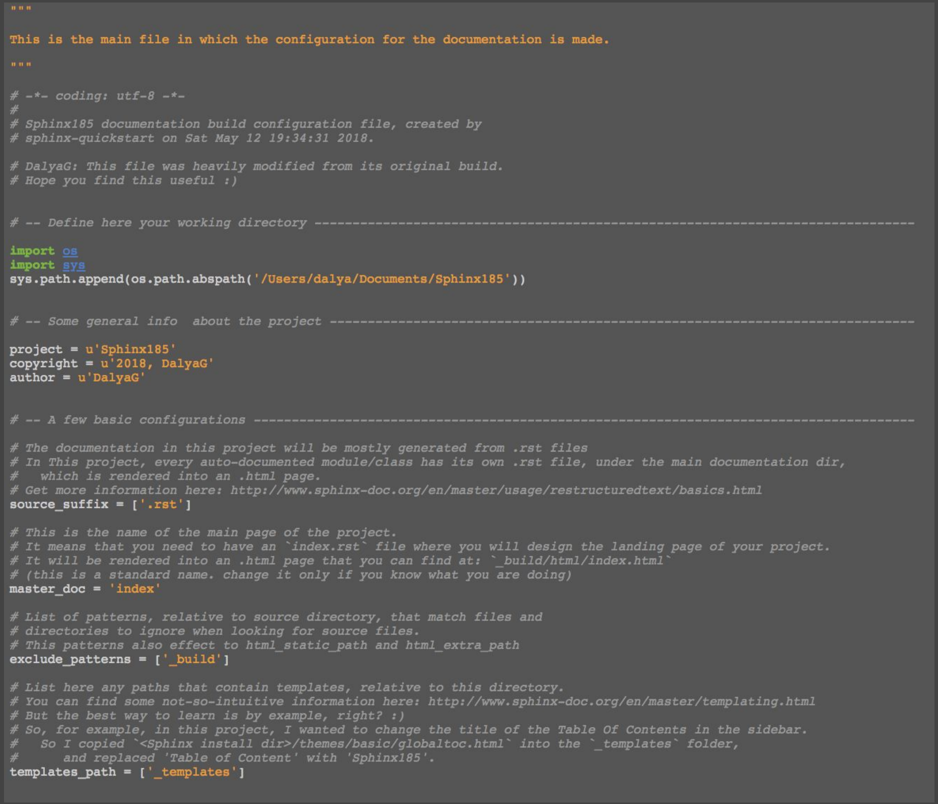
Help your colleagues help themselves - a Sphinx tutorial

Dalya Gartzman

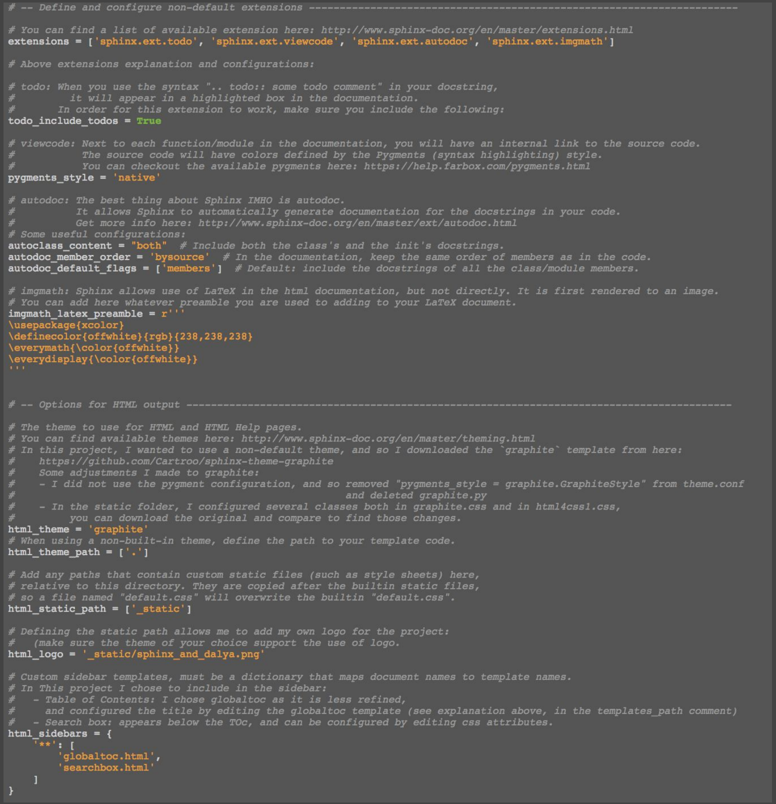
github.com/DalyaG/Sphinx185

Main configuration file - conf.py

Part II - Main configuration file - conf.py

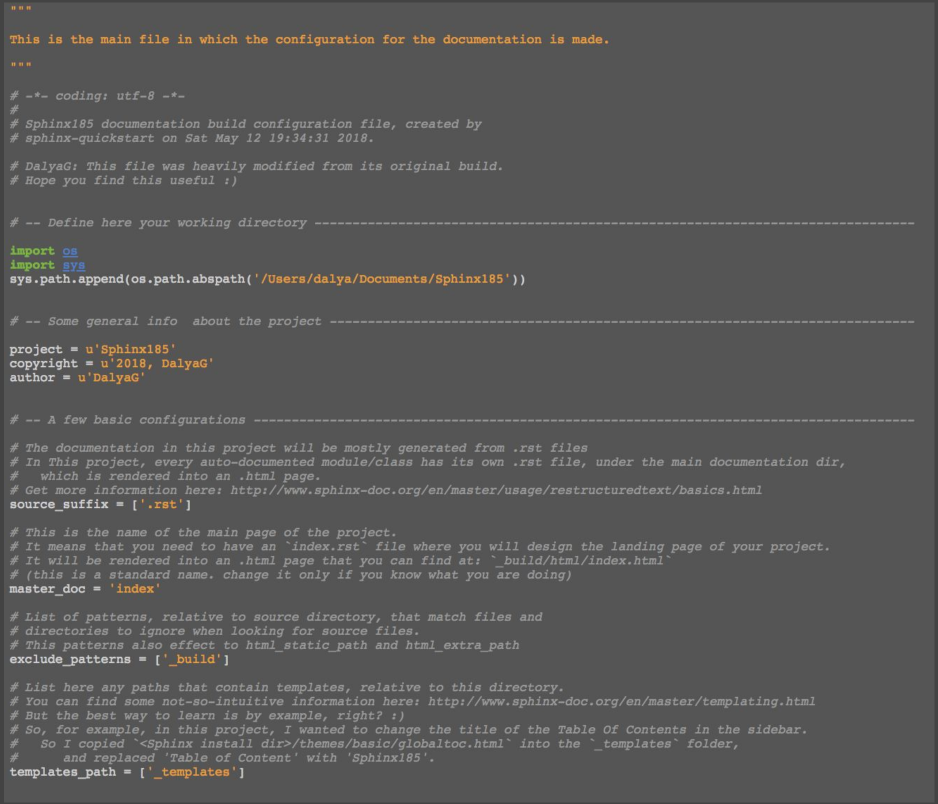


```python
"""

This is the main file in which the configuration for the documentation is made.

"""

# -*- coding: utf-8 -*-
#
# Sphinx185 documentation build configuration file, created by
# sphinx-quickstart on Sat May 12 19:34:31 2018.

# DalyaG: This file was heavily modified from its original build.
# Hope you find this useful :)


# -- Define here your working directory ---------------------------------------------------------------

import os
import sys
sys.path.append(os.path.abspath('/Users/dalya/Documents/Sphinx185'))


# -- Some general info  about the project -------------------------------------------------------------

project = u'Sphinx185'
copyright = u'2018, DalyaG'
author = u'DalyaG'


# -- A few basic configurations -----------------------------------------------------------------------

# The documentation in this project will be mostly generated from .rst files
# In This project, every auto-documented module/class has its own .rst file, under the main documentation dir,
#   which is rendered into an .html page.
# Get more information here: http://www.sphinx-doc.org/en/master/usage/restructuredtext/basics.html
source_suffix = ['.rst']

# This is the name of the main page of the project.
# It means that you need to have an `index.rst` file where you will design the landing page of your project.
# It will be rendered into an .html page that you can find at: `_build/html/index.html`
# (this is a standard name. change it only if you know what you are doing)
master_doc = 'index'

# List of patterns, relative to source directory, that match files and
# directories to ignore when looking for source files.
# This patterns also effect to html_static_path and html_extra_path
exclude_patterns = ['_build']

# List here any paths that contain templates, relative to this directory.
# You can find some not-so-intuitive information here: http://www.sphinx-doc.org/en/master/templating.html
# But the best way to learn is by example, right? :)
# So, for example, in this project, I wanted to change the title of the Table Of Contents in the sidebar.
#   So I copied `<Sphinx install dir>/themes/basic/globaltoc.html` into the `_templates` folder,
#      and replaced 'Table of Content' with 'Sphinx185'.
templates_path = ['_templates']
```



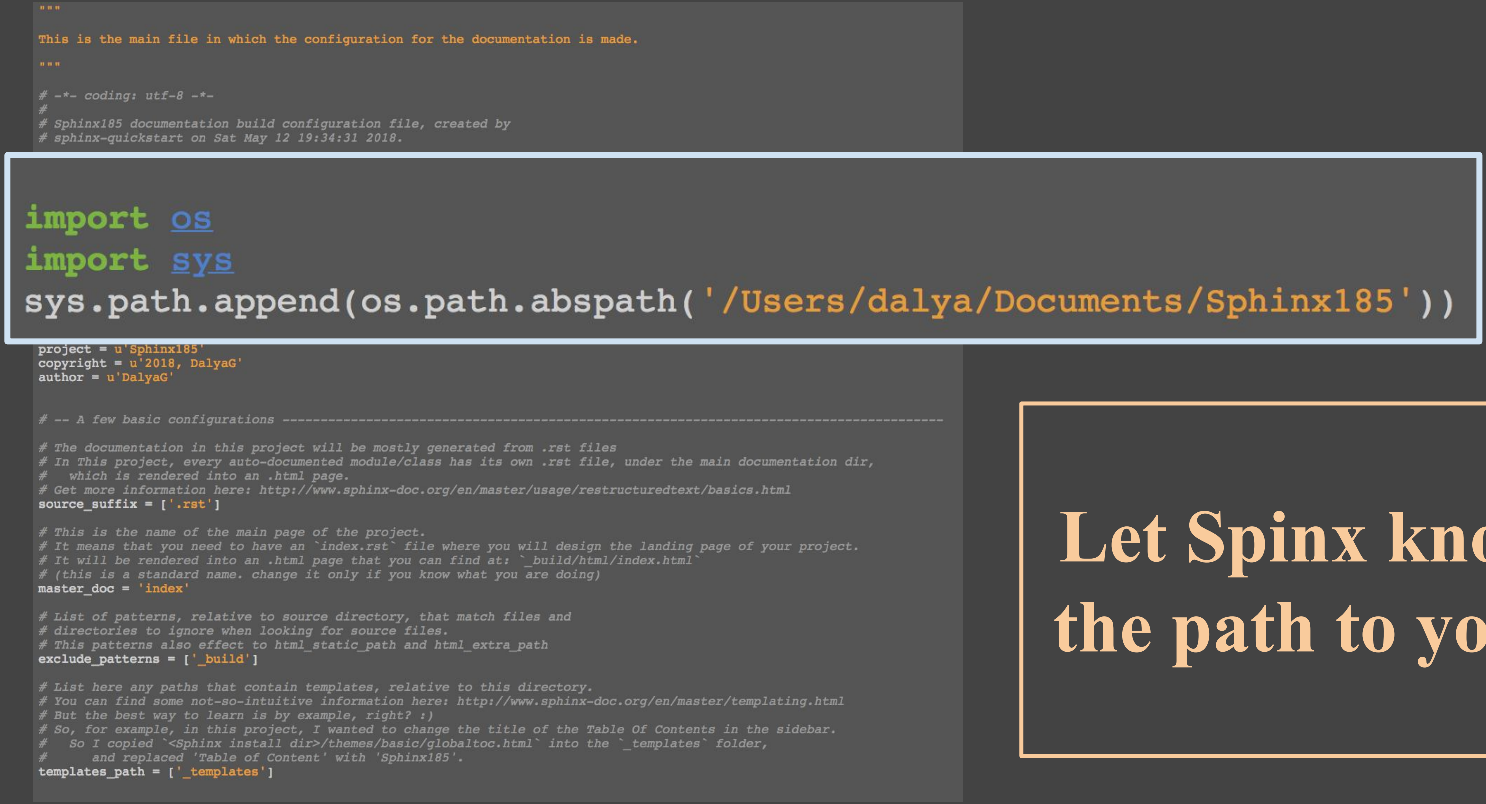
```python
# -- Define and configure non-default extensions ---------------------------------------------------------

# You can find a list of available extension here: http://www.sphinx-doc.org/en/master/extensions.html
extensions = ['sphinx.ext.todo', 'sphinx.ext.viewcode', 'sphinx.ext.autodoc', 'sphinx.ext.imgmath']

# Above extensions explanation and configurations:

# todo: When you use the syntax ".. todo:: some todo comment" in your docstring,
#          it will appear in a highlighted box in the documentation.
#        In order for this extension to work, make sure you include the following:
todo_include_todos = True

# viewcode: Next to each function/module in the documentation, you will have an internal link to the source code.
#            The source code will have colors defined by the Pygments (syntax highlighting) style.
#            You can checkout the available pygments here: https://help.farbox.com/pygments.html
pygments_style = 'native'

# autodoc: The best thing about Sphinx IMHO is autodoc.
#           It allows Sphinx to automatically generate documentation for the docstrings in your code.
#           Get more info here: http://www.sphinx-doc.org/en/master/ext/autodoc.html
# Some useful configurations:
autoclass_content = "both"  # Include both the class's and the init's docstrings.
autodoc_member_order = 'bysource'  # In the documentation, keep the same order of members as in the code.
autodoc_default_flags = ['members']  # Default: include the docstrings of all the class/module members.

# imgmath: Sphinx allows use of LaTeX in the html documentation, but not directly. It is first rendered to an image.
# You can add here whatever preamble you are used to adding to your LaTeX document.
imgmath_latex_preamble = r'''
\usepackage{xcolor}
\definecolor{offwhite}{rgb}{238,238,238}
\everymath{\color{offwhite}}
\everydisplay{\color{offwhite}}
'''


# -- Options for HTML output ------------------------------------------------------------------------------

# The theme to use for HTML and HTML Help pages.
# You can find available themes here: http://www.sphinx-doc.org/en/master/theming.html
# In this project, I wanted to use a non-default theme, and so I downloaded the `graphite` template from here:
#    https://github.com/Cartroo/sphinx-theme-graphite
#    Some adjustments I made to graphite:
#    - I did not use the pygment configuration, and so removed "pygments_style = graphite.GraphiteStyle" from theme.conf
#                                                         and deleted graphite.py
#    - In the static folder, I configured several classes both in graphite.css and in html4css1.css,
#         you can download the original and compare to find those changes.
html_theme = 'graphite'
# When using a non-built-in theme, define the path to your template code.
html_theme_path = ['.']

# Add any paths that contain custom static files (such as style sheets) here,
# relative to this directory. They are copied after the builtin static files,
# so a file named "default.css" will overwrite the builtin "default.css".
html_static_path = ['_static']

# Defining the static path allows me to add my own logo for the project:
#   (make sure the theme of your choice support the use of logo.
html_logo = '_static/sphinx_and_dalya.png'

# Custom sidebar templates, must be a dictionary that maps document names to template names.
# In This project I chose to include in the sidebar:
#   - Table of Contents: I chose globaltoc as it is less refined,
#     and configured the title by editing the globaltoc template (see explanation above, in the templates_path comment)
#   - Search box: appears below the TOc, and can be configured by editing css attributes.
html_sidebars = {
    '**': [
        'globaltoc.html',
        'searchbox.html'
    ]
}
```

Part II - Main configuration file - conf.py



```
"""

This is the main file in which the configuration for the documentation is made.

"""

# -*- coding: utf-8 -*-
#
# Sphinx185 documentation build configuration file, created by
# sphinx-quickstart on Sat May 12 19:34:31 2018.

# DalyaG: This file was heavily modified from its original build.
# Hope you find this useful :)


# -- Define here your working directory -------------------------------------------------------------------------------

import os
import sys
sys.path.append(os.path.abspath('/Users/dalya/Documents/Sphinx185'))


# -- Some general info  about the project ------------------------------------------------------------------------------

project = u'Sphinx185'
copyright = u'2018, DalyaG'
author = u'DalyaG'


# -- A few basic configurations ---------------------------------------------------------------------------------------

# The documentation in this project will be mostly generated from .rst files
# In This project, every auto-documented module/class has its own .rst file, under the main documentation dir,
#   which is rendered into an .html page.
# Get more information here: http://www.sphinx-doc.org/en/master/usage/restructuredtext/basics.html
source_suffix = ['.rst']

# This is the name of the main page of the project.
# It means that you need to have an `index.rst` file where you will design the landing page of your project.
# It will be rendered into an .html page that you can find at: `_build/html/index.html`
# (this is a standard name. change it only if you know what you are doing)
master_doc = 'index'

# List of patterns, relative to source directory, that match files and
# directories to ignore when looking for source files.
# This patterns also effect to html_static_path and html_extra_path
exclude_patterns = ['_build']

# List here any paths that contain templates, relative to this directory.
# You can find some not-so-intuitive information here: http://www.sphinx-doc.org/en/master/templating.html
# But the best way to learn is by example, right? :)
# So, for example, in this project, I wanted to change the title of the Table Of Contents in the sidebar.
#   So I copied `<Sphinx install dir>/themes/basic/globaltoc.html` into the `_templates` folder,
#      and replaced 'Table of Content' with 'Sphinx185'.
templates_path = ['_templates']
```

Part II - Main configuration file - conf.py



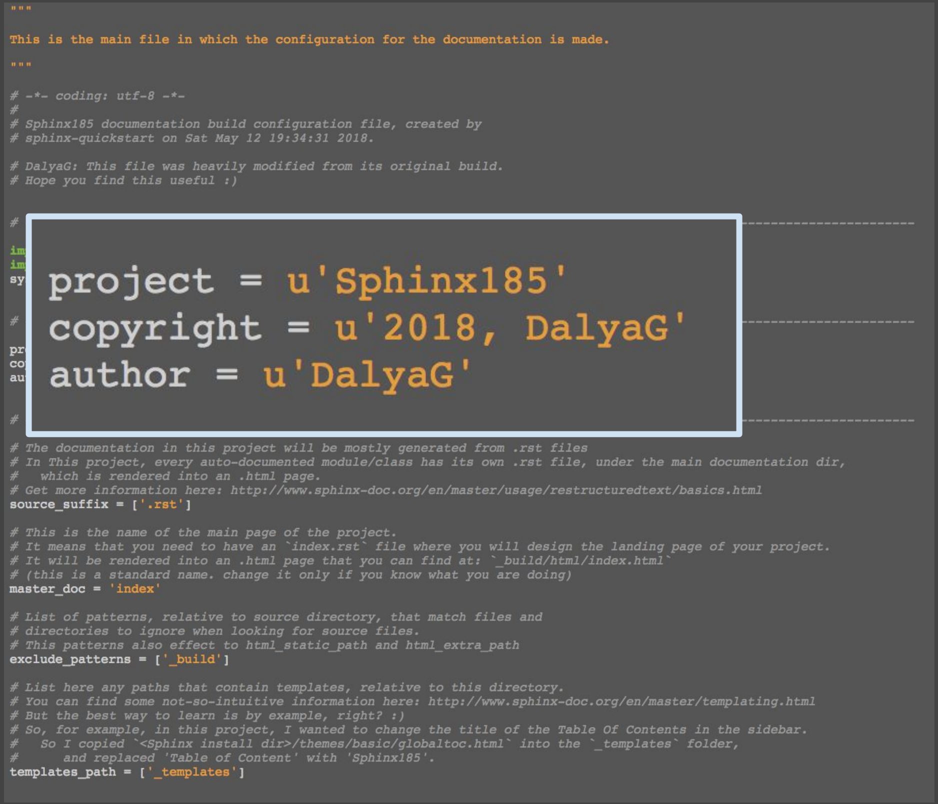
```
"""

This is the main file in which the configuration for the documentation is made.

"""

# -*- coding: utf-8 -*-
#
# Sphinx185 documentation build configuration file, created by
# sphinx-quickstart on Sat May 12 19:34:31 2018.
```

```
import os
import sys
sys.path.append(os.path.abspath('/Users/dalya/Documents/Sphinx185'))
```

```
project = u'Sphinx185'
copyright = u'2018, DalyaG'
author = u'DalyaG'


# -- A few basic configurations --------------------------------------------------------------------------------------

# The documentation in this project will be mostly generated from .rst files
# In This project, every auto-documented module/class has its own .rst file, under the main documentation dir,
#   which is rendered into an .html page.
# Get more information here: http://www.sphinx-doc.org/en/master/usage/restructuredtext/basics.html
source_suffix = ['.rst']

# This is the name of the main page of the project.
# It means that you need to have an `index.rst` file where you will design the landing page of your project.
# It will be rendered into an .html page that you can find at: `_build/html/index.html`
# (this is a standard name. change it only if you know what you are doing)
master_doc = 'index'

# List of patterns, relative to source directory, that match files and
# directories to ignore when looking for source files.
# This patterns also effect to html_static_path and html_extra_path
exclude_patterns = ['_build']

# List here any paths that contain templates, relative to this directory.
# You can find some not-so-intuitive information here: http://www.sphinx-doc.org/en/master/templating.html
# But the best way to learn is by example, right? :)
# So, for example, in this project, I wanted to change the title of the Table Of Contents in the sidebar.
#   So I copied `<Sphinx install dir>/themes/basic/globaltoc.html` into the `_templates` folder,
#      and replaced 'Table of Content' with 'Sphinx185'.
templates_path = ['_templates']
```

