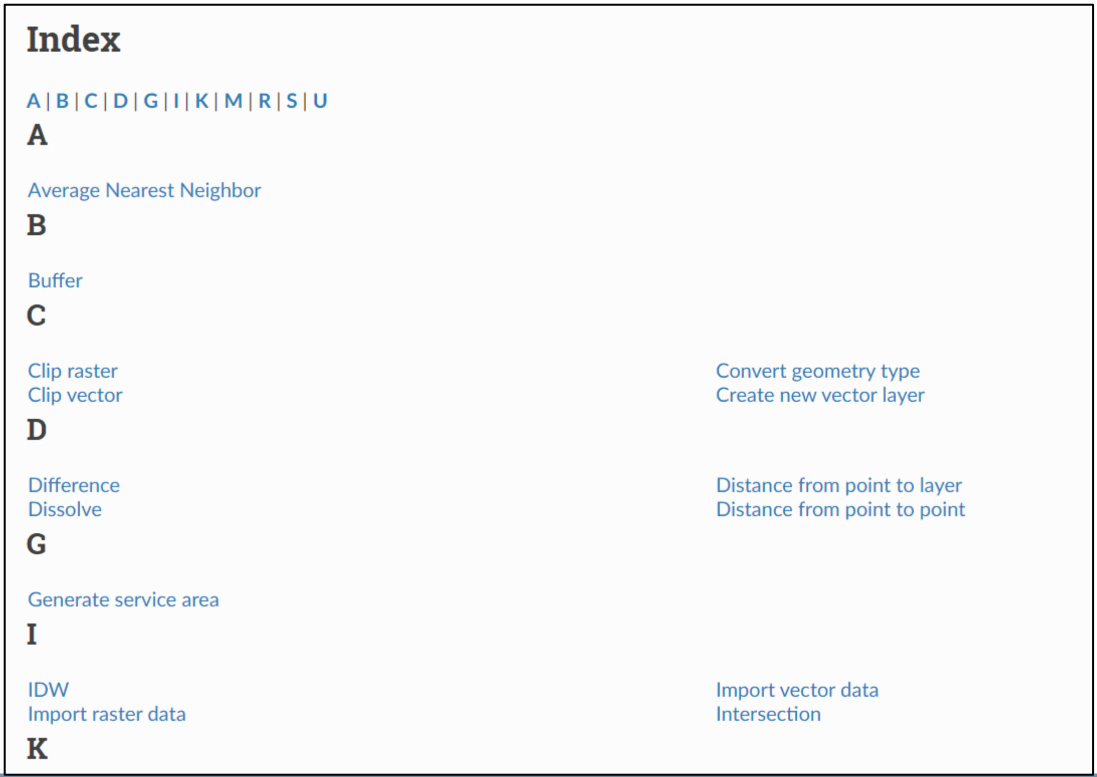
- Index
- Module Index
- Search Page

© Copyright 2020, Lorenzo Revision `f2469ecb`.

Built with Sphinx using a theme provided by Read the Docs.

Appendix A - index

It could be useful to create an alphabetical index of the sections/elements contained in our documentation. This can be done by including each word that we want to be in the index into an index rst directive, e.g. `:index:`QGIS <single: QGIS>``. This would create a single index entry.



Index

A | B | C | D | G | I | K | M | R | S | U

A

Average Nearest Neighbor

B

Buffer

C

Clip raster
Clip vector
Convert geometry type
Create new vector layer

D

Difference
Dissolve
Distance from point to layer
Distance from point to point

G

Generate service area

I

IDW
Import raster data
Import vector data
Intersection

K

Example of an index created with ReStructuredText

Appendix B – custom CSS

Another useful feature is the possiility to add custom CSS code to style our web documentation. To do so, we need to add a new CSS file in the _static folder of our documentation.

In order for it to work, we need to connect the CSS stylesheet to our documentation: this is done in the conf.py file adding the line `html_css_files = [‘name_of_the_file.css’]`.

If you want to try for yourself, you can try to modify the width of the main content of the documentation using the following code:

```
.wy-nav-content{
   max-width: 1000px !important;
}
```

Thank you for your attention

Questions?