Let Spinx know what is the path to your project

Part II - Main configuration file - conf.py

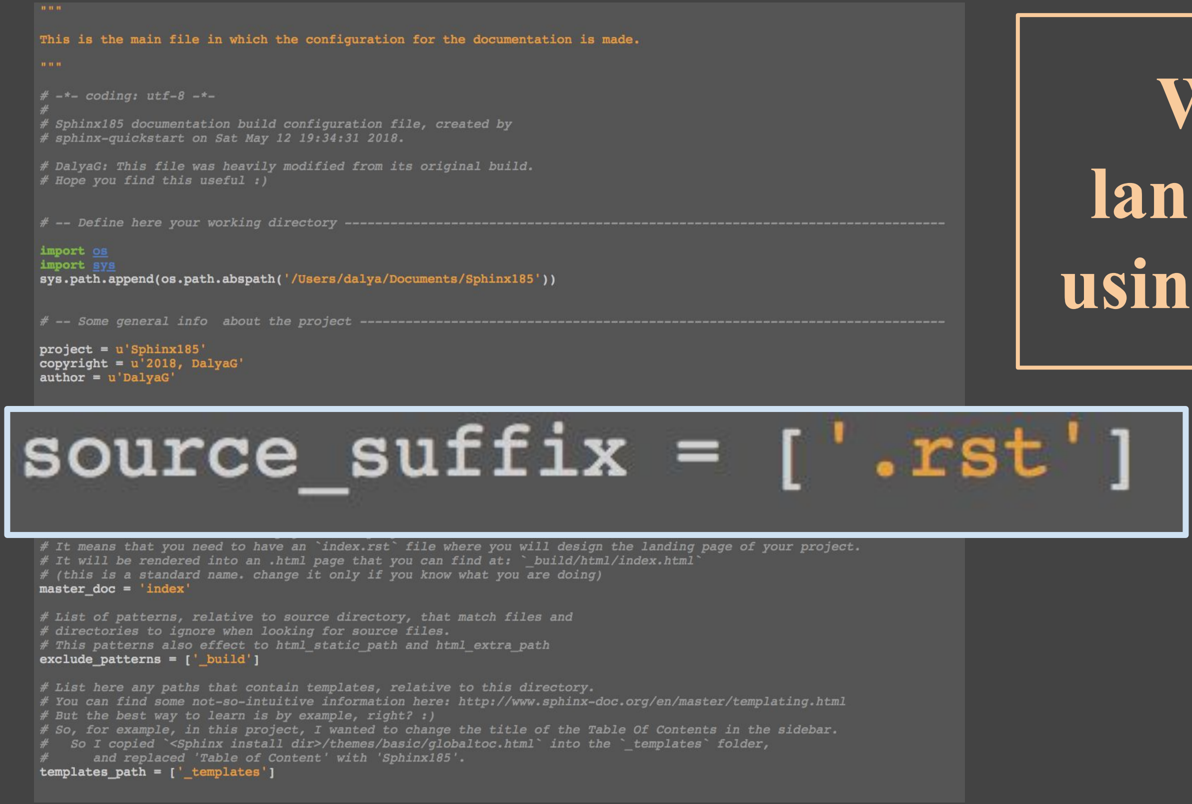


```
"""

This is the main file in which the configuration for the documentation is made.

"""

# -*- coding: utf-8 -*-
#
# Sphinx185 documentation build configuration file, created by
# sphinx-quickstart on Sat May 12 19:34:31 2018.

# DalyaG: This file was heavily modified from its original build.
# Hope you find this useful :)


#                                                                  ------------------------

im
im
sy

#                                                                  ------------------------

pr
co
au

#                                                                  ------------------------
# The documentation in this project will be mostly generated from .rst files
# In This project, every auto-documented module/class has its own .rst file, under the main documentation dir,
#   which is rendered into an .html page.
# Get more information here: http://www.sphinx-doc.org/en/master/usage/restructuredtext/basics.html
source_suffix = ['.rst']

# This is the name of the main page of the project.
# It means that you need to have an `index.rst` file where you will design the landing page of your project.
# It will be rendered into an .html page that you can find at: `_build/html/index.html`
# (this is a standard name. change it only if you know what you are doing)
master_doc = 'index'

# List of patterns, relative to source directory, that match files and
# directories to ignore when looking for source files.
# This patterns also effect to html_static_path and html_extra_path
exclude_patterns = ['_build']

# List here any paths that contain templates, relative to this directory.
# You can find some not-so-intuitive information here: http://www.sphinx-doc.org/en/master/templating.html
# But the best way to learn is by example, right? :)
# So, for example, in this project, I wanted to change the title of the Table Of Contents in the sidebar.
#   So I copied `<Sphinx install dir>/themes/basic/globaltoc.html` into the `_templates` folder,
#      and replaced 'Table of Content' with 'Sphinx185'.
templates_path = ['_templates']
```

```
project = u'Sphinx185'
copyright = u'2018, DalyaG'
author = u'DalyaG'
```

Some basic info about the project. This will appear in the documentation.

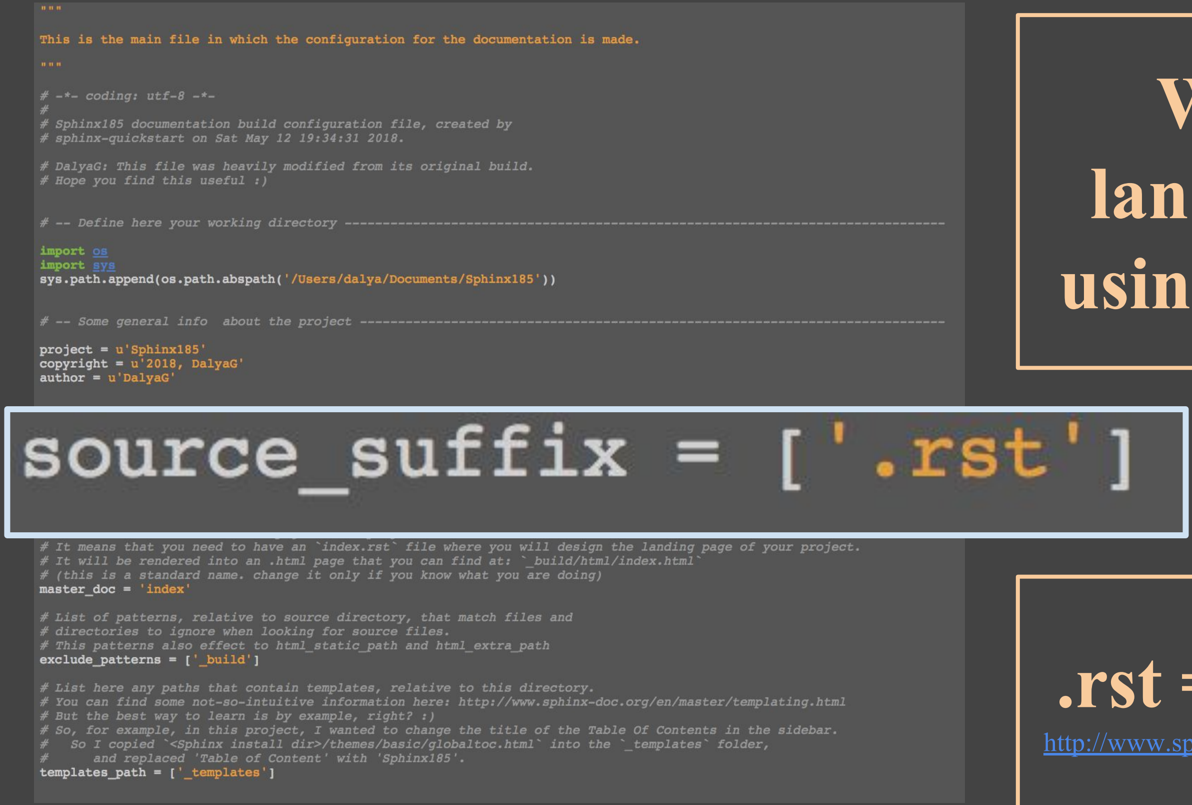
Part II - Main configuration file - conf.py



```
"""

This is the main file in which the configuration for the documentation is made.

"""

# -*- coding: utf-8 -*-
#
# Sphinx185 documentation build configuration file, created by
# sphinx-quickstart on Sat May 12 19:34:31 2018.

# DalyaG: This file was heavily modified from its original build.
# Hope you find this useful :)


# -- Define here your working directory -----------------------------------------------------------------------------

import os
import sys
sys.path.append(os.path.abspath('/Users/dalya/Documents/Sphinx185'))


# -- Some general info  about the project ---------------------------------------------------------------------------

project = u'Sphinx185'
copyright = u'2018, DalyaG'
author = u'DalyaG'

source_suffix = ['.rst']

# It means that you need to have an `index.rst` file where you will design the landing page of your project.
# It will be rendered into an .html page that you can find at: `_build/html/index.html`
# (this is a standard name. change it only if you know what you are doing)
master_doc = 'index'

# List of patterns, relative to source directory, that match files and
# directories to ignore when looking for source files.
# This patterns also effect to html_static_path and html_extra_path
exclude_patterns = ['_build']

# List here any paths that contain templates, relative to this directory.
# You can find some not-so-intuitive information here: http://www.sphinx-doc.org/en/master/templating.html
# But the best way to learn is by example, right? :)
# So, for example, in this project, I wanted to change the title of the Table Of Contents in the sidebar.
#   So I copied `<Sphinx install dir>/themes/basic/globaltoc.html` into the `_templates` folder,
#      and replaced 'Table of Content' with 'Sphinx185'.
templates_path = ['_templates']
```

What markdown languages will you be using to generate pages

Part II - Main configuration file - conf.py



```
"""

This is the main file in which the configuration for the documentation is made.

"""

# -*- coding: utf-8 -*-
#
# Sphinx185 documentation build configuration file, created by
# sphinx-quickstart on Sat May 12 19:34:31 2018.

# DalyaG: This file was heavily modified from its original build.
# Hope you find this useful :)


# -- Define here your working directory ---------------------------------------------------------------------------------

import os
import sys
sys.path.append(os.path.abspath('/Users/dalya/Documents/Sphinx185'))


# -- Some general info  about the project -----------------------------------------------------------------------------

project = u'Sphinx185'
copyright = u'2018, DalyaG'
author = u'DalyaG'
```

```
source_suffix = ['.rst']
```

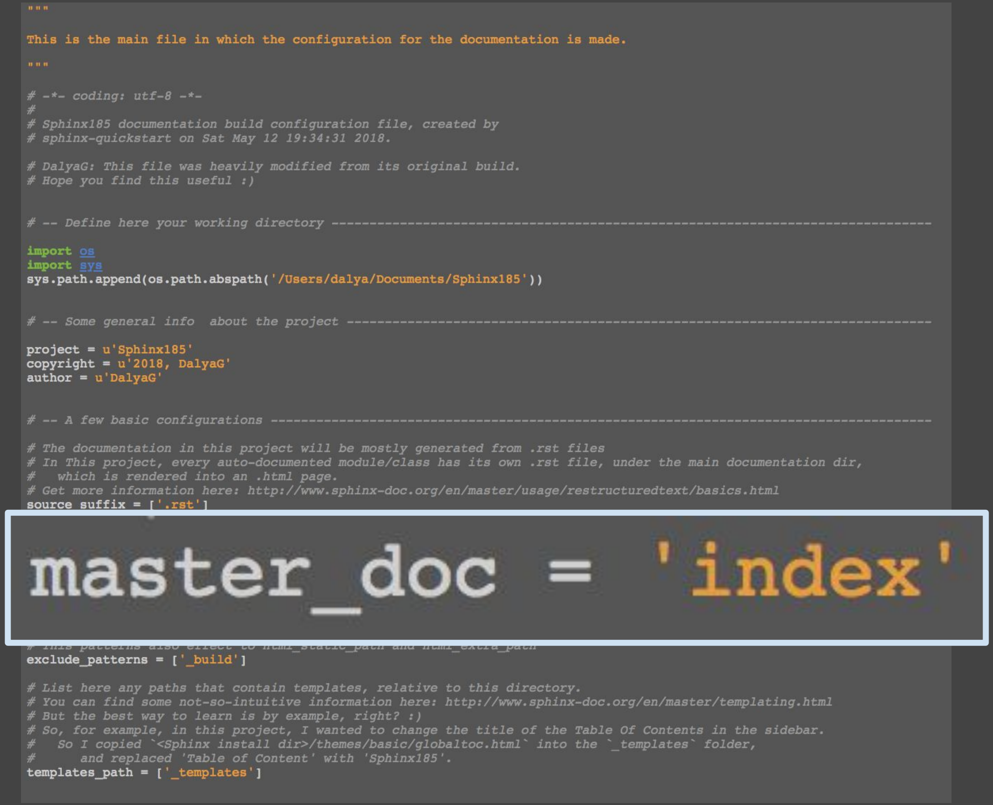
```
# It means that you need to have an `index.rst` file where you will design the landing page of your project.
# It will be rendered into an .html page that you can find at: `_build/html/index.html`
# (this is a standard name. change it only if you know what you are doing)
master_doc = 'index'

# List of patterns, relative to source directory, that match files and
# directories to ignore when looking for source files.
# This patterns also effect to html_static_path and html_extra_path
exclude_patterns = ['_build']

# List here any paths that contain templates, relative to this directory.
# You can find some not-so-intuitive information here: http://www.sphinx-doc.org/en/master/templating.html
# But the best way to learn is by example, right? :)
# So, for example, in this project, I wanted to change the title of the Table Of Contents in the sidebar.
#   So I copied `<Sphinx install dir>/themes/basic/globaltoc.html` into the `_templates` folder,
#      and replaced 'Table of Content' with 'Sphinx185'.
templates_path = ['_templates']
```

What markdown languages will you be using to generate pages

.rst = reStructuredText

http://www.sphinx-doc.org/en/master/usage/restructuredtext/basics.html

Part II - Main configuration file - conf.py



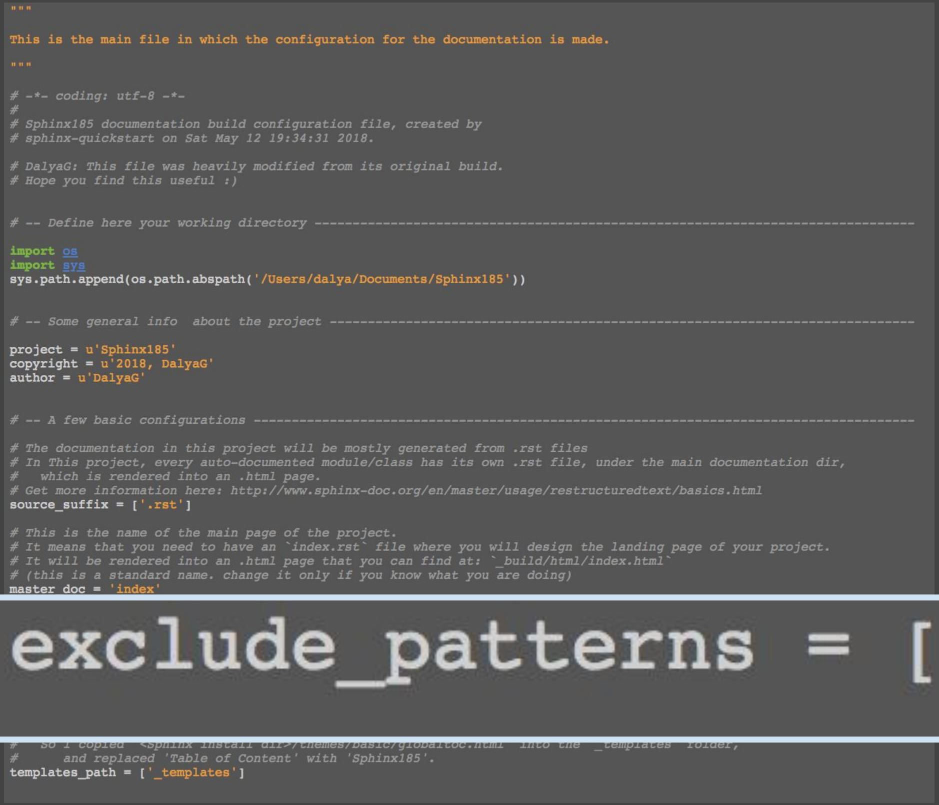
```
"""
This is the main file in which the configuration for the documentation is made.
"""

# -*- coding: utf-8 -*-
#
# Sphinx185 documentation build configuration file, created by
# sphinx-quickstart on Sat May 12 19:34:31 2018.

# DalyaG: This file was heavily modified from its original build.
# Hope you find this useful :)


# -- Define here your working directory -------------------------------------------------------------------------

import os
import sys
sys.path.append(os.path.abspath('/Users/dalya/Documents/Sphinx185'))


# -- Some general info  about the project -----------------------------------------------------------------------

project = u'Sphinx185'
copyright = u'2018, DalyaG'
author = u'DalyaG'


# -- A few basic configurations ----------------------------------------------------------------------------------

# The documentation in this project will be mostly generated from .rst files
# In This project, every auto-documented module/class has its own .rst file, under the main documentation dir,
#   which is rendered into an .html page.
# Get more information here: http://www.sphinx-doc.org/en/master/usage/restructuredtext/basics.html
source_suffix = ['.rst']
```

```
master_doc = 'index'
```

```
# This patterns also effect to html_static_path and html_extra_path
exclude_patterns = ['_build']

# List here any paths that contain templates, relative to this directory.
# You can find some not-so-intuitive information here: http://www.sphinx-doc.org/en/master/templating.html
# But the best way to learn is by example, right? :)
# So, for example, in this project, I wanted to change the title of the Table Of Contents in the sidebar.
#   So I copied `<Sphinx install dir>/themes/basic/globaltoc.html` into the `_templates` folder,
#      and replaced 'Table of Content' with 'Sphinx185'.
templates_path = ['_templates']
```

What is the name of the master document (landing page)

Part II - Main configuration file - conf.py



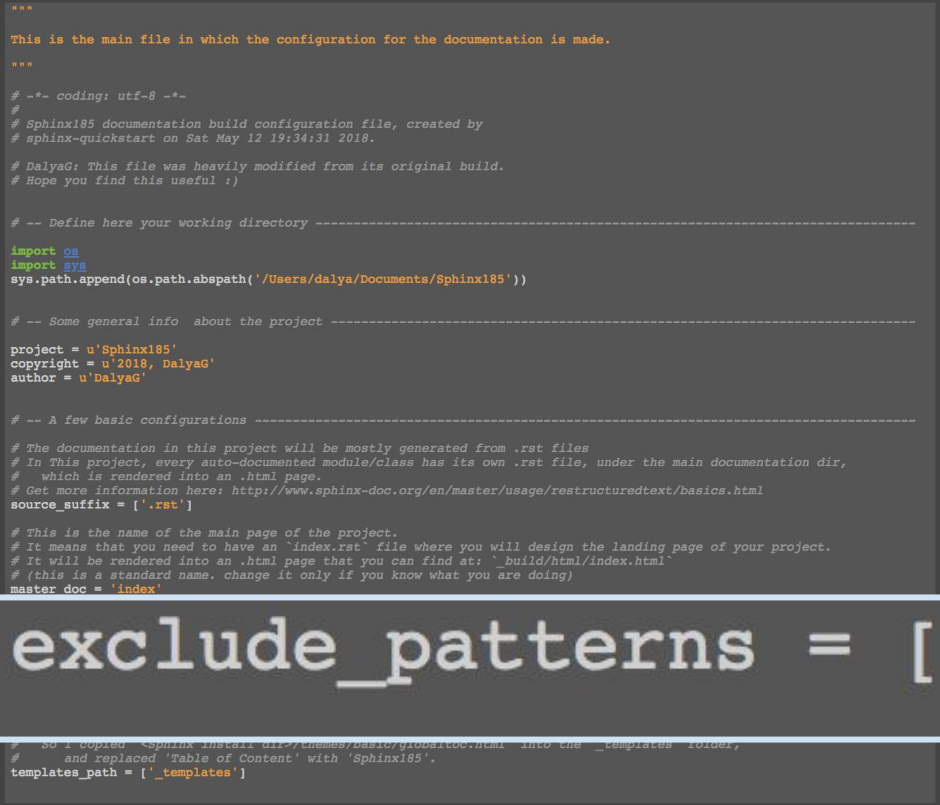
```
"""

This is the main file in which the configuration for the documentation is made.

"""

# -*- coding: utf-8 -*-
#
# Sphinx185 documentation build configuration file, created by
# sphinx-quickstart on Sat May 12 19:34:31 2018.

# DalyaG: This file was heavily modified from its original build.
# Hope you find this useful :)


# -- Define here your working directory ---------------------------------------------------------------------------------

import os
import sys
sys.path.append(os.path.abspath('/Users/dalya/Documents/Sphinx185'))


# -- Some general info  about the project -------------------------------------------------------------------------------

project = u'Sphinx185'
copyright = u'2018, DalyaG'
author = u'DalyaG'


# -- A few basic configurations -----------------------------------------------------------------------------------------

# The documentation in this project will be mostly generated from .rst files
# In This project, every auto-documented module/class has its own .rst file, under the main documentation dir,
#   which is rendered into an .html page.
# Get more information here: http://www.sphinx-doc.org/en/master/usage/restructuredtext/basics.html
source_suffix = ['.rst']

# This is the name of the main page of the project.
# It means that you need to have an `index.rst` file where you will design the landing page of your project.
# It will be rendered into an .html page that you can find at: `_build/html/index.html`
# (this is a standard name. change it only if you know what you are doing)
master_doc = 'index'
```

```
exclude_patterns = ['_build']
```

```
#   So I copied `<Sphinx install dir>/themes/basic/globaltoc.html` into the `_templates` folder,
#      and replaced 'Table of Content' with 'Sphinx185'.
templates_path = ['_templates']
```

Inside '_build' is the output of the documentation.

Part II - Main configuration file - conf.py



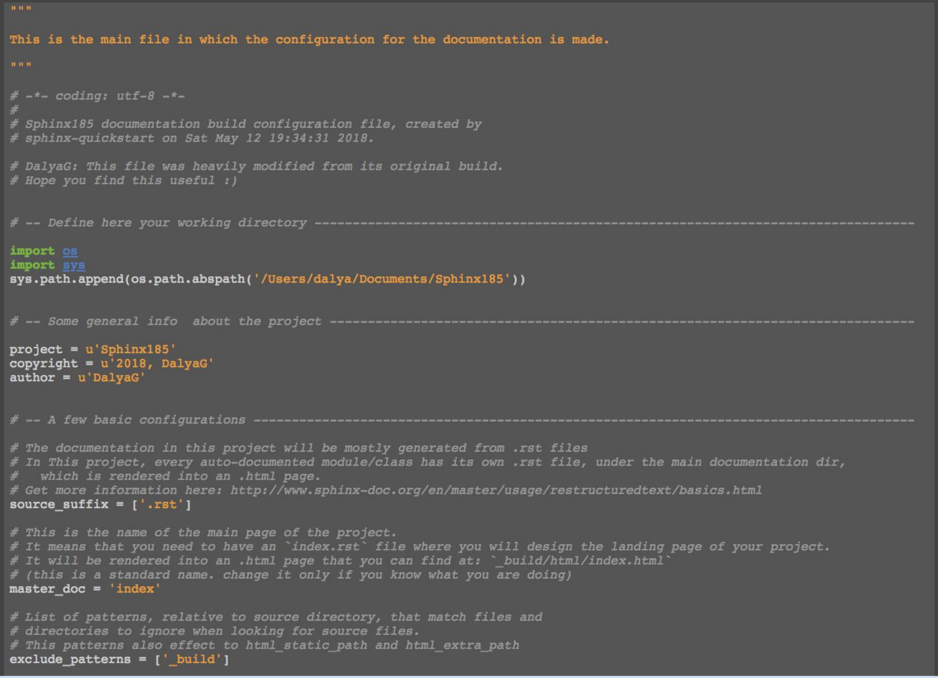
```python
"""

This is the main file in which the configuration for the documentation is made.

"""

# -*- coding: utf-8 -*-
#
# Sphinx185 documentation build configuration file, created by
# sphinx-quickstart on Sat May 12 19:34:31 2018.

# DalyaG: This file was heavily modified from its original build.
# Hope you find this useful :)


# -- Define here your working directory ---------------------------------------------------------------------------

import os
import sys
sys.path.append(os.path.abspath('/Users/dalya/Documents/Sphinx185'))


# -- Some general info  about the project -------------------------------------------------------------------------

project = u'Sphinx185'
copyright = u'2018, DalyaG'
author = u'DalyaG'


# -- A few basic configurations -----------------------------------------------------------------------------------

# The documentation in this project will be mostly generated from .rst files
# In This project, every auto-documented module/class has its own .rst file, under the main documentation dir,
#   which is rendered into an .html page.
# Get more information here: http://www.sphinx-doc.org/en/master/usage/restructuredtext/basics.html
source_suffix = ['.rst']

# This is the name of the main page of the project.
# It means that you need to have an `index.rst` file where you will design the landing page of your project.
# It will be rendered into an .html page that you can find at: `_build/html/index.html`
# (this is a standard name. change it only if you know what you are doing)
master_doc = 'index'

#   So I copied `<Sphinx install dir>/themes/basic/globaltoc.html` into the `_templates` folder,
#      and replaced 'Table of Content' with 'Sphinx185'.
templates_path = ['_templates']
```

```python
exclude_patterns = ['_build']
```

Inside '_build' is the output of the documentation.

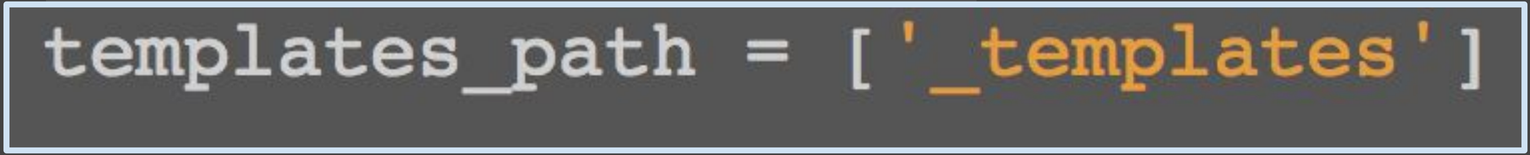
And we don't want to recursively document what we recursively document what we...

Part II - Main configuration file - conf.py



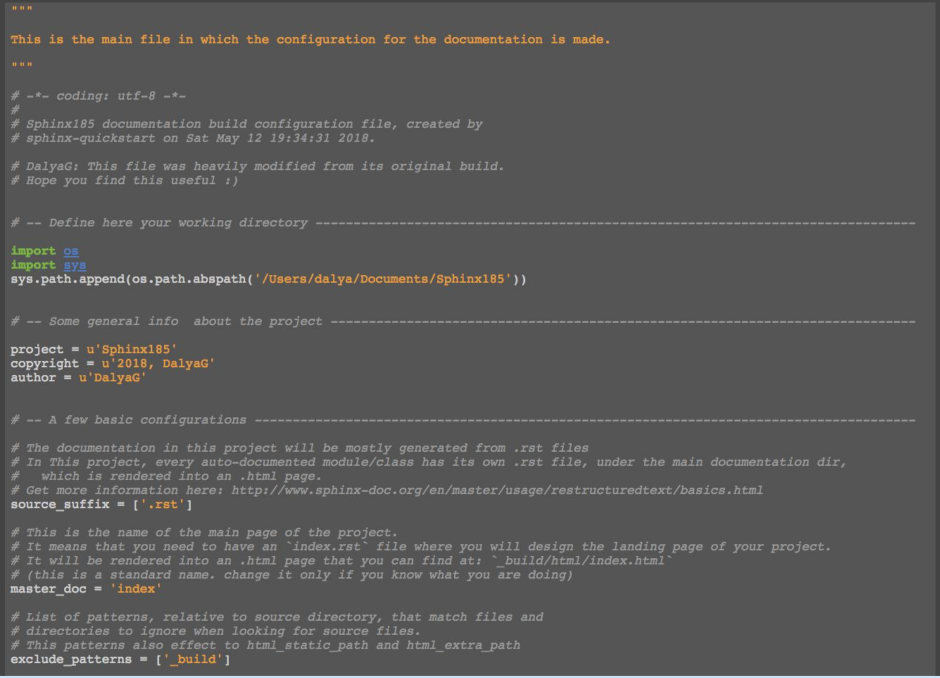
```
"""

This is the main file in which the configuration for the documentation is made.

"""

# -*- coding: utf-8 -*-
#
# Sphinx185 documentation build configuration file, created by
# sphinx-quickstart on Sat May 12 19:34:31 2018.

# DalyaG: This file was heavily modified from its original build.
# Hope you find this useful :)


# -- Define here your working directory ---------------------------------------------------------------------------

import os
import sys
sys.path.append(os.path.abspath('/Users/dalya/Documents/Sphinx185'))


# -- Some general info  about the project -------------------------------------------------------------------------

project = u'Sphinx185'
copyright = u'2018, DalyaG'
author = u'DalyaG'


# -- A few basic configurations -----------------------------------------------------------------------------------

# The documentation in this project will be mostly generated from .rst files
# In This project, every auto-documented module/class has its own .rst file, under the main documentation dir,
#   which is rendered into an .html page.
# Get more information here: http://www.sphinx-doc.org/en/master/usage/restructuredtext/basics.html
source_suffix = ['.rst']

# This is the name of the main page of the project.
# It means that you need to have an `index.rst` file where you will design the landing page of your project.
# It will be rendered into an .html page that you can find at: `_build/html/index.html`
# (this is a standard name. change it only if you know what you are doing)
master_doc = 'index'

# List of patterns, relative to source directory, that match files and
# directories to ignore when looking for source files.
# This patterns also effect to html_static_path and html_extra_path
exclude_patterns = ['_build']
```

Templates are html exmaples for designing predifined elements

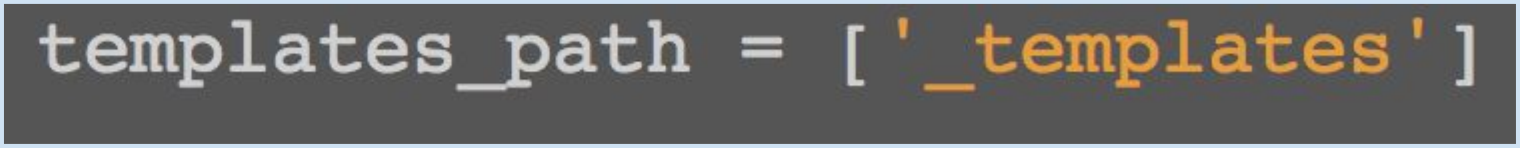


```
templates_path = ['_templates']
```

Part II - Main configuration file - conf.py



```
"""

This is the main file in which the configuration for the documentation is made.

"""

# -*- coding: utf-8 -*-
#
# Sphinx185 documentation build configuration file, created by
# sphinx-quickstart on Sat May 12 19:34:31 2018.

# DalyaG: This file was heavily modified from its original build.
# Hope you find this useful :)


# -- Define here your working directory ---------------------------------------------------------------------

import os
import sys
sys.path.append(os.path.abspath('/Users/dalya/Documents/Sphinx185'))


# -- Some general info  about the project -------------------------------------------------------------------

project = u'Sphinx185'
copyright = u'2018, DalyaG'
author = u'DalyaG'


# -- A few basic configurations -----------------------------------------------------------------------------

# The documentation in this project will be mostly generated from .rst files
# In This project, every auto-documented module/class has its own .rst file, under the main documentation dir,
#   which is rendered into an .html page.
# Get more information here: http://www.sphinx-doc.org/en/master/usage/restructuredtext/basics.html
source_suffix = ['.rst']

# This is the name of the main page of the project.
# It means that you need to have an `index.rst` file where you will design the landing page of your project.
# It will be rendered into an .html page that you can find at: `_build/html/index.html`
# (this is a standard name. change it only if you know what you are doing)
master_doc = 'index'

# List of patterns, relative to source directory, that match files and
# directories to ignore when looking for source files.
# This patterns also effect to html_static_path and html_extra_path
exclude_patterns = ['_build']
```

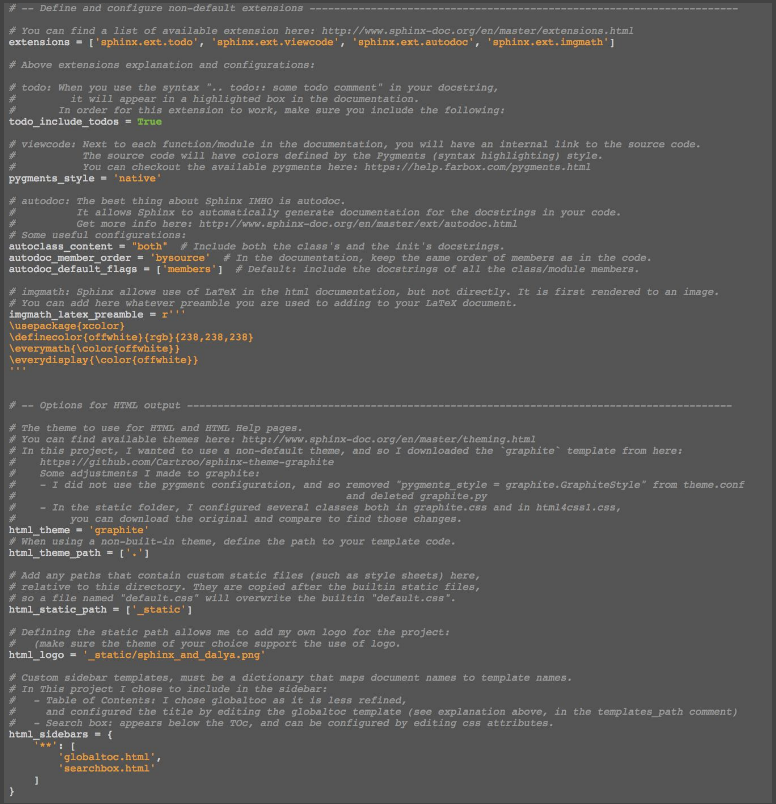


```
templates_path = ['_templates']
```

Templates are html exmaples for designing predifined elements

We will see:
a template for sidebar

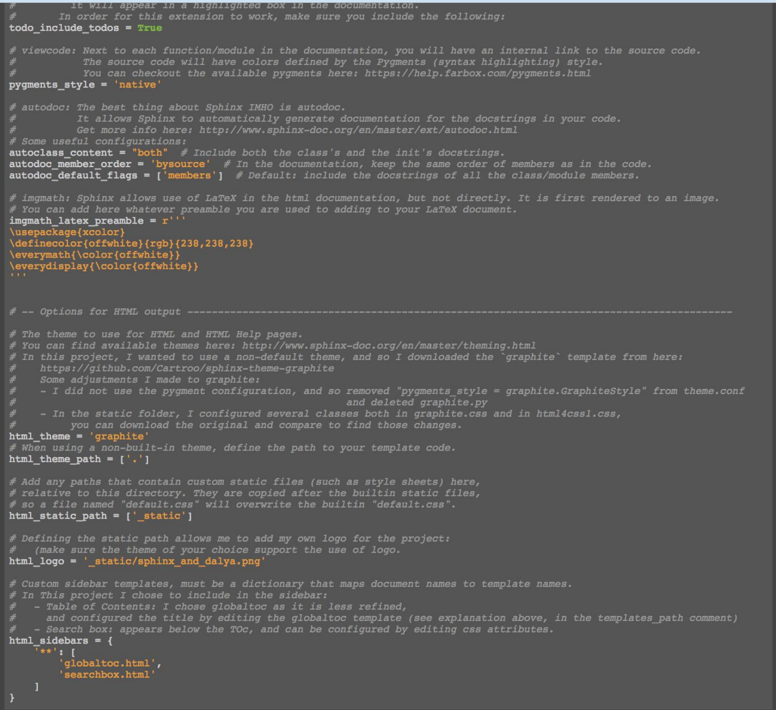
Part II - Main configuration file - conf.py



```
# -- Define and configure non-default extensions ----------------------------------------------------------------------

# You can find a list of available extension here: http://www.sphinx-doc.org/en/master/extensions.html
extensions = ['sphinx.ext.todo', 'sphinx.ext.viewcode', 'sphinx.ext.autodoc', 'sphinx.ext.imgmath']

# Above extensions explanation and configurations:

# todo: When you use the syntax ".. todo:: some todo comment" in your docstring,
#         it will appear in a highlighted box in the documentation.
#       In order for this extension to work, make sure you include the following:
todo_include_todos = True

# viewcode: Next to each function/module in the documentation, you will have an internal link to the source code.
#           The source code will have colors defined by the Pygments (syntax highlighting) style.
#           You can checkout the available pygments here: https://help.farbox.com/pygments.html
pygments_style = 'native'

# autodoc: The best thing about Sphinx IMHO is autodoc.
#          It allows Sphinx to automatically generate documentation for the docstrings in your code.
#          Get more info here: http://www.sphinx-doc.org/en/master/ext/autodoc.html
# Some useful configurations:
autoclass_content = "both"  # Include both the class's and the init's docstrings.
autodoc_member_order = 'bysource'  # In the documentation, keep the same order of members as in the code.
autodoc_default_flags = ['members']  # Default: include the docstrings of all the class/module members.

# imgmath: Sphinx allows use of LaTeX in the html documentation, but not directly. It is first rendered to an image.
# You can add here whatever preamble you are used to adding to your LaTeX document.
imgmath_latex_preamble = r'''
\usepackage{xcolor}
\definecolor{offwhite}{rgb}{238,238,238}
\everymath{\color{offwhite}}
\everydisplay{\color{offwhite}}
'''


# -- Options for HTML output ------------------------------------------------------------------------------------------

# The theme to use for HTML and HTML Help pages.
# You can find available themes here: http://www.sphinx-doc.org/en/master/theming.html
# In this project, I wanted to use a non-default theme, and so I downloaded the `graphite` template from here:
#    https://github.com/Cartroo/sphinx-theme-graphite
#    Some adjustments I made to graphite:
#    - I did not use the pygment configuration, and so removed "pygments_style = graphite.GraphiteStyle" from theme.conf
#                                                       and deleted graphite.py
#    - In the static folder, I configured several classes both in graphite.css and in html4css1.css,
#         you can download the original and compare to find those changes.
html_theme = 'graphite'
# When using a non-built-in theme, define the path to your template code.
html_theme_path = ['.']

# Add any paths that contain custom static files (such as style sheets) here,
# relative to this directory. They are copied after the builtin static files,
# so a file named "default.css" will overwrite the builtin "default.css".
html_static_path = ['_static']

# Defining the static path allows me to add my own logo for the project:
#   (make sure the theme of your choice support the use of logo.
html_logo = '_static/sphinx_and_dalya.png'

# Custom sidebar templates, must be a dictionary that maps document names to template names.
# In This project I chose to include in the sidebar:
#   - Table of Contents: I chose globaltoc as it is less refined,
#     and configured the title by editing the globaltoc template (see explanation above, in the templates_path comment)
#   - Search box: appears below the TOc, and can be configured by editing css attributes.
html_sidebars = {
    '**': [
        'globaltoc.html',
        'searchbox.html'
    ]
}
```

Part II - Main configuration file - conf.py

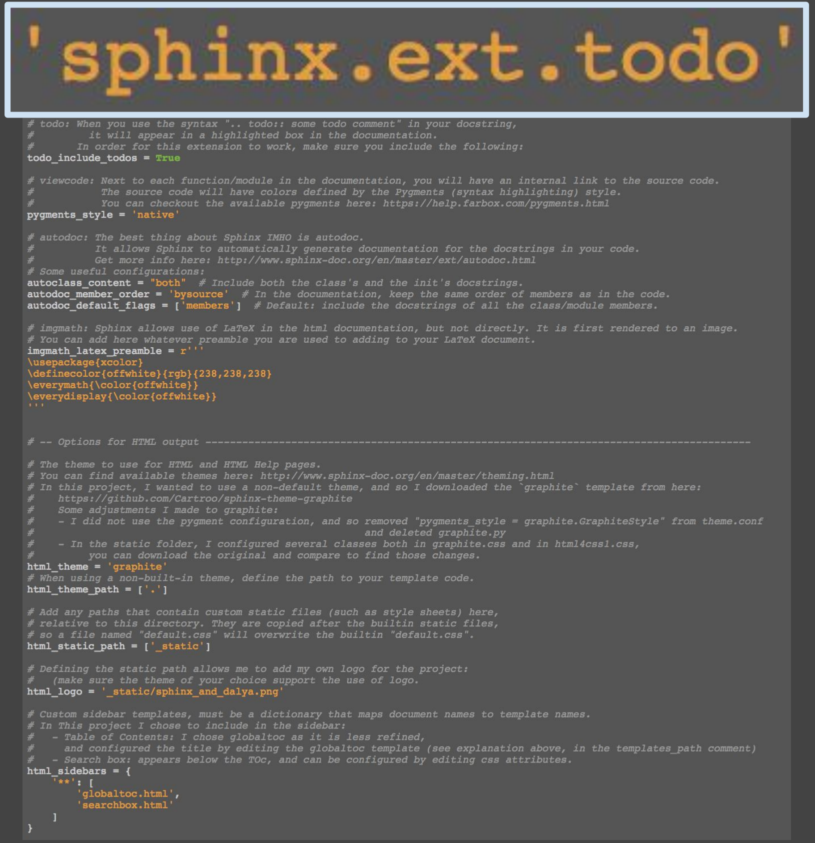
```
extensions = ['sphinx.ext.todo', 'sphinx.ext.viewcode', 'sphinx.ext.autodoc',
'sphinx.ext.imgmath']
```



```
#       It will appear in a highlighted box in the documentation.
#       In order for this extension to work, make sure you include the following:
todo_include_todos = True

# viewcode: Next to each function/module in the documentation, you will have an internal link to the source code.
#           The source code will have colors defined by the Pygments (syntax highlighting) style.
#           You can checkout the available pygments here: https://help.farbox.com/pygments.html
pygments_style = 'native'

# autodoc: The best thing about Sphinx IMHO is autodoc.
#          It allows Sphinx to automatically generate documentation for the docstrings in your code.
#          Get more info here: http://www.sphinx-doc.org/en/master/ext/autodoc.html
# Some useful configurations:
autoclass_content = "both"  # Include both the class's and the init's docstrings.
autodoc_member_order = 'bysource'  # In the documentation, keep the same order of members as in the code.
autodoc_default_flags = ['members']  # Default: include the docstrings of all the class/module members.

# imgmath: Sphinx allows use of LaTeX in the html documentation, but not directly. It is first rendered to an image.
# You can add here whatever preamble you are used to adding to your LaTeX document.
imgmath_latex_preamble = r'''
\usepackage{xcolor}
\definecolor{offwhite}{rgb}{238,238,238}
\everymath{\color{offwhite}}
\everydisplay{\color{offwhite}}
'''


# -- Options for HTML output -------------------------------------------------------------------------------------

# The theme to use for HTML and HTML Help pages.
# You can find available themes here: http://www.sphinx-doc.org/en/master/theming.html
# In this project, I wanted to use a non-default theme, and so I downloaded the `graphite` template from here:
#    https://github.com/Cartroo/sphinx-theme-graphite
#    Some adjustments I made to graphite:
#    - I did not use the pygment configuration, and so removed "pygments_style = graphite.GraphiteStyle" from theme.conf
#                                                    and deleted graphite.py
#    - In the static folder, I configured several classes both in graphite.css and in html4css1.css,
#         you can download the original and compare to find those changes.
html_theme = 'graphite'
# When using a non-built-in theme, define the path to your template code.
html_theme_path = ['.']

# Add any paths that contain custom static files (such as style sheets) here,
# relative to this directory. They are copied after the builtin static files,
# so a file named "default.css" will overwrite the builtin "default.css".
html_static_path = ['_static']

# Defining the static path allows me to add my own logo for the project:
#   (make sure the theme of your choice support the use of logo.
html_logo = '_static/sphinx_and_dalya.png'

# Custom sidebar templates, must be a dictionary that maps document names to template names.
# In This project I chose to include in the sidebar:
#   - Table of Contents: I chose globaltoc as it is less refined,
#     and configured the title by editing the globaltoc template (see explanation above, in the templates_path comment)
#   - Search box: appears below the TOc, and can be configured by editing css attributes.
html_sidebars = {
    '**': [
        'globaltoc.html',
        'searchbox.html'
    ]
}
```

Let Sphinx know what extentions to use

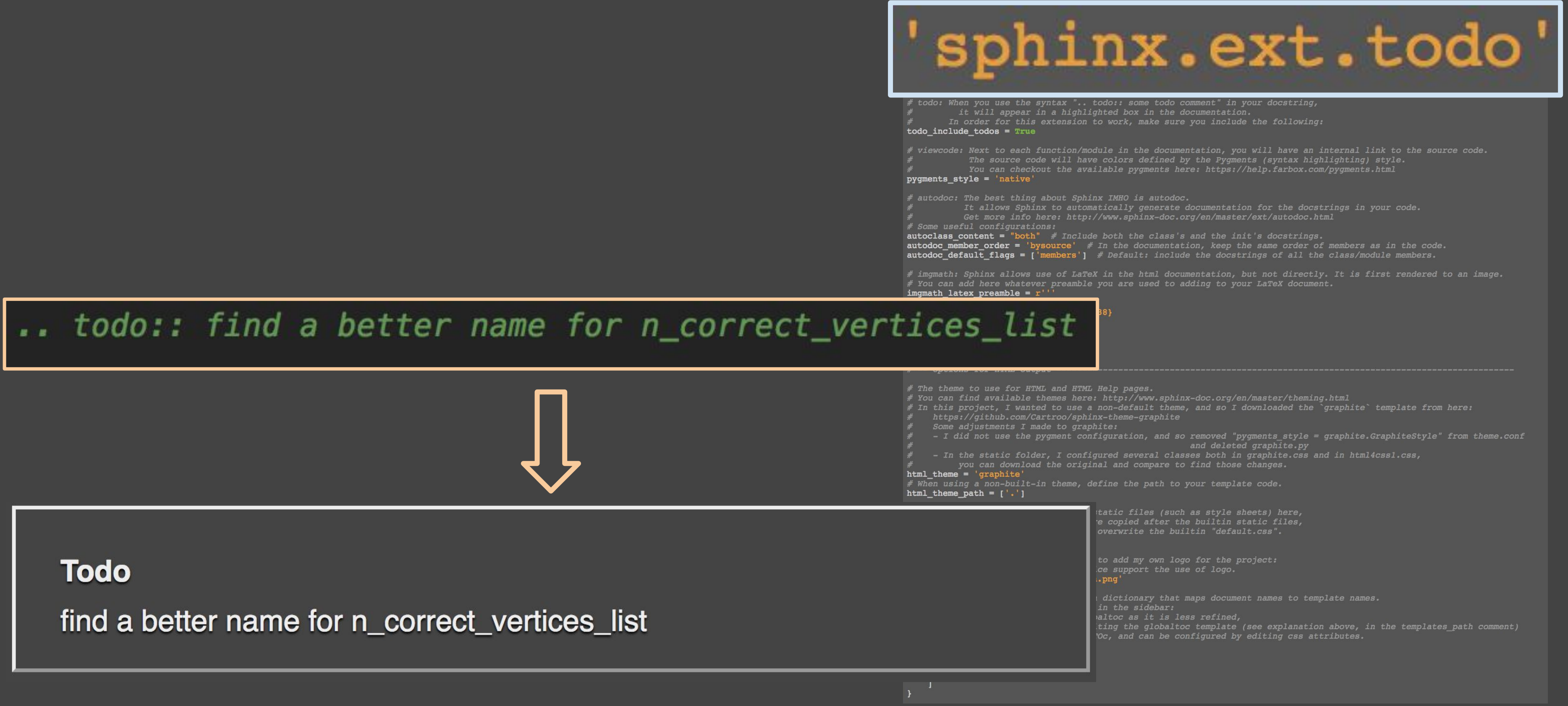
Part II - Main configuration file - conf.py



'sphinx.ext.todo'

```
# todo: When you use the syntax ".. todo:: some todo comment" in your docstring,
#         it will appear in a highlighted box in the documentation.
#       In order for this extension to work, make sure you include the following:
todo_include_todos = True

# viewcode: Next to each function/module in the documentation, you will have an internal link to the source code.
#           The source code will have colors defined by the Pygments (syntax highlighting) style.
#           You can checkout the available pygments here: https://help.farbox.com/pygments.html
pygments_style = 'native'

# autodoc: The best thing about Sphinx IMHO is autodoc.
#          It allows Sphinx to automatically generate documentation for the docstrings in your code.
#          Get more info here: http://www.sphinx-doc.org/en/master/ext/autodoc.html
# Some useful configurations:
autoclass_content = "both"  # Include both the class's and the init's docstrings.
autodoc_member_order = 'bysource'  # In the documentation, keep the same order of members as in the code.
autodoc_default_flags = ['members']  # Default: include the docstrings of all the class/module members.

# imgmath: Sphinx allows use of LaTeX in the html documentation, but not directly. It is first rendered to an image.
# You can add here whatever preamble you are used to adding to your LaTeX document.
imgmath_latex_preamble = r'''
\usepackage{xcolor}
\definecolor{offwhite}{rgb}{238,238,238}
\everymath{\color{offwhite}}
\everydisplay{\color{offwhite}}
'''


# -- Options for HTML output -------------------------------------------------------------------------

# The theme to use for HTML and HTML Help pages.
# You can find available themes here: http://www.sphinx-doc.org/en/master/theming.html
# In this project, I wanted to use a non-default theme, and so I downloaded the `graphite` template from here:
#    https://github.com/Cartroo/sphinx-theme-graphite
#    Some adjustments I made to graphite:
#    - I did not use the pygment configuration, and so removed "pygments_style = graphite.GraphiteStyle" from theme.conf
#                                                          and deleted graphite.py
#    - In the static folder, I configured several classes both in graphite.css and in html4css1.css,
#         you can download the original and compare to find those changes.
html_theme = 'graphite'
# When using a non-built-in theme, define the path to your template code.
html_theme_path = ['.']

# Add any paths that contain custom static files (such as style sheets) here,
# relative to this directory. They are copied after the builtin static files,
# so a file named "default.css" will overwrite the builtin "default.css".
html_static_path = ['_static']

# Defining the static path allows me to add my own logo for the project:
#   (make sure the theme of your choice support the use of logo.
html_logo = '_static/sphinx_and_dalya.png'

# Custom sidebar templates, must be a dictionary that maps document names to template names.
# In This project I chose to include in the sidebar:
#   - Table of Contents: I chose globaltoc as it is less refined,
#     and configured the title by editing the globaltoc template (see explanation above, in the templates_path comment)
#   - Search box: appears below the TOc, and can be configured by editing css attributes.
html_sidebars = {
    '**': [
        'globaltoc.html',
        'searchbox.html'
    ]
}
```

Part II - Main configuration file - conf.py



```
'sphinx.ext.todo'
```

```
# todo: When you use the syntax ".. todo:: some todo comment" in your docstring,
#         it will appear in a highlighted box in the documentation.
#       In order for this extension to work, make sure you include the following:
todo_include_todos = True

# viewcode: Next to each function/module in the documentation, you will have an internal link to the source code.
#            The source code will have colors defined by the Pygments (syntax highlighting) style.
#            You can checkout the available pygments here: https://help.farbox.com/pygments.html
pygments_style = 'native'

# autodoc: The best thing about Sphinx IMHO is autodoc.
#           It allows Sphinx to automatically generate documentation for the docstrings in your code.
#           Get more info here: http://www.sphinx-doc.org/en/master/ext/autodoc.html
# Some useful configurations:
autoclass_content = "both"  # Include both the class's and the init's docstrings.
autodoc_member_order = 'bysource'  # In the documentation, keep the same order of members as in the code.
autodoc_default_flags = ['members']  # Default: include the docstrings of all the class/module members.

# imgmath: Sphinx allows use of LaTeX in the html documentation, but not directly. It is first rendered to an image.
# You can add here whatever preamble you are used to adding to your LaTeX document.
imgmath_latex_preamble = r'''
                                            38}

# ---------------------------------------------------------------------------------------------------------------

# The theme to use for HTML and HTML Help pages.
# You can find available themes here: http://www.sphinx-doc.org/en/master/theming.html
# In this project, I wanted to use a non-default theme, and so I downloaded the `graphite` template from here:
#    https://github.com/Cartroo/sphinx-theme-graphite
#    Some adjustments I made to graphite:
#    - I did not use the pygment configuration, and so removed "pygments_style = graphite.GraphiteStyle" from theme.conf
#                                                 and deleted graphite.py
#    - In the static folder, I configured several classes both in graphite.css and in html4css1.css,
#         you can download the original and compare to find those changes.
html_theme = 'graphite'
# When using a non-built-in theme, define the path to your template code.
html_theme_path = ['.']

                                             tatic files (such as style sheets) here,
                                             e copied after the builtin static files,
                                             overwrite the builtin "default.css".

                                             to add my own logo for the project:
                                             ce support the use of logo.
                                             .png'

                                             dictionary that maps document names to template names.
                                             in the sidebar:
                                             altoc as it is less refined,
                                             ting the globaltoc template (see explanation above, in the templates_path comment)
                                             Oc, and can be configured by editing css attributes.

    ]
}
```

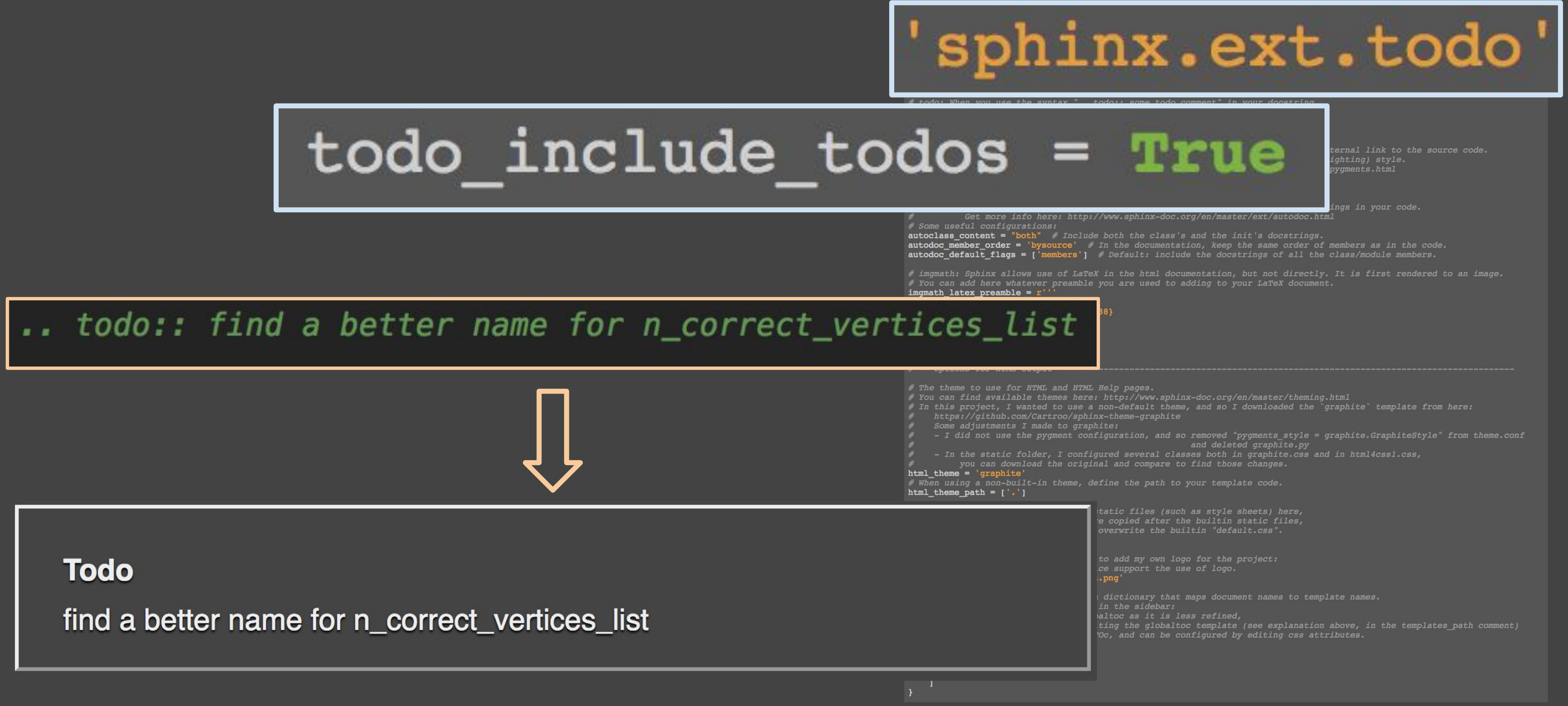
```
.. todo:: find a better name for n_correct_vertices_list
```

⇩

Todo

find a better name for n_correct_vertices_list

Part II - Main configuration file - conf.py



```
'sphinx.ext.todo'
```

```
todo_include_todos = True
```

```
.. todo:: find a better name for n_correct_vertices_list
```

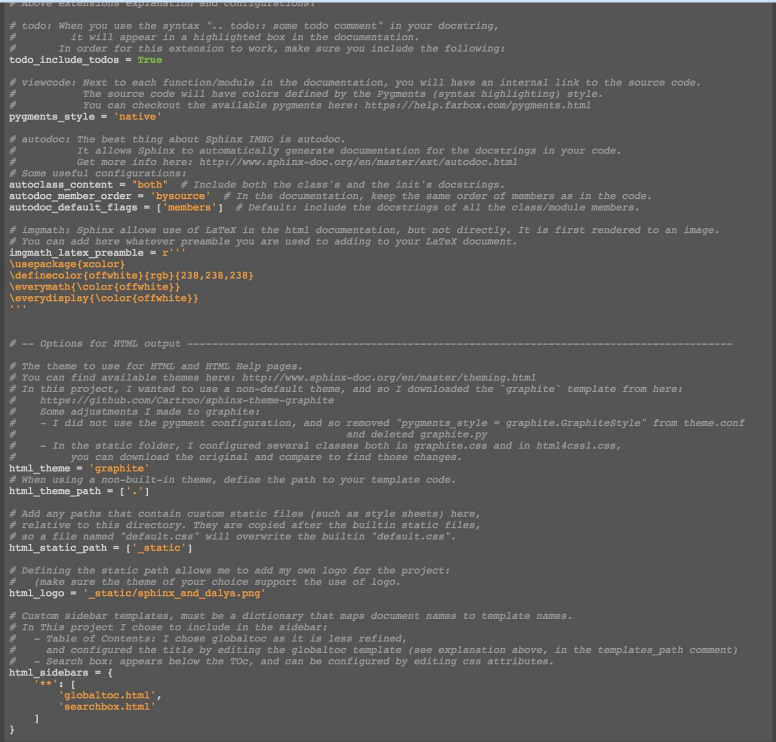
Todo

find a better name for n_correct_vertices_list

```
# todo: When you use the syntax ".. todo:: some todo comment" in your docstring
ternal link to the source code.
ighting) style.
pygments.html

ings in your code.
#          Get more info here: http://www.sphinx-doc.org/en/master/ext/autodoc.html
# Some useful configurations:
autoclass_content = "both"  # Include both the class's and the init's docstrings.
autodoc_member_order = 'bysource'  # In the documentation, keep the same order of members as in the code.
autodoc_default_flags = ['members']  # Default: include the docstrings of all the class/module members.

# imgmath: Sphinx allows use of LaTeX in the html documentation, but not directly. It is first rendered to an image.
# You can add here whatever preamble you are used to adding to your LaTeX document.
imgmath_latex_preamble = r'''
38}

# The theme to use for HTML and HTML Help pages.
# You can find available themes here: http://www.sphinx-doc.org/en/master/theming.html
# In this project, I wanted to use a non-default theme, and so I downloaded the `graphite` template from here:
#    https://github.com/Cartroo/sphinx-theme-graphite
#    Some adjustments I made to graphite:
#    - I did not use the pygment configuration, and so removed "pygments_style = graphite.GraphiteStyle" from theme.conf
#                                          and deleted graphite.py
#    - In the static folder, I configured several classes both in graphite.css and in html4css1.css,
#         you can download the original and compare to find those changes.
html_theme = 'graphite'
# When using a non-built-in theme, define the path to your template code.
html_theme_path = ['.']

tatic files (such as style sheets) here,
e copied after the builtin static files,
overwrite the builtin "default.css".

to add my own logo for the project:
ce support the use of logo.
.png'

dictionary that maps document names to template names.
in the sidebar:
altoc as it is less refined,
ting the globaltoc template (see explanation above, in the templates_path comment)
Oc, and can be configured by editing css attributes.

    ]
}
```

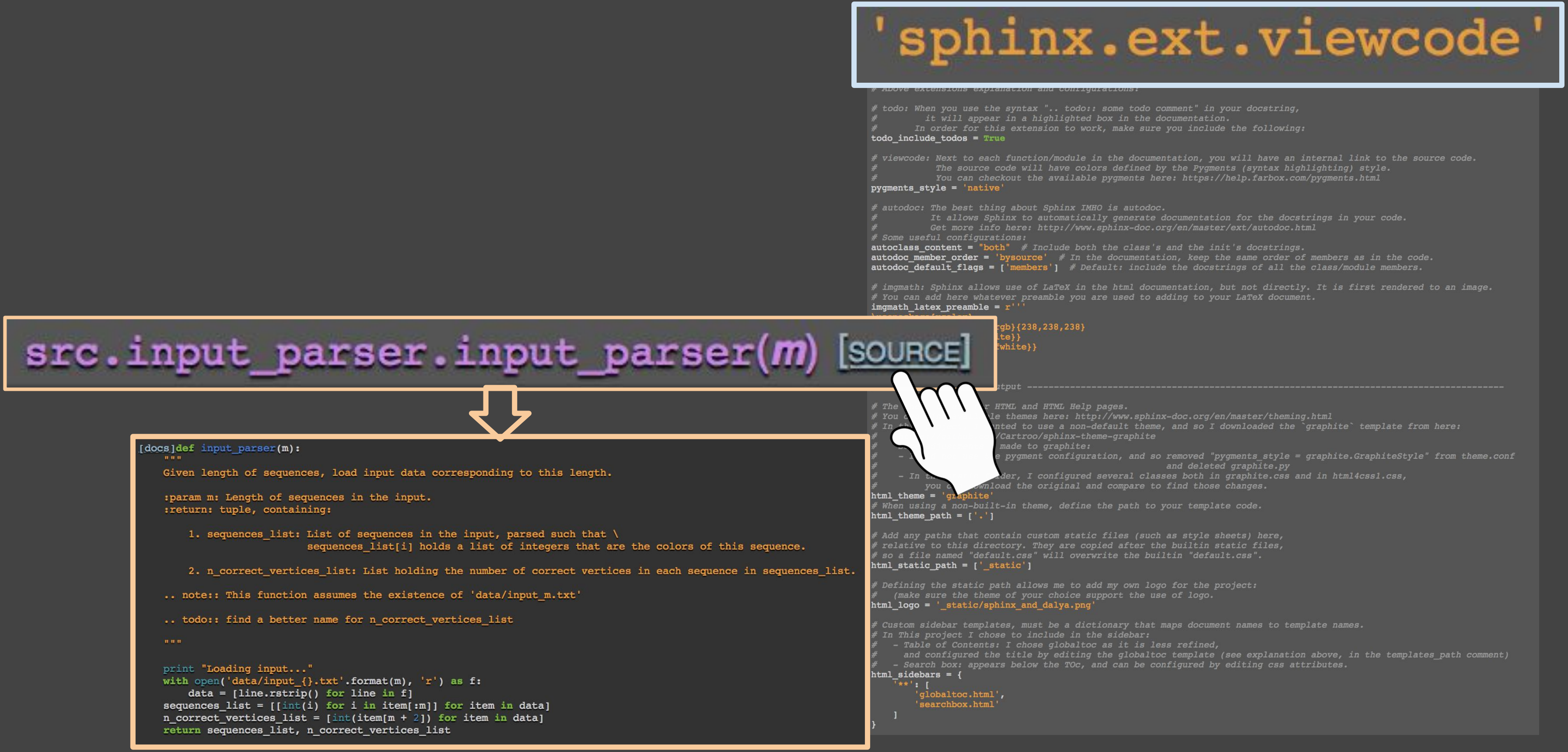
Part II - Main configuration file - conf.py

'sphinx.ext.viewcode'



```
# Above extensions explanation and configurations:

# todo: When you use the syntax ".. todo:: some todo comment" in your docstring,
#         it will appear in a highlighted box in the documentation.
#       In order for this extension to work, make sure you include the following:
todo_include_todos = True

# viewcode: Next to each function/module in the documentation, you will have an internal link to the source code.
#           The source code will have colors defined by the Pygments (syntax highlighting) style.
#           You can checkout the available pygments here: https://help.farbox.com/pygments.html
pygments_style = 'native'

# autodoc: The best thing about Sphinx IMHO is autodoc.
#          It allows Sphinx to automatically generate documentation for the docstrings in your code.
#          Get more info here: http://www.sphinx-doc.org/en/master/ext/autodoc.html
# Some useful configurations:
autoclass_content = "both"  # Include both the class's and the init's docstrings.
autodoc_member_order = 'bysource'  # In the documentation, keep the same order of members as in the code.
autodoc_default_flags = ['members']  # Default: include the docstrings of all the class/module members.

# imgmath: Sphinx allows use of LaTeX in the html documentation, but not directly. It is first rendered to an image.
# You can add here whatever preamble you are used to adding to your LaTeX document.
imgmath_latex_preamble = r'''
\usepackage{xcolor}
\definecolor{offwhite}{rgb}{238,238,238}
\everymath{\color{offwhite}}
\everydisplay{\color{offwhite}}
'''


# -- Options for HTML output ----------------------------------------------------------------------------------

# The theme to use for HTML and HTML Help pages.
# You can find available themes here: http://www.sphinx-doc.org/en/master/theming.html
# In this project, I wanted to use a non-default theme, and so I downloaded the `graphite` template from here:
#    https://github.com/Cartroo/sphinx-theme-graphite
#    Some adjustments I made to graphite:
#    - I did not use the pygment configuration, and so removed "pygments_style = graphite.GraphiteStyle" from theme.conf
#                                                        and deleted graphite.py
#    - In the static folder, I configured several classes both in graphite.css and in html4css1.css,
#        you can download the original and compare to find those changes.
html_theme = 'graphite'
# When using a non-built-in theme, define the path to your template code.
html_theme_path = ['.']

# Add any paths that contain custom static files (such as style sheets) here,
# relative to this directory. They are copied after the builtin static files,
# so a file named "default.css" will overwrite the builtin "default.css".
html_static_path = ['_static']

# Defining the static path allows me to add my own logo for the project:
#   (make sure the theme of your choice support the use of logo.
html_logo = '_static/sphinx_and_dalya.png'

# Custom sidebar templates, must be a dictionary that maps document names to template names.
# In This project I chose to include in the sidebar:
#   - Table of Contents: I chose globaltoc as it is less refined,
#     and configured the title by editing the globaltoc template (see explanation above, in the templates_path comment)
#   - Search box: appears below the TOc, and can be configured by editing css attributes.
html_sidebars = {
    '**': [
        'globaltoc.html',
        'searchbox.html'
    ]
}
```

Part II - Main configuration file - conf.py



```
'sphinx.ext.viewcode'
```

```
# Above extensions explanation and configurations:

# todo: When you use the syntax ".. todo:: some todo comment" in your docstring,
#         it will appear in a highlighted box in the documentation.
#       In order for this extension to work, make sure you include the following:
todo_include_todos = True

# viewcode: Next to each function/module in the documentation, you will have an internal link to the source code.
#           The source code will have colors defined by the Pygments (syntax highlighting) style.
#           You can checkout the available pygments here: https://help.farbox.com/pygments.html
pygments_style = 'native'

# autodoc: The best thing about Sphinx IMHO is autodoc.
#          It allows Sphinx to automatically generate documentation for the docstrings in your code.
#          Get more info here: http://www.sphinx-doc.org/en/master/ext/autodoc.html
# Some useful configurations:
autoclass_content = "both"  # Include both the class's and the init's docstrings.
autodoc_member_order = 'bysource'  # In the documentation, keep the same order of members as in the code.
autodoc_default_flags = ['members']  # Default: include the docstrings of all the class/module members.

# imgmath: Sphinx allows use of LaTeX in the html documentation, but not directly. It is first rendered to an image.
# You can add here whatever preamble you are used to adding to your LaTeX document.
imgmath_latex_preamble = r'''
                  rgb}{238,238,238}
                  ite}}
                  fwhite}}


                  tput ----------------------------------------------------------------------------------------

# The             r HTML and HTML Help pages.
# You c           le themes here: http://www.sphinx-doc.org/en/master/theming.html
# In th           nted to use a non-default theme, and so I downloaded the `graphite` template from here:
#                 /Cartroo/sphinx-theme-graphite
#                  made to graphite:
#    - I          e pygment configuration, and so removed "pygments_style = graphite.GraphiteStyle" from theme.conf
#                                                    and deleted graphite.py
#    - In th        der, I configured several classes both in graphite.css and in html4css1.css,
#         you c     wnload the original and compare to find those changes.
html_theme = 'graphite'
# When using a non-built-in theme, define the path to your template code.
html_theme_path = ['.']

# Add any paths that contain custom static files (such as style sheets) here,
# relative to this directory. They are copied after the builtin static files,
# so a file named "default.css" will overwrite the builtin "default.css".
html_static_path = ['_static']

# Defining the static path allows me to add my own logo for the project:
#   (make sure the theme of your choice support the use of logo.
html_logo = '_static/sphinx_and_dalya.png'

# Custom sidebar templates, must be a dictionary that maps document names to template names.
# In This project I chose to include in the sidebar:
#   - Table of Contents: I chose globaltoc as it is less refined,
#     and configured the title by editing the globaltoc template (see explanation above, in the templates_path comment)
#   - Search box: appears below the TOc, and can be configured by editing css attributes.
html_sidebars = {
    '**': [
        'globaltoc.html',
        'searchbox.html'
    ]
}
```

```
src.input_parser.input_parser(m) [SOURCE]
```

```
[docs]def input_parser(m):
    """
    Given length of sequences, load input data corresponding to this length.

    :param m: Length of sequences in the input.
    :return: tuple, containing:

        1. sequences_list: List of sequences in the input, parsed such that \
                           sequences_list[i] holds a list of integers that are the colors of this sequence.

        2. n_correct_vertices_list: List holding the number of correct vertices in each sequence in sequences_list.

    .. note:: This function assumes the existence of 'data/input_m.txt'

    .. todo:: find a better name for n_correct_vertices_list

    """

    print "Loading input..."
    with open('data/input_{}.txt'.format(m), 'r') as f:
        data = [line.rstrip() for line in f]
    sequences_list = [[int(i) for i in item[:m]] for item in data]
    n_correct_vertices_list = [int(item[m + 2]) for item in data]
    return sequences_list, n_correct_vertices_list
```

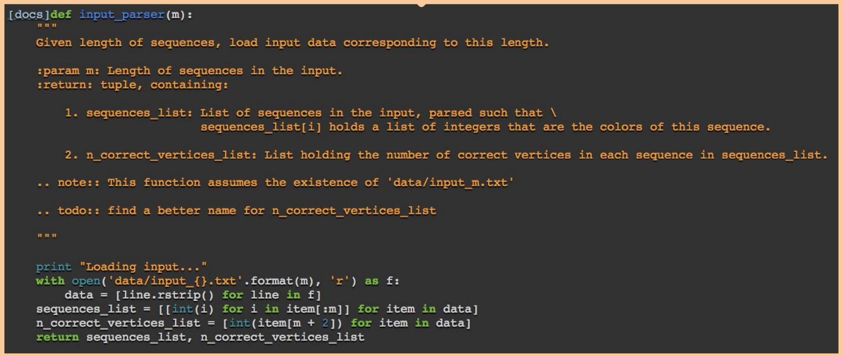
Part II - Main configuration file - conf.py

'sphinx.ext.viewcode'

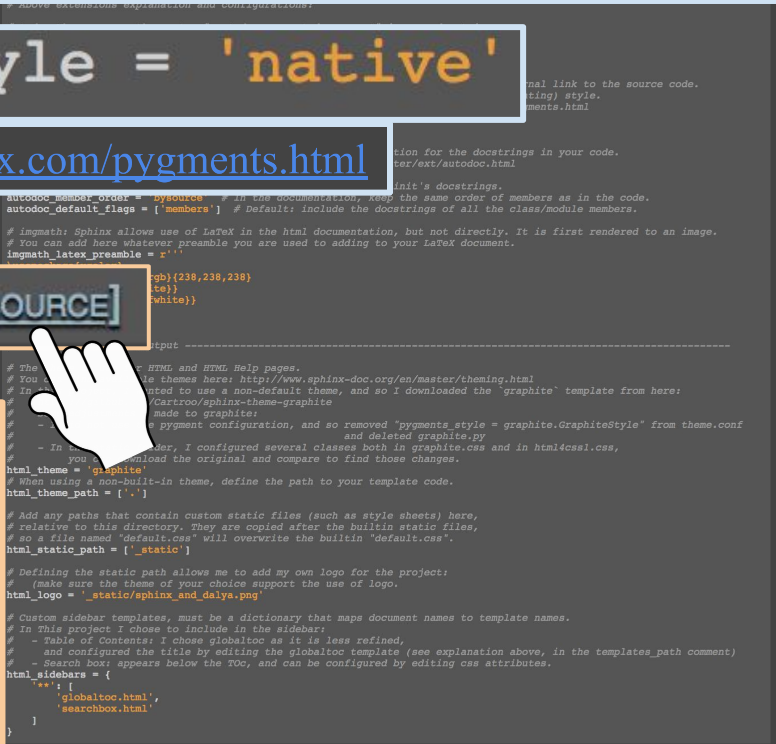
pygments_style = 'native'

https://help.farbox.com/pygments.html

src.input_parser.input_parser(*m*) [SOURCE]



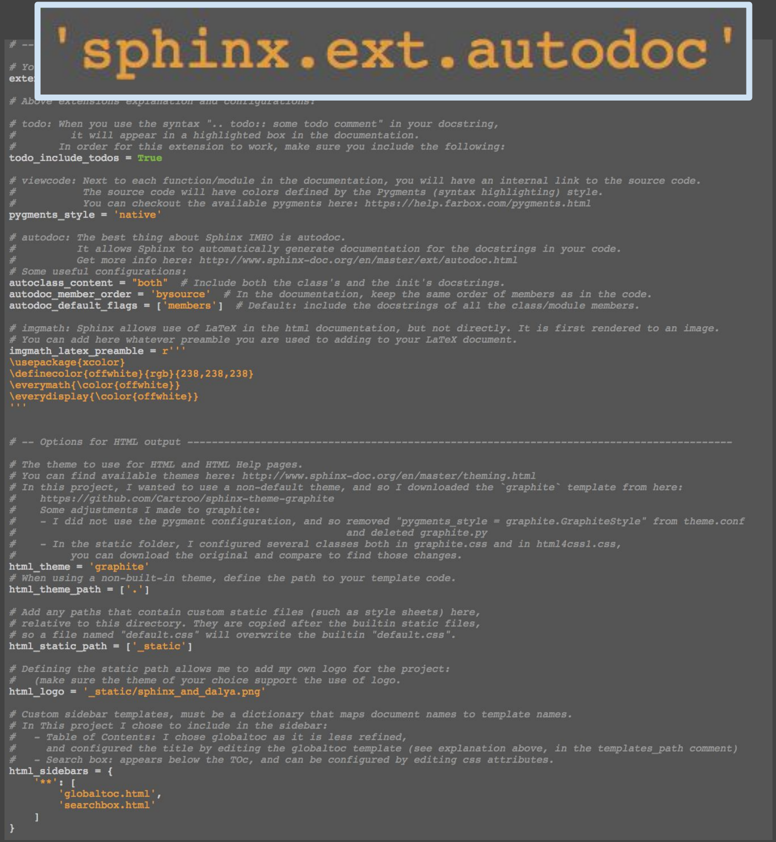
```
[docs]def input_parser(m):
    """
    Given length of sequences, load input data corresponding to this length.

    :param m: Length of sequences in the input.
    :return: tuple, containing:

        1. sequences_list: List of sequences in the input, parsed such that \
                           sequences_list[i] holds a list of integers that are the colors of this sequence.

        2. n_correct_vertices_list: List holding the number of correct vertices in each sequence in sequences_list.

    .. note:: This function assumes the existence of 'data/input_m.txt'

    .. todo:: find a better name for n_correct_vertices_list

    """

    print "Loading input..."
    with open('data/input_{}.txt'.format(m), 'r') as f:
        data = [line.rstrip() for line in f]
    sequences_list = [[int(i) for i in item[:m]] for item in data]
    n_correct_vertices_list = [int(item[m + 2]) for item in data]
    return sequences_list, n_correct_vertices_list
```



```
# Above extensions explanation and configurations:

                                                  rnal link to the source code.
                                                  ting) style.
                                                  ments.html

                                                  tion for the docstrings in your code.
                                                  ter/ext/autodoc.html

                                                  init's docstrings.
autodoc_member_order = 'bysource'  # In the documentation, keep the same order of members as in the code.
autodoc_default_flags = ['members']  # Default: include the docstrings of all the class/module members.

# imgmath: Sphinx allows use of LaTeX in the html documentation, but not directly. It is first rendered to an image.
# You can add here whatever preamble you are used to adding to your LaTeX document.
imgmath_latex_preamble = r'''

           rgb}{238,238,238}
           ite}}
           fwhite}}


           tput ------------------------------------------------------------------------------------

# The          r HTML and HTML Help pages.
# You c        le themes here: http://www.sphinx-doc.org/en/master/theming.html
# In th        nted to use a non-default theme, and so I downloaded the `graphite` template from here:
#              /Cartroo/sphinx-theme-graphite
#              made to graphite:
#   - I        e pygment configuration, and so removed "pygments_style = graphite.GraphiteStyle" from theme.conf
#                                                 and deleted graphite.py
#   - In t     der, I configured several classes both in graphite.css and in html4css1.css,
#         you c  wnload the original and compare to find those changes.
html_theme = 'graphite'
# When using a non-built-in theme, define the path to your template code.
html_theme_path = ['.']

# Add any paths that contain custom static files (such as style sheets) here,
# relative to this directory. They are copied after the builtin static files,
# so a file named "default.css" will overwrite the builtin "default.css".
html_static_path = ['_static']

# Defining the static path allows me to add my own logo for the project:
#   (make sure the theme of your choice support the use of logo.
html_logo = '_static/sphinx_and_dalya.png'

# Custom sidebar templates, must be a dictionary that maps document names to template names.
# In This project I chose to include in the sidebar:
#   - Table of Contents: I chose globaltoc as it is less refined,
#     and configured the title by editing the globaltoc template (see explanation above, in the templates_path comment)
#   - Search box: appears below the TOc, and can be configured by editing css attributes.
html_sidebars = {
    '**': [
        'globaltoc.html',
        'searchbox.html'
    ]
}
```

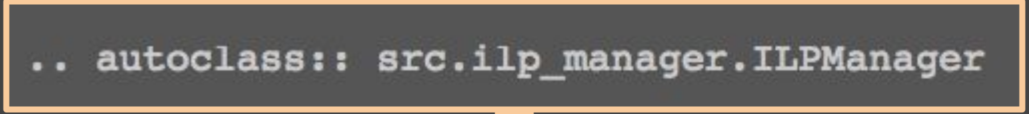
Part II - Main configuration file - conf.py

'sphinx.ext.autodoc'

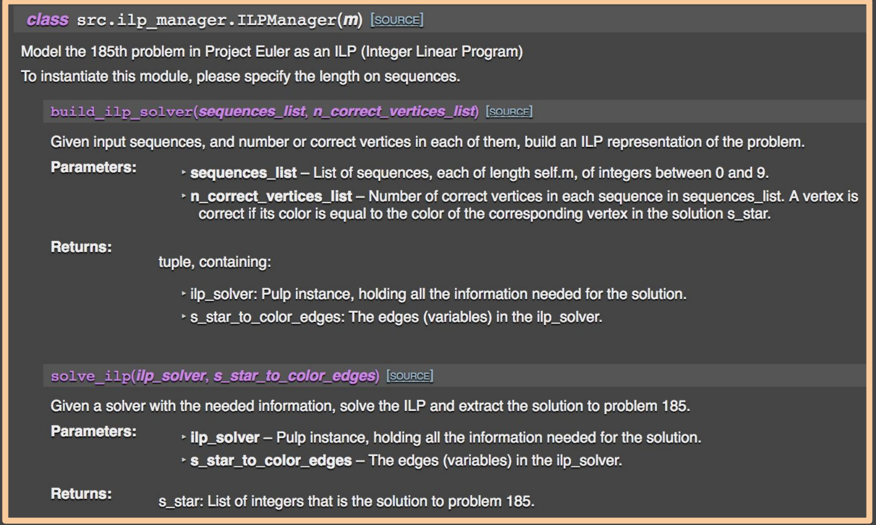


```
# --

# Yo
exte

# Above extensions explanation and configurations:

# todo: When you use the syntax ".. todo:: some todo comment" in your docstring,
#         it will appear in a highlighted box in the documentation.
#       In order for this extension to work, make sure you include the following:
todo_include_todos = True

# viewcode: Next to each function/module in the documentation, you will have an internal link to the source code.
#           The source code will have colors defined by the Pygments (syntax highlighting) style.
#           You can checkout the available pygments here: https://help.farbox.com/pygments.html
pygments_style = 'native'

# autodoc: The best thing about Sphinx IMHO is autodoc.
#          It allows Sphinx to automatically generate documentation for the docstrings in your code.
#          Get more info here: http://www.sphinx-doc.org/en/master/ext/autodoc.html
# Some useful configurations:
autoclass_content = "both"  # Include both the class's and the init's docstrings.
autodoc_member_order = 'bysource'  # In the documentation, keep the same order of members as in the code.
autodoc_default_flags = ['members']  # Default: include the docstrings of all the class/module members.

# imgmath: Sphinx allows use of LaTeX in the html documentation, but not directly. It is first rendered to an image.
# You can add here whatever preamble you are used to adding to your LaTeX document.
imgmath_latex_preamble = r'''
\usepackage{xcolor}
\definecolor{offwhite}{rgb}{238,238,238}
\everymath{\color{offwhite}}
\everydisplay{\color{offwhite}}
'''


# -- Options for HTML output -------------------------------------------------------------------------------------

# The theme to use for HTML and HTML Help pages.
# You can find available themes here: http://www.sphinx-doc.org/en/master/theming.html
# In this project, I wanted to use a non-default theme, and so I downloaded the `graphite` template from here:
#    https://github.com/Cartroo/sphinx-theme-graphite
#    Some adjustments I made to graphite:
#    - I did not use the pygment configuration, and so removed "pygments_style = graphite.GraphiteStyle" from theme.conf
#                                                       and deleted graphite.py
#    - In the static folder, I configured several classes both in graphite.css and in html4css1.css,
#        you can download the original and compare to find those changes.
html_theme = 'graphite'
# When using a non-built-in theme, define the path to your template code.
html_theme_path = ['.']

# Add any paths that contain custom static files (such as style sheets) here,
# relative to this directory. They are copied after the builtin static files,
# so a file named "default.css" will overwrite the builtin "default.css".
html_static_path = ['_static']

# Defining the static path allows me to add my own logo for the project:
#   (make sure the theme of your choice support the use of logo.
html_logo = '_static/sphinx_and_dalya.png'

# Custom sidebar templates, must be a dictionary that maps document names to template names.
# In This project I chose to include in the sidebar:
#   - Table of Contents: I chose globaltoc as it is less refined,
#     and configured the title by editing the globaltoc template (see explanation above, in the templates_path comment)
#   - Search box: appears below the TOc, and can be configured by editing css attributes.
html_sidebars = {
    '**': [
        'globaltoc.html',
        'searchbox.html'
    ]
}
```

Part II - Main configuration file - conf.py



```
.. autoclass:: src.ilp_manager.ILPManager
```



class src.ilp_manager.ILPManager(*m*) [SOURCE]

Model the 185th problem in Project Euler as an ILP (Integer Linear Program)

To instantiate this module, please specify the length on sequences.

build_ilp_solver(*sequences_list*, *n_correct_vertices_list*) [SOURCE]

Given input sequences, and number or correct vertices in each of them, build an ILP representation of the problem.

Parameters:

- **sequences_list** – List of sequences, each of length self.m, of integers between 0 and 9.
- **n_correct_vertices_list** – Number of correct vertices in each sequence in sequences_list. A vertex is correct if its color is equal to the color of the corresponding vertex in the solution s_star.

Returns:

tuple, containing:

- ilp_solver: Pulp instance, holding all the information needed for the solution.
- s_star_to_color_edges: The edges (variables) in the ilp_solver.

solve_ilp(*ilp_solver*, *s_star_to_color_edges*) [SOURCE]

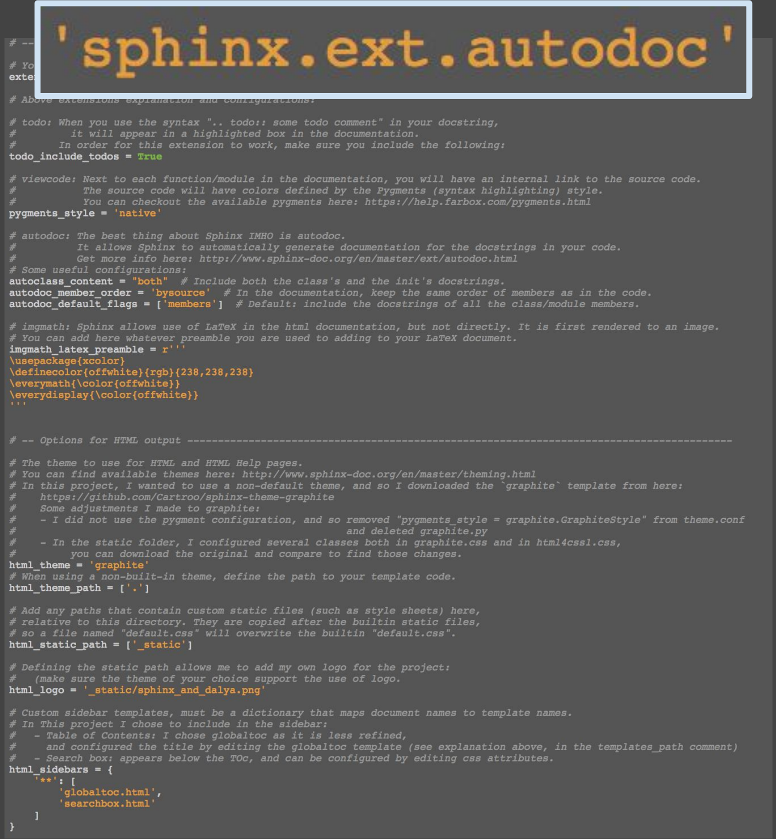
Given a solver with the needed information, solve the ILP and extract the solution to problem 185.

Parameters:

- **ilp_solver** – Pulp instance, holding all the information needed for the solution.
- **s_star_to_color_edges** – The edges (variables) in the ilp_solver.

Returns:

s_star: List of integers that is the solution to problem 185.



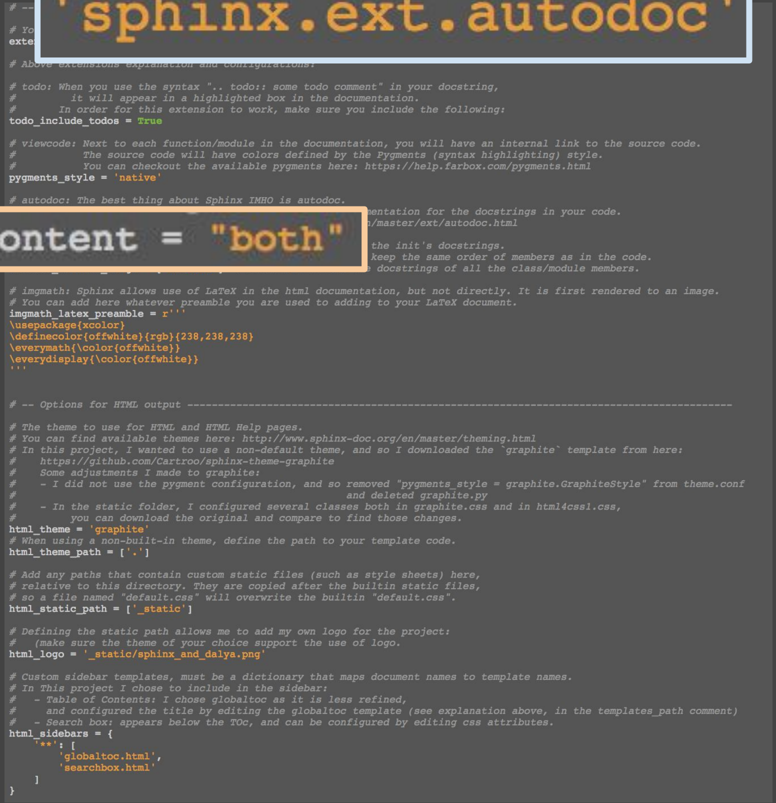
```
'sphinx.ext.autodoc'
```

```
# --
# Yo
exte

# Above extensions explanation and configurations:

# todo: When you use the syntax ".. todo:: some todo comment" in your docstring,
#        it will appear in a highlighted box in the documentation.
#       In order for this extension to work, make sure you include the following:
todo_include_todos = True

# viewcode: Next to each function/module in the documentation, you will have an internal link to the source code.
#           The source code will have colors defined by the Pygments (syntax highlighting) style.
#           You can checkout the available pygments here: https://help.farbox.com/pygments.html
pygments_style = 'native'

# autodoc: The best thing about Sphinx IMHO is autodoc.
#          It allows Sphinx to automatically generate documentation for the docstrings in your code.
#          Get more info here: http://www.sphinx-doc.org/en/master/ext/autodoc.html
# Some useful configurations:
autoclass_content = "both"  # Include both the class's and the init's docstrings.
autodoc_member_order = 'bysource'  # In the documentation, keep the same order of members as in the code.
autodoc_default_flags = ['members']  # Default: include the docstrings of all the class/module members.

# imgmath: Sphinx allows use of LaTeX in the html documentation, but not directly. It is first rendered to an image.
# You can add here whatever preamble you are used to adding to your LaTeX document.
imgmath_latex_preamble = r'''
\usepackage{xcolor}
\definecolor{offwhite}{rgb}{238,238,238}
\everymath{\color{offwhite}}
\everydisplay{\color{offwhite}}
'''


# -- Options for HTML output ----------------------------------------------------------------------------------

# The theme to use for HTML and HTML Help pages.
# You can find available themes here: http://www.sphinx-doc.org/en/master/theming.html
# In this project, I wanted to use a non-default theme, and so I downloaded the `graphite` template from here:
#    https://github.com/Cartroo/sphinx-theme-graphite
#    Some adjustments I made to graphite:
#    - I did not use the pygment configuration, and so removed "pygments_style = graphite.GraphiteStyle" from theme.conf
#                                                     and deleted graphite.py
#    - In the static folder, I configured several classes both in graphite.css and in html4css1.css,
#        you can download the original and compare to find those changes.
html_theme = 'graphite'
# When using a non-built-in theme, define the path to your template code.
html_theme_path = ['.']

# Add any paths that contain custom static files (such as style sheets) here,
# relative to this directory. They are copied after the builtin static files,
# so a file named "default.css" will overwrite the builtin "default.css".
html_static_path = ['_static']

# Defining the static path allows me to add my own logo for the project:
#   (make sure the theme of your choice support the use of logo.
html_logo = '_static/sphinx_and_dalya.png'

# Custom sidebar templates, must be a dictionary that maps document names to template names.
# In This project I chose to include in the sidebar:
#   - Table of Contents: I chose globaltoc as it is less refined,
#     and configured the title by editing the globaltoc template (see explanation above, in the templates_path comment)
#   - Search box: appears below the TOc, and can be configured by editing css attributes.
html_sidebars = {
    '**': [
        'globaltoc.html',
        'searchbox.html'
    ]
}
```

Part II - Main configuration file - conf.py

'sphinx.ext.autodoc'

autoclass_content = "both"

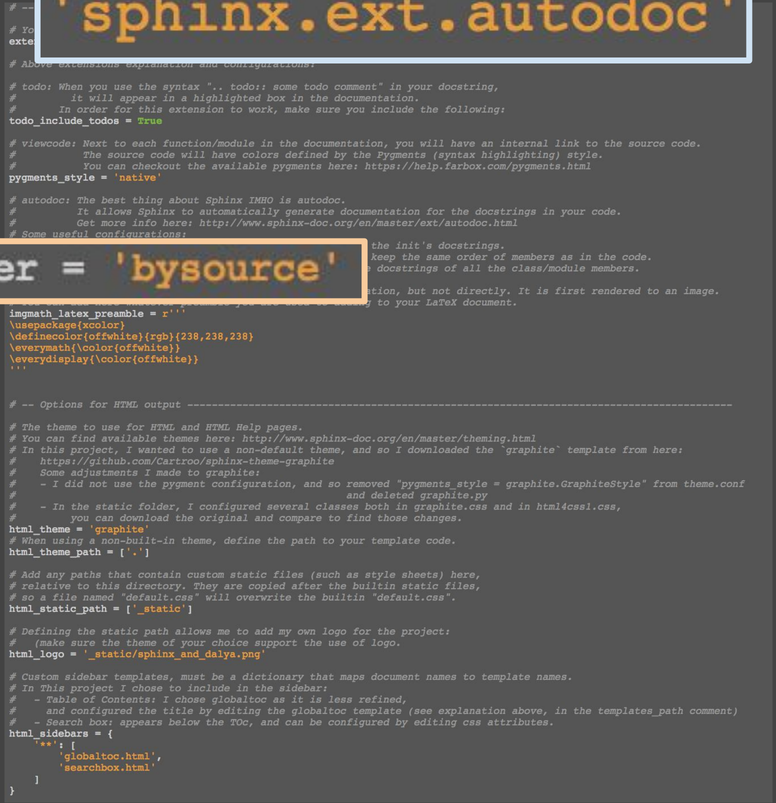


```
# --
# Yo
exte

# Above extensions explanation and configurations:

# todo: When you use the syntax ".. todo:: some todo comment" in your docstring,
#         it will appear in a highlighted box in the documentation.
#       In order for this extension to work, make sure you include the following:
todo_include_todos = True

# viewcode: Next to each function/module in the documentation, you will have an internal link to the source code.
#           The source code will have colors defined by the Pygments (syntax highlighting) style.
#           You can checkout the available pygments here: https://help.farbox.com/pygments.html
pygments_style = 'native'

# autodoc: The best thing about Sphinx IMHO is autodoc.
                                                     mentation for the docstrings in your code.
                                                     n/master/ext/autodoc.html

                                                      the init's docstrings.
                                                      keep the same order of members as in the code.
                                                     e docstrings of all the class/module members.

# imgmath: Sphinx allows use of LaTeX in the html documentation, but not directly. It is first rendered to an image.
# You can add here whatever preamble you are used to adding to your LaTeX document.
imgmath_latex_preamble = r'''
\usepackage{xcolor}
\definecolor{offwhite}{rgb}{238,238,238}
\everymath{\color{offwhite}}
\everydisplay{\color{offwhite}}
'''


# -- Options for HTML output -------------------------------------------------------------------------

# The theme to use for HTML and HTML Help pages.
# You can find available themes here: http://www.sphinx-doc.org/en/master/theming.html
# In this project, I wanted to use a non-default theme, and so I downloaded the `graphite` template from here:
#    https://github.com/Cartroo/sphinx-theme-graphite
#    Some adjustments I made to graphite:
#    - I did not use the pygment configuration, and so removed "pygments_style = graphite.GraphiteStyle" from theme.conf
#                                                         and deleted graphite.py
#    - In the static folder, I configured several classes both in graphite.css and in html4css1.css,
#         you can download the original and compare to find those changes.
html_theme = 'graphite'
# When using a non-built-in theme, define the path to your template code.
html_theme_path = ['.']

# Add any paths that contain custom static files (such as style sheets) here,
# relative to this directory. They are copied after the builtin static files,
# so a file named "default.css" will overwrite the builtin "default.css".
html_static_path = ['_static']

# Defining the static path allows me to add my own logo for the project:
#   (make sure the theme of your choice support the use of logo.
html_logo = '_static/sphinx_and_dalya.png'

# Custom sidebar templates, must be a dictionary that maps document names to template names.
# In This project I chose to include in the sidebar:
#   - Table of Contents: I chose globaltoc as it is less refined,
#     and configured the title by editing the globaltoc template (see explanation above, in the templates_path comment)
#   - Search box: appears below the TOc, and can be configured by editing css attributes.
html_sidebars = {
    '**': [
        'globaltoc.html',
        'searchbox.html'
    ]
}
```

Include both the class's and the init's docstrings

Part II - Main configuration file - conf.py

```
'sphinx.ext.autodoc'
```



```
# --

# Yo
exte

# Above extensions explanation and configurations:

# todo: When you use the syntax ".. todo:: some todo comment" in your docstring,
#          it will appear in a highlighted box in the documentation.
#        In order for this extension to work, make sure you include the following:
todo_include_todos = True

# viewcode: Next to each function/module in the documentation, you will have an internal link to the source code.
#            The source code will have colors defined by the Pygments (syntax highlighting) style.
#            You can checkout the available pygments here: https://help.farbox.com/pygments.html
pygments_style = 'native'

# autodoc: The best thing about Sphinx IMHO is autodoc.
#           It allows Sphinx to automatically generate documentation for the docstrings in your code.
#           Get more info here: http://www.sphinx-doc.org/en/master/ext/autodoc.html
# Some useful configurations:
                                                              the init's docstrings.
                                                              keep the same order of members as in the code.
                                                              e docstrings of all the class/module members.

                                                              ation, but not directly. It is first rendered to an image.
                                                              g to your LaTeX document.
imgmath_latex_preamble = r'''
\usepackage{xcolor}
\definecolor{offwhite}{rgb}{238,238,238}
\everymath{\color{offwhite}}
\everydisplay{\color{offwhite}}
'''


# -- Options for HTML output -------------------------------------------------------------------------------

# The theme to use for HTML and HTML Help pages.
# You can find available themes here: http://www.sphinx-doc.org/en/master/theming.html
# In this project, I wanted to use a non-default theme, and so I downloaded the `graphite` template from here:
#    https://github.com/Cartroo/sphinx-theme-graphite
#    Some adjustments I made to graphite:
#    - I did not use the pygment configuration, and so removed "pygments_style = graphite.GraphiteStyle" from theme.conf
#                                                       and deleted graphite.py
#    - In the static folder, I configured several classes both in graphite.css and in html4css1.css,
#         you can download the original and compare to find those changes.
html_theme = 'graphite'
# When using a non-built-in theme, define the path to your template code.
html_theme_path = ['.']

# Add any paths that contain custom static files (such as style sheets) here,
# relative to this directory. They are copied after the builtin static files,
# so a file named "default.css" will overwrite the builtin "default.css".
html_static_path = ['_static']

# Defining the static path allows me to add my own logo for the project:
#   (make sure the theme of your choice support the use of logo.
html_logo = '_static/sphinx_and_dalya.png'

# Custom sidebar templates, must be a dictionary that maps document names to template names.
# In This project I chose to include in the sidebar:
#   - Table of Contents: I chose globaltoc as it is less refined,
#     and configured the title by editing the globaltoc template (see explanation above, in the templates_path comment)
#   - Search box: appears below the TOc, and can be configured by editing css attributes.
html_sidebars = {
    '**': [
        'globaltoc.html',
        'searchbox.html'
    ]
}
```

```
autodoc_member_order = 'bysource'
```

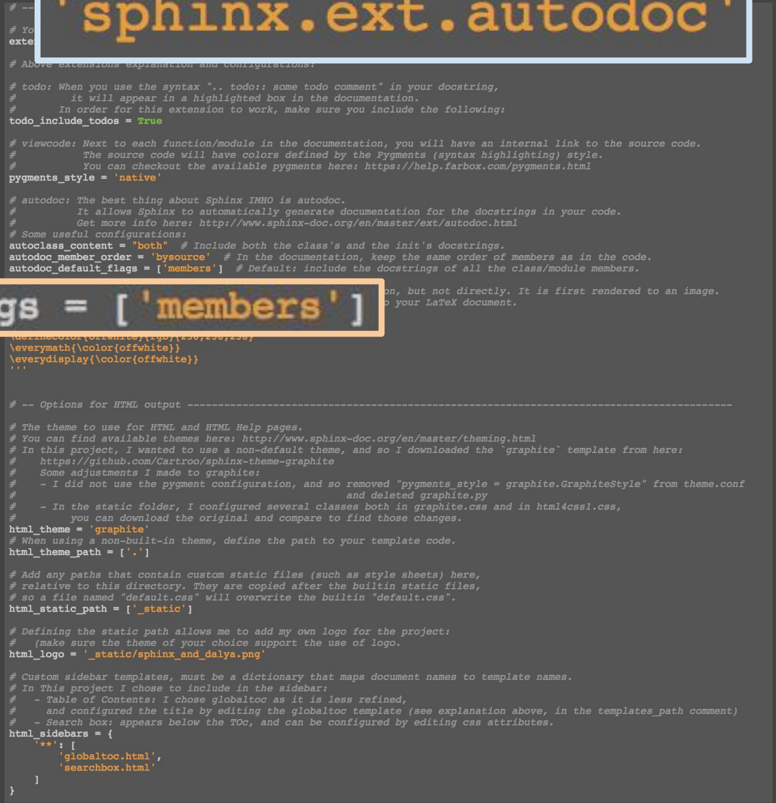
In the documentation, keep the same order of members as in the code

Part II - Main configuration file - conf.py

'sphinx.ext.autodoc'

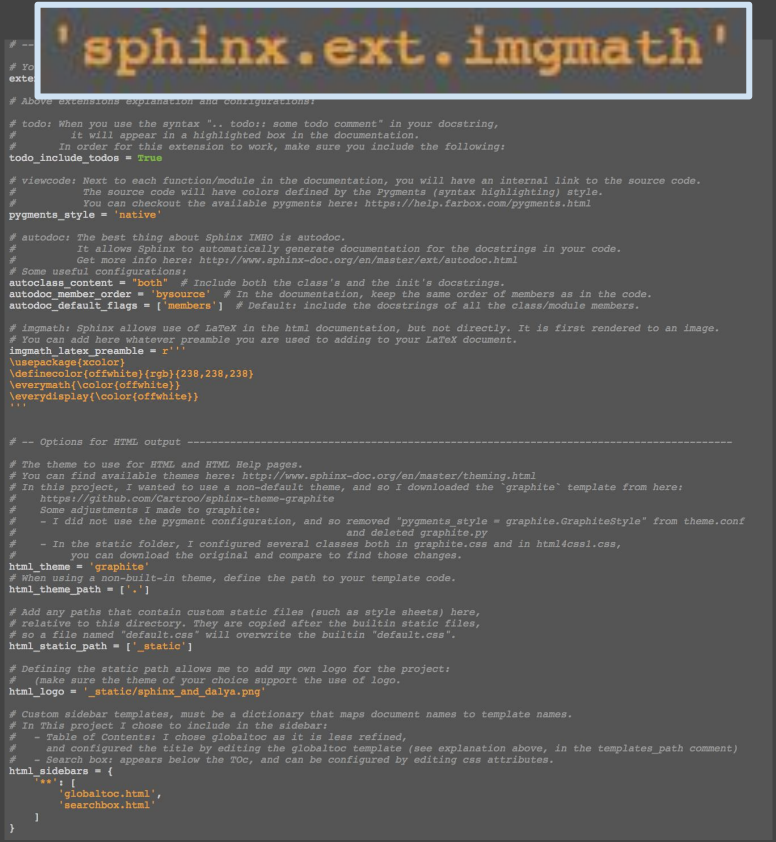
autodoc_default_flags = ['members']

Default: include the docstrings of all the class/module members



```
# --

# Yo
exte

# Above extensions explanation and configurations:

# todo: When you use the syntax ".. todo:: some todo comment" in your docstring,
#          it will appear in a highlighted box in the documentation.
#        In order for this extension to work, make sure you include the following:
todo_include_todos = True

# viewcode: Next to each function/module in the documentation, you will have an internal link to the source code.
#            The source code will have colors defined by the Pygments (syntax highlighting) style.
#            You can checkout the available pygments here: https://help.farbox.com/pygments.html
pygments_style = 'native'

# autodoc: The best thing about Sphinx IMHO is autodoc.
#           It allows Sphinx to automatically generate documentation for the docstrings in your code.
#           Get more info here: http://www.sphinx-doc.org/en/master/ext/autodoc.html
# Some useful configurations:
autoclass_content = "both"  # Include both the class's and the init's docstrings.
autodoc_member_order = 'bysource'  # In the documentation, keep the same order of members as in the code.
autodoc_default_flags = ['members']  # Default: include the docstrings of all the class/module members.

                                                                  on, but not directly. It is first rendered to an image.
                                                                  o your LaTeX document.

\everymath{\color{offwhite}}
\everydisplay{\color{offwhite}}
'''


# -- Options for HTML output ---------------------------------------------------------------------------------

# The theme to use for HTML and HTML Help pages.
# You can find available themes here: http://www.sphinx-doc.org/en/master/theming.html
# In this project, I wanted to use a non-default theme, and so I downloaded the `graphite` template from here:
#    https://github.com/Cartroo/sphinx-theme-graphite
#    Some adjustments I made to graphite:
#    - I did not use the pygment configuration, and so removed "pygments_style = graphite.GraphiteStyle" from theme.conf
#                                                            and deleted graphite.py
#    - In the static folder, I configured several classes both in graphite.css and in html4css1.css,
#         you can download the original and compare to find those changes.
html_theme = 'graphite'
# When using a non-built-in theme, define the path to your template code.
html_theme_path = ['.']

# Add any paths that contain custom static files (such as style sheets) here,
# relative to this directory. They are copied after the builtin static files,
# so a file named "default.css" will overwrite the builtin "default.css".
html_static_path = ['_static']

# Defining the static path allows me to add my own logo for the project:
#   (make sure the theme of your choice support the use of logo.
html_logo = '_static/sphinx_and_dalya.png'

# Custom sidebar templates, must be a dictionary that maps document names to template names.
# In This project I chose to include in the sidebar:
#   - Table of Contents: I chose globaltoc as it is less refined,
#     and configured the title by editing the globaltoc template (see explanation above, in the templates_path comment)
#   - Search box: appears below the TOc, and can be configured by editing css attributes.
html_sidebars = {
    '**': [
        'globaltoc.html',
        'searchbox.html'
    ]
}
```

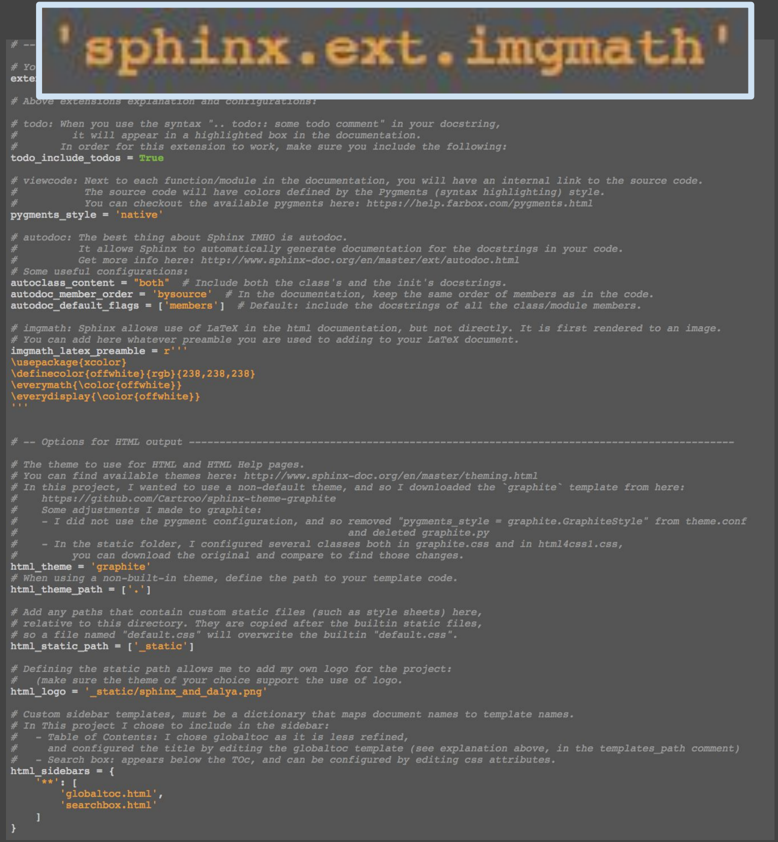
Part II - Main configuration file - conf.py



'sphinx.ext.imgmath'

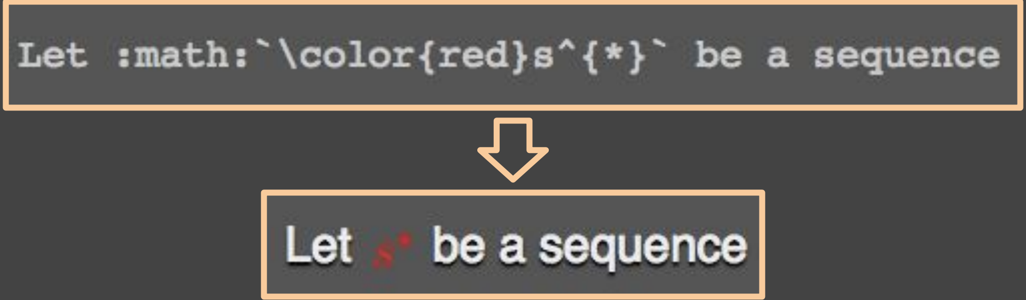
```
# --

# Yo
exte

# Above extensions explanation and configurations:

# todo: When you use the syntax ".. todo:: some todo comment" in your docstring,
#         it will appear in a highlighted box in the documentation.
#       In order for this extension to work, make sure you include the following:
todo_include_todos = True

# viewcode: Next to each function/module in the documentation, you will have an internal link to the source code.
#           The source code will have colors defined by the Pygments (syntax highlighting) style.
#           You can checkout the available pygments here: https://help.farbox.com/pygments.html
pygments_style = 'native'

# autodoc: The best thing about Sphinx IMHO is autodoc.
#          It allows Sphinx to automatically generate documentation for the docstrings in your code.
#          Get more info here: http://www.sphinx-doc.org/en/master/ext/autodoc.html
# Some useful configurations:
autoclass_content = "both"  # Include both the class's and the init's docstrings.
autodoc_member_order = 'bysource'  # In the documentation, keep the same order of members as in the code.
autodoc_default_flags = ['members']  # Default: include the docstrings of all the class/module members.

# imgmath: Sphinx allows use of LaTeX in the html documentation, but not directly. It is first rendered to an image.
# You can add here whatever preamble you are used to adding to your LaTeX document.
imgmath_latex_preamble = r'''
\usepackage{xcolor}
\definecolor{offwhite}{rgb}{238,238,238}
\everymath{\color{offwhite}}
\everydisplay{\color{offwhite}}
'''


# -- Options for HTML output -------------------------------------------------------------------------------------

# The theme to use for HTML and HTML Help pages.
# You can find available themes here: http://www.sphinx-doc.org/en/master/theming.html
# In this project, I wanted to use a non-default theme, and so I downloaded the `graphite` template from here:
#    https://github.com/Cartroo/sphinx-theme-graphite
#    Some adjustments I made to graphite:
#    - I did not use the pygment configuration, and so removed "pygments_style = graphite.GraphiteStyle" from theme.conf
#                                                      and deleted graphite.py
#    - In the static folder, I configured several classes both in graphite.css and in html4css1.css,
#         you can download the original and compare to find those changes.
html_theme = 'graphite'
# When using a non-built-in theme, define the path to your template code.
html_theme_path = ['.']

# Add any paths that contain custom static files (such as style sheets) here,
# relative to this directory. They are copied after the builtin static files,
# so a file named "default.css" will overwrite the builtin "default.css".
html_static_path = ['_static']

# Defining the static path allows me to add my own logo for the project:
#   (make sure the theme of your choice support the use of logo.
html_logo = '_static/sphinx_and_dalya.png'

# Custom sidebar templates, must be a dictionary that maps document names to template names.
# In This project I chose to include in the sidebar:
#   - Table of Contents: I chose globaltoc as it is less refined,
#     and configured the title by editing the globaltoc template (see explanation above, in the templates_path comment)
#   - Search box: appears below the TOc, and can be configured by editing css attributes.
html_sidebars = {
    '**': [
        'globaltoc.html',
        'searchbox.html'
    ]
}
```

Part II - Main configuration file - conf.py



'sphinx.ext.imgmath'

```
# --
# Yo
exte

# Above extensions explanation and configurations:

# todo: When you use the syntax ".. todo:: some todo comment" in your docstring,
#         it will appear in a highlighted box in the documentation.
#       In order for this extension to work, make sure you include the following:
todo_include_todos = True

# viewcode: Next to each function/module in the documentation, you will have an internal link to the source code.
#           The source code will have colors defined by the Pygments (syntax highlighting) style.
#           You can checkout the available pygments here: https://help.farbox.com/pygments.html
pygments_style = 'native'

# autodoc: The best thing about Sphinx IMHO is autodoc.
#          It allows Sphinx to automatically generate documentation for the docstrings in your code.
#          Get more info here: http://www.sphinx-doc.org/en/master/ext/autodoc.html
# Some useful configurations:
autoclass_content = "both"  # Include both the class's and the init's docstrings.
autodoc_member_order = 'bysource'  # In the documentation, keep the same order of members as in the code.
autodoc_default_flags = ['members']  # Default: include the docstrings of all the class/module members.

# imgmath: Sphinx allows use of LaTeX in the html documentation, but not directly. It is first rendered to an image.
# You can add here whatever preamble you are used to adding to your LaTeX document.
imgmath_latex_preamble = r'''
\usepackage{xcolor}
\definecolor{offwhite}{rgb}{238,238,238}
\everymath{\color{offwhite}}
\everydisplay{\color{offwhite}}
'''


# -- Options for HTML output -------------------------------------------------------------------------

# The theme to use for HTML and HTML Help pages.
# You can find available themes here: http://www.sphinx-doc.org/en/master/theming.html
# In this project, I wanted to use a non-default theme, and so I downloaded the `graphite` template from here:
#    https://github.com/Cartroo/sphinx-theme-graphite
#    Some adjustments I made to graphite:
#    - I did not use the pygment configuration, and so removed "pygments_style = graphite.GraphiteStyle" from theme.conf
#                                                 and deleted graphite.py
#    - In the static folder, I configured several classes both in graphite.css and in html4css1.css,
#         you can download the original and compare to find those changes.
html_theme = 'graphite'
# When using a non-built-in theme, define the path to your template code.
html_theme_path = ['.']

# Add any paths that contain custom static files (such as style sheets) here,
# relative to this directory. They are copied after the builtin static files,
# so a file named "default.css" will overwrite the builtin "default.css".
html_static_path = ['_static']

# Defining the static path allows me to add my own logo for the project:
#   (make sure the theme of your choice support the use of logo.
html_logo = '_static/sphinx_and_dalya.png'

# Custom sidebar templates, must be a dictionary that maps document names to template names.
# In This project I chose to include in the sidebar:
#   - Table of Contents: I chose globaltoc as it is less refined,
#     and configured the title by editing the globaltoc template (see explanation above, in the templates_path comment)
#   - Search box: appears below the TOc, and can be configured by editing css attributes.
html_sidebars = {
    '**': [
        'globaltoc.html',
        'searchbox.html'
    ]
}
```



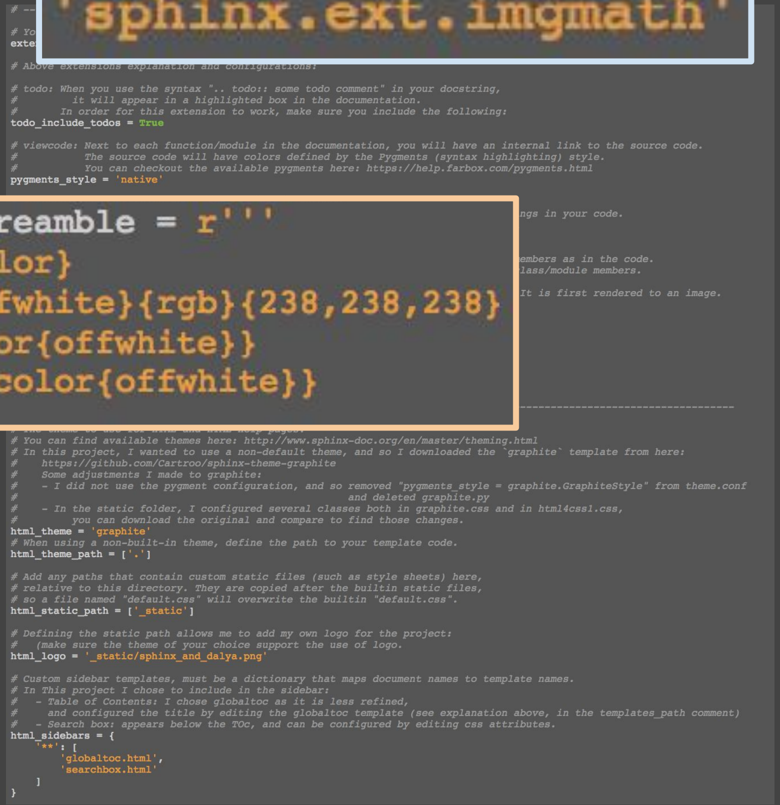
```
Let :math:`\color{red}s^{*}` be a sequence
```

Let s^{*} be a sequence

Part II - Main configuration file - conf.py

```
'sphinx.ext.imgmath'
```

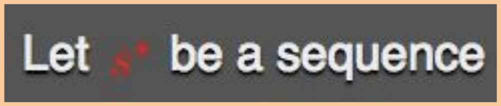
```
imgmath_latex_preamble = r'''
\usepackage{xcolor}
\definecolor{offwhite}{rgb}{238,238,238}
\everymath{\color{offwhite}}
\everydisplay{\color{offwhite}}
'''
```



```
# --
# Yo
exte
# Above extensions explanation and configurations:

# todo: When you use the syntax ".. todo:: some todo comment" in your docstring,
#         it will appear in a highlighted box in the documentation.
#       In order for this extension to work, make sure you include the following:
todo_include_todos = True

# viewcode: Next to each function/module in the documentation, you will have an internal link to the source code.
#           The source code will have colors defined by the Pygments (syntax highlighting) style.
#           You can checkout the available pygments here: https://help.farbox.com/pygments.html
pygments_style = 'native'

ngs in your code.

embers as in the code.
lass/module members.

It is first rendered to an image.

-----------------------------------
# You can find available themes here: http://www.sphinx-doc.org/en/master/theming.html
# In this project, I wanted to use a non-default theme, and so I downloaded the `graphite` template from here:
#    https://github.com/Cartroo/sphinx-theme-graphite
#    Some adjustments I made to graphite:
#    - I did not use the pygment configuration, and so removed "pygments_style = graphite.GraphiteStyle" from theme.conf
#                                                          and deleted graphite.py
#    - In the static folder, I configured several classes both in graphite.css and in html4css1.css,
#         you can download the original and compare to find those changes.
html_theme = 'graphite'
# When using a non-built-in theme, define the path to your template code.
html_theme_path = ['.']

# Add any paths that contain custom static files (such as style sheets) here,
# relative to this directory. They are copied after the builtin static files,
# so a file named "default.css" will overwrite the builtin "default.css".
html_static_path = ['_static']

# Defining the static path allows me to add my own logo for the project:
#   (make sure the theme of your choice support the use of logo.
html_logo = '_static/sphinx_and_dalya.png'

# Custom sidebar templates, must be a dictionary that maps document names to template names.
# In This project I chose to include in the sidebar:
#   - Table of Contents: I chose globaltoc as it is less refined,
#     and configured the title by editing the globaltoc template (see explanation above, in the templates_path comment)
#   - Search box: appears below the TOc, and can be configured by editing css attributes.
html_sidebars = {
    '**': [
        'globaltoc.html',
        'searchbox.html'
    ]
}
```

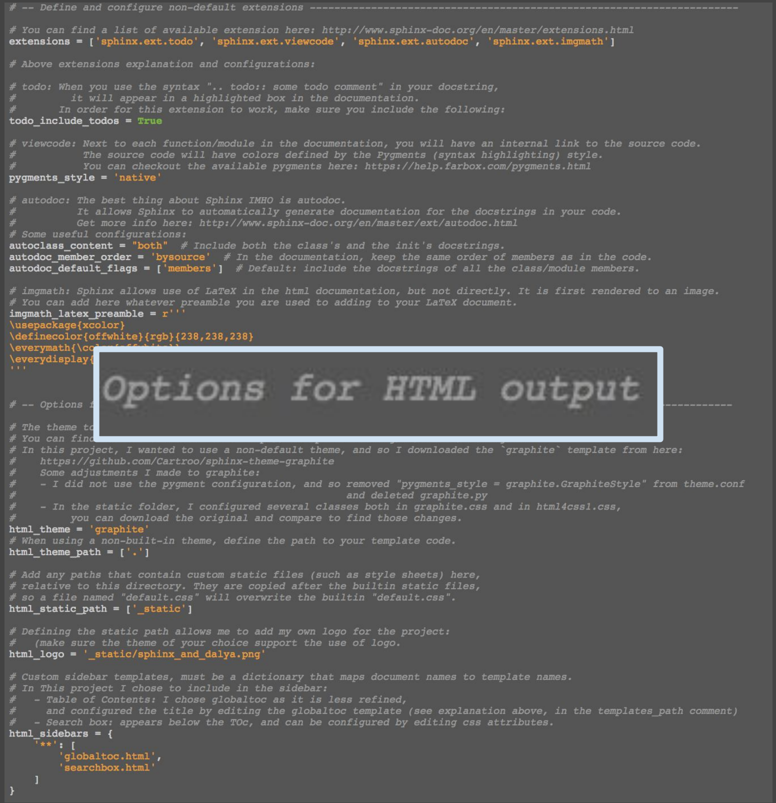
```
Let :math:`\color{red}s^{*}` be a sequence
```

⇩



Let s^{*} be a sequence

Part II - Main configuration file - conf.py



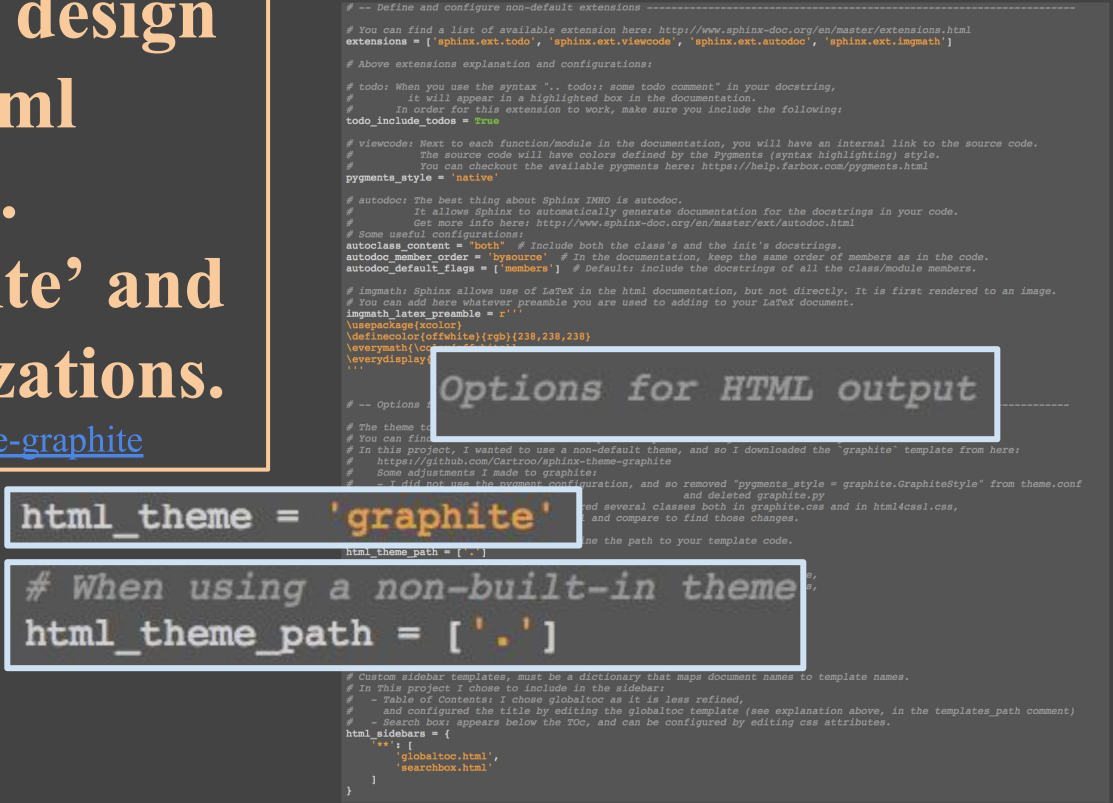
```
# -- Define and configure non-default extensions ---------------------------------------------------------------------

# You can find a list of available extension here: http://www.sphinx-doc.org/en/master/extensions.html
extensions = ['sphinx.ext.todo', 'sphinx.ext.viewcode', 'sphinx.ext.autodoc', 'sphinx.ext.imgmath']

# Above extensions explanation and configurations:

# todo: When you use the syntax ".. todo:: some todo comment" in your docstring,
#         it will appear in a highlighted box in the documentation.
#       In order for this extension to work, make sure you include the following:
todo_include_todos = True

# viewcode: Next to each function/module in the documentation, you will have an internal link to the source code.
#           The source code will have colors defined by the Pygments (syntax highlighting) style.
#           You can checkout the available pygments here: https://help.farbox.com/pygments.html
pygments_style = 'native'

# autodoc: The best thing about Sphinx IMHO is autodoc.
#          It allows Sphinx to automatically generate documentation for the docstrings in your code.
#          Get more info here: http://www.sphinx-doc.org/en/master/ext/autodoc.html
# Some useful configurations:
autoclass_content = "both"  # Include both the class's and the init's docstrings.
autodoc_member_order = 'bysource'  # In the documentation, keep the same order of members as in the code.
autodoc_default_flags = ['members']  # Default: include the docstrings of all the class/module members.

# imgmath: Sphinx allows use of LaTeX in the html documentation, but not directly. It is first rendered to an image.
# You can add here whatever preamble you are used to adding to your LaTeX document.
imgmath_latex_preamble = r'''
\usepackage{xcolor}
\definecolor{offwhite}{rgb}{238,238,238}
\everymath{\co
\everydisplay{
'''


# -- Options f                                                                                          ------------

# The theme to
# You can find
# In this project, I wanted to use a non-default theme, and so I downloaded the `graphite` template from here:
#    https://github.com/Cartroo/sphinx-theme-graphite
#    Some adjustments I made to graphite:
#    - I did not use the pygment configuration, and so removed "pygments_style = graphite.GraphiteStyle" from theme.conf
#                                                         and deleted graphite.py
#    - In the static folder, I configured several classes both in graphite.css and in html4css1.css,
#         you can download the original and compare to find those changes.
html_theme = 'graphite'
# When using a non-built-in theme, define the path to your template code.
html_theme_path = ['.']

# Add any paths that contain custom static files (such as style sheets) here,
# relative to this directory. They are copied after the builtin static files,
# so a file named "default.css" will overwrite the builtin "default.css".
html_static_path = ['_static']

# Defining the static path allows me to add my own logo for the project:
#   (make sure the theme of your choice support the use of logo.
html_logo = '_static/sphinx_and_dalya.png'

# Custom sidebar templates, must be a dictionary that maps document names to template names.
# In This project I chose to include in the sidebar:
#   - Table of Contents: I chose globaltoc as it is less refined,
#     and configured the title by editing the globaltoc template (see explanation above, in the templates_path comment)
#   - Search box: appears below the TOc, and can be configured by editing css attributes.
html_sidebars = {
    '**': [
        'globaltoc.html',
        'searchbox.html'
    ]
}
```

Options for HTML output

Part II - Main configuration file - conf.py

The themes holds the design for the output html documentation.
I downloaded ‘graphite’ and made some personalizations.

https://github.com/Cartroo/sphinx-theme-graphite



```
# -- Define and configure non-default extensions -------------------------------------------------------

# You can find a list of available extension here: http://www.sphinx-doc.org/en/master/extensions.html
extensions = ['sphinx.ext.todo', 'sphinx.ext.viewcode', 'sphinx.ext.autodoc', 'sphinx.ext.imgmath']

# Above extensions explanation and configurations:

# todo: When you use the syntax ".. todo:: some todo comment" in your docstring,
#          it will appear in a highlighted box in the documentation.
#        In order for this extension to work, make sure you include the following:
todo_include_todos = True

# viewcode: Next to each function/module in the documentation, you will have an internal link to the source code.
#            The source code will have colors defined by the Pygments (syntax highlighting) style.
#            You can checkout the available pygments here: https://help.farbox.com/pygments.html
pygments_style = 'native'

# autodoc: The best thing about Sphinx IMHO is autodoc.
#           It allows Sphinx to automatically generate documentation for the docstrings in your code.
#           Get more info here: http://www.sphinx-doc.org/en/master/ext/autodoc.html
# Some useful configurations:
autoclass_content = "both"  # Include both the class's and the init's docstrings.
autodoc_member_order = 'bysource'  # In the documentation, keep the same order of members as in the code.
autodoc_default_flags = ['members']  # Default: include the docstrings of all the class/module members.

# imgmath: Sphinx allows use of LaTeX in the html documentation, but not directly. It is first rendered to an image.
# You can add here whatever preamble you are used to adding to your LaTeX document.
imgmath_latex_preamble = r'''
\usepackage{xcolor}
\definecolor{offwhite}{rgb}{238,238,238}
\everymath{\co
\everydisplay{
'''


# -- Options f                                                                                   ------------

# The theme to
# You can find
# In this project, I wanted to use a non-default theme, and so I downloaded the `graphite` template from here:
#    https://github.com/Cartroo/sphinx-theme-graphite
#    Some adjustments I made to graphite:
#    - I did not use the pygment configuration, and so removed "pygments_style = graphite.GraphiteStyle" from theme.conf
                                                      and deleted graphite.py
                                           red several classes both in graphite.css and in html4css1.css,
                                           l and compare to find those changes.

                                           ine the path to your template code.
html_theme_path = ['.']

                                                                                          e,
                                                                                          s,





# Custom sidebar templates, must be a dictionary that maps document names to template names.
# In This project I chose to include in the sidebar:
#   - Table of Contents: I chose globaltoc as it is less refined,
#     and configured the title by editing the globaltoc template (see explanation above, in the templates_path comment)
#   - Search box: appears below the TOc, and can be configured by editing css attributes.
html_sidebars = {
    '**': [
        'globaltoc.html',
        'searchbox.html'
    ]
}
```

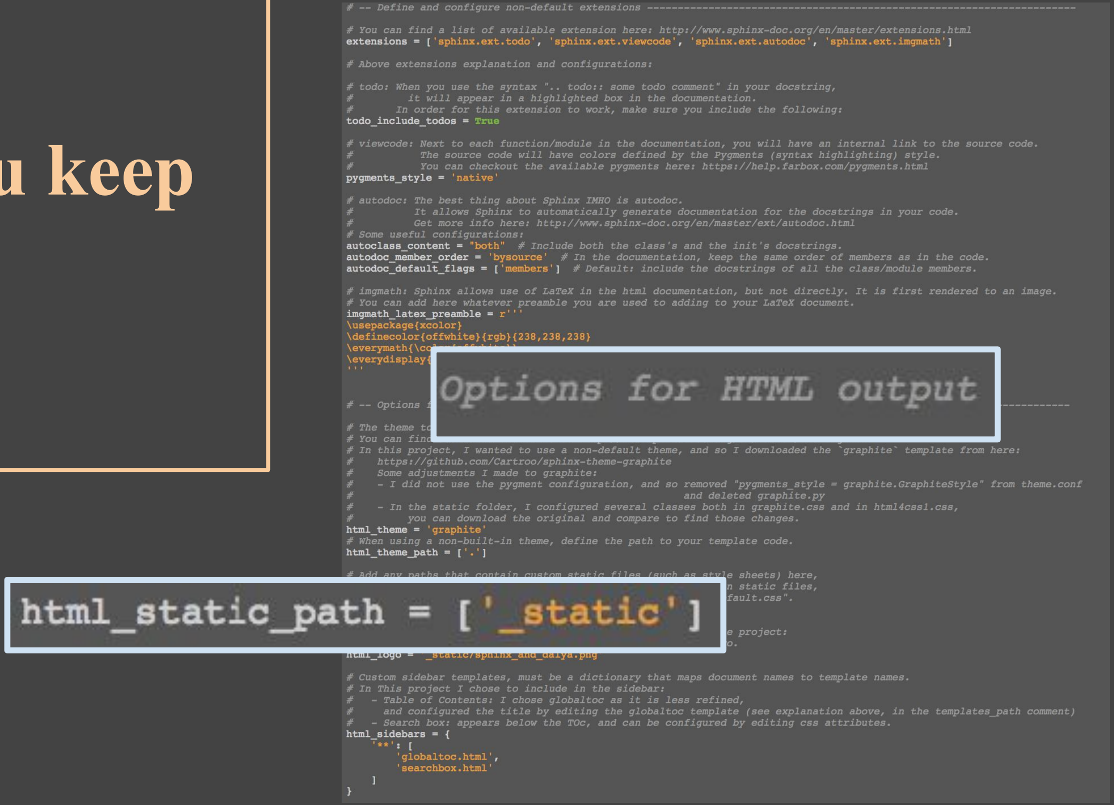
```
Options for HTML output
```

```
html_theme = 'graphite'
```

```
# When using a non-built-in theme
html_theme_path = ['.']
```

Part II - Main configuration file - conf.py

The folder where you keep images etc.



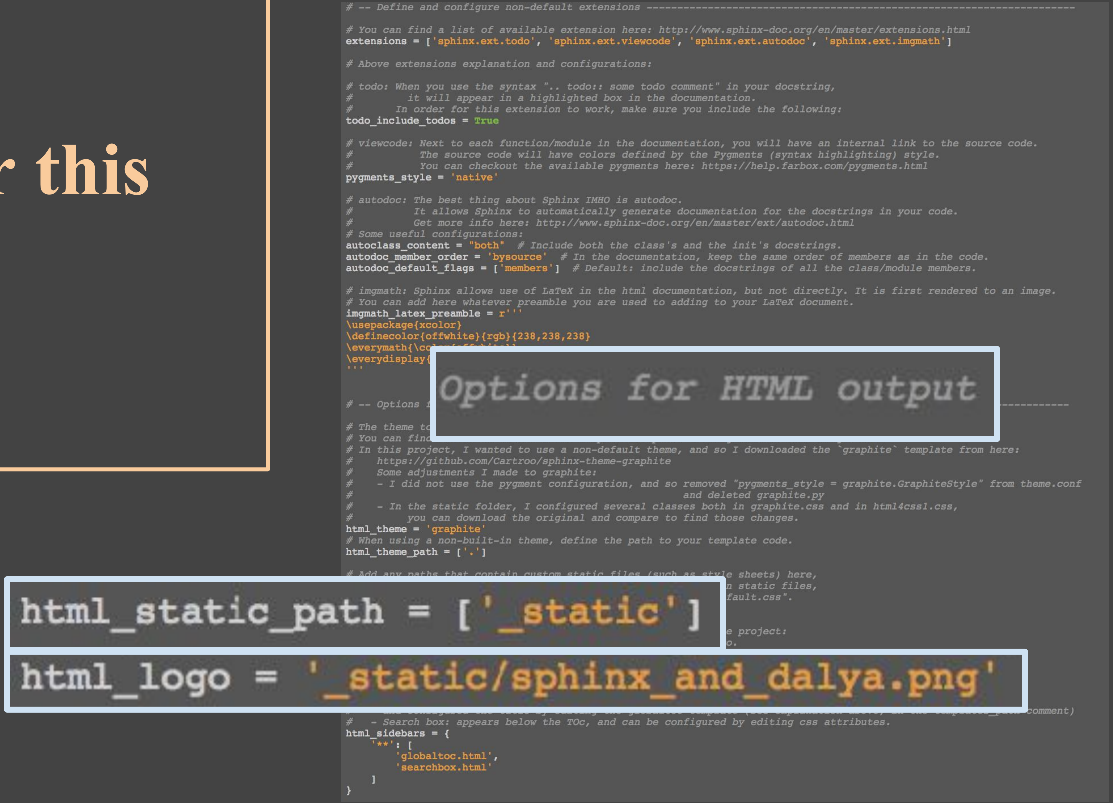
```
# -- Define and configure non-default extensions ----------------------------------------------------------------

# You can find a list of available extension here: http://www.sphinx-doc.org/en/master/extensions.html
extensions = ['sphinx.ext.todo', 'sphinx.ext.viewcode', 'sphinx.ext.autodoc', 'sphinx.ext.imgmath']

# Above extensions explanation and configurations:

# todo: When you use the syntax ".. todo:: some todo comment" in your docstring,
#         it will appear in a highlighted box in the documentation.
#       In order for this extension to work, make sure you include the following:
todo_include_todos = True

# viewcode: Next to each function/module in the documentation, you will have an internal link to the source code.
#            The source code will have colors defined by the Pygments (syntax highlighting) style.
#            You can checkout the available pygments here: https://help.farbox.com/pygments.html
pygments_style = 'native'

# autodoc: The best thing about Sphinx IMHO is autodoc.
#           It allows Sphinx to automatically generate documentation for the docstrings in your code.
#           Get more info here: http://www.sphinx-doc.org/en/master/ext/autodoc.html
# Some useful configurations:
autoclass_content = "both"  # Include both the class's and the init's docstrings.
autodoc_member_order = 'bysource'  # In the documentation, keep the same order of members as in the code.
autodoc_default_flags = ['members']  # Default: include the docstrings of all the class/module members.

# imgmath: Sphinx allows use of LaTeX in the html documentation, but not directly. It is first rendered to an image.
# You can add here whatever preamble you are used to adding to your LaTeX document.
imgmath_latex_preamble = r'''
\usepackage{xcolor}
\definecolor{offwhite}{rgb}{238,238,238}
\everymath{\co
\everydisplay{
'''


# -- Options f                                                                              -------------

# The theme to
# You can find
# In this project, I wanted to use a non-default theme, and so I downloaded the `graphite` template from here:
#    https://github.com/Cartroo/sphinx-theme-graphite
#    Some adjustments I made to graphite:
#    - I did not use the pygment configuration, and so removed "pygments_style = graphite.GraphiteStyle" from theme.conf
#                                                               and deleted graphite.py
#    - In the static folder, I configured several classes both in graphite.css and in html4css1.css,
#          you can download the original and compare to find those changes.
html_theme = 'graphite'
# When using a non-built-in theme, define the path to your template code.
html_theme_path = ['.']

# Add any paths that contain custom static files (such as style sheets) here,
                                                                   n static files,
                                                                   fault.css".



                                                                   e project:
                                                                   o.
html_logo = '_static/sphinx_and_dalya.png'

# Custom sidebar templates, must be a dictionary that maps document names to template names.
# In This project I chose to include in the sidebar:
#   - Table of Contents: I chose globaltoc as it is less refined,
#     and configured the title by editing the globaltoc template (see explanation above, in the templates_path comment)
#   - Search box: appears below the TOc, and can be configured by editing css attributes.
html_sidebars = {
    '**': [
        'globaltoc.html',
        'searchbox.html'
    ]
}
```

```
Options for HTML output
```

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

Part II - Main configuration file - conf.py

Such as the logo for this project



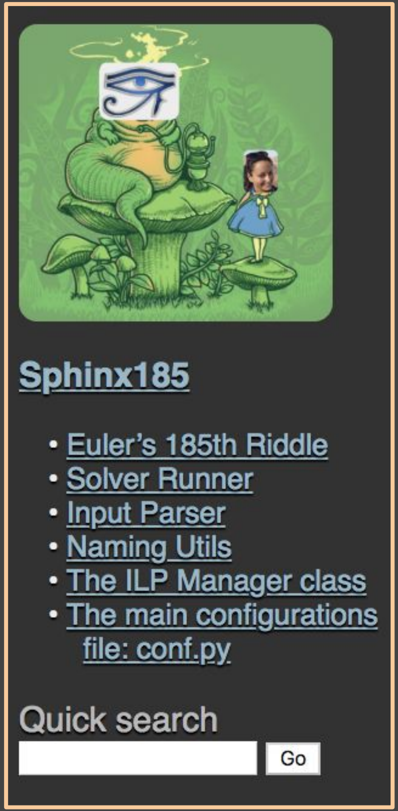
```
# -- Define and configure non-default extensions ---------------------------------------------------------

# You can find a list of available extension here: http://www.sphinx-doc.org/en/master/extensions.html
extensions = ['sphinx.ext.todo', 'sphinx.ext.viewcode', 'sphinx.ext.autodoc', 'sphinx.ext.imgmath']

# Above extensions explanation and configurations:

# todo: When you use the syntax ".. todo:: some todo comment" in your docstring,
#         it will appear in a highlighted box in the documentation.
#       In order for this extension to work, make sure you include the following:
todo_include_todos = True

# viewcode: Next to each function/module in the documentation, you will have an internal link to the source code.
#           The source code will have colors defined by the Pygments (syntax highlighting) style.
#           You can checkout the available pygments here: https://help.farbox.com/pygments.html
pygments_style = 'native'

# autodoc: The best thing about Sphinx IMHO is autodoc.
#          It allows Sphinx to automatically generate documentation for the docstrings in your code.
#          Get more info here: http://www.sphinx-doc.org/en/master/ext/autodoc.html
# Some useful configurations:
autoclass_content = "both"  # Include both the class's and the init's docstrings.
autodoc_member_order = 'bysource'  # In the documentation, keep the same order of members as in the code.
autodoc_default_flags = ['members']  # Default: include the docstrings of all the class/module members.

# imgmath: Sphinx allows use of LaTeX in the html documentation, but not directly. It is first rendered to an image.
# You can add here whatever preamble you are used to adding to your LaTeX document.
imgmath_latex_preamble = r'''
\usepackage{xcolor}
\definecolor{offwhite}{rgb}{238,238,238}
\everymath{\co
\everydisplay{
'''


# -- Options f                                                                              ------------

# The theme to
# You can find
# In this project, I wanted to use a non-default theme, and so I downloaded the `graphite` template from here:
#    https://github.com/Cartroo/sphinx-theme-graphite
#    Some adjustments I made to graphite:
#    - I did not use the pygment configuration, and so removed "pygments_style = graphite.GraphiteStyle" from theme.conf
#                                                            and deleted graphite.py
#    - In the static folder, I configured several classes both in graphite.css and in html4css1.css,
#         you can download the original and compare to find those changes.
html_theme = 'graphite'
# When using a non-built-in theme, define the path to your template code.
html_theme_path = ['.']

# Add any paths that contain custom static files (such as style sheets) here,
                                                                     n static files,
                                                                     fault.css".

                                                                     e project:
                                                                     o.


                                                                                                   comment)
#   - Search box: appears below the TOc, and can be configured by editing css attributes.
html_sidebars = {
    '**': [
        'globaltoc.html',
        'searchbox.html'
    ]
}
```

Options for HTML output

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

```
html_logo = '_static/sphinx_and_dalya.png'
```

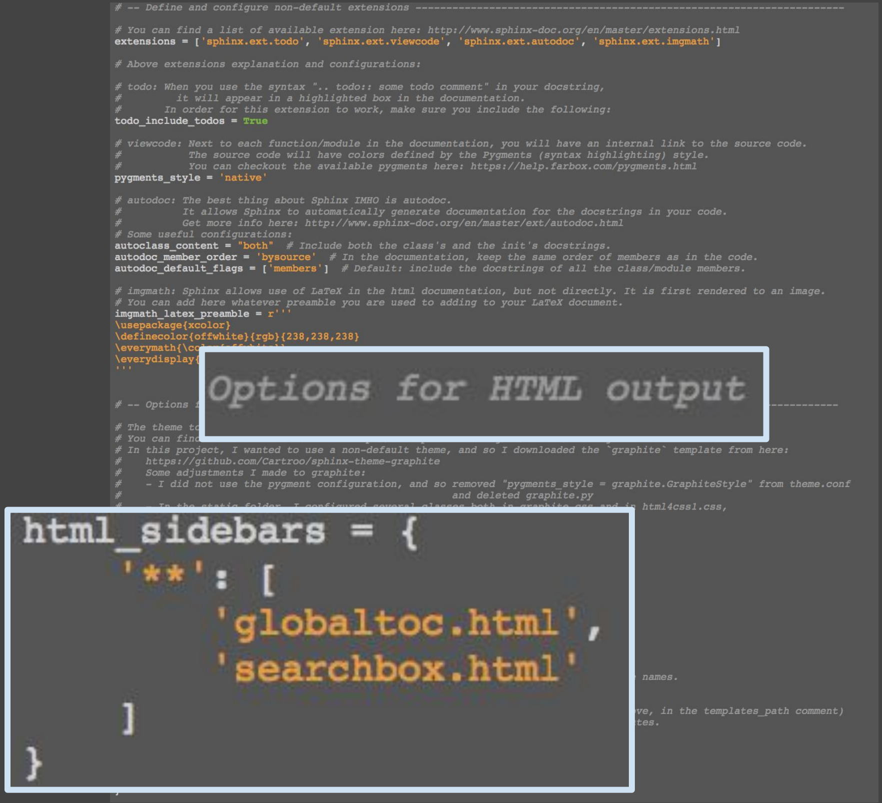
Part II - Main configuration file - conf.py



Sphinx185

- Euler’s 185th Riddle
- Solver Runner
- Input Parser
- Naming Utils
- The ILP Manager class
- The main configurations file: conf.py

Quick search



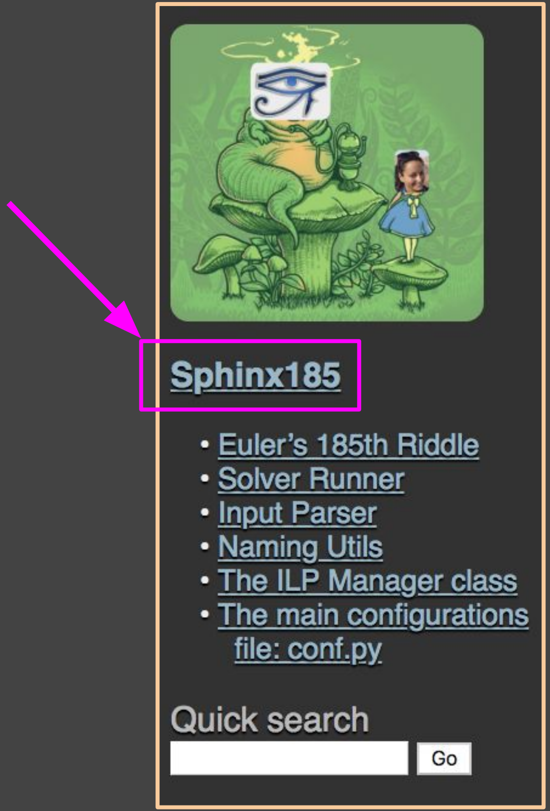
```
# -- Define and configure non-default extensions ---------------------------------------------------------------------------

# You can find a list of available extension here: http://www.sphinx-doc.org/en/master/extensions.html
extensions = ['sphinx.ext.todo', 'sphinx.ext.viewcode', 'sphinx.ext.autodoc', 'sphinx.ext.imgmath']

# Above extensions explanation and configurations:

# todo: When you use the syntax ".. todo:: some todo comment" in your docstring,
#          it will appear in a highlighted box in the documentation.
#        In order for this extension to work, make sure you include the following:
todo_include_todos = True

# viewcode: Next to each function/module in the documentation, you will have an internal link to the source code.
#            The source code will have colors defined by the Pygments (syntax highlighting) style.
#            You can checkout the available pygments here: https://help.farbox.com/pygments.html
pygments_style = 'native'

# autodoc: The best thing about Sphinx IMHO is autodoc.
#           It allows Sphinx to automatically generate documentation for the docstrings in your code.
#           Get more info here: http://www.sphinx-doc.org/en/master/ext/autodoc.html
# Some useful configurations:
autoclass_content = "both"  # Include both the class's and the init's docstrings.
autodoc_member_order = 'bysource'  # In the documentation, keep the same order of members as in the code.
autodoc_default_flags = ['members']  # Default: include the docstrings of all the class/module members.

# imgmath: Sphinx allows use of LaTeX in the html documentation, but not directly. It is first rendered to an image.
# You can add here whatever preamble you are used to adding to your LaTeX document.
imgmath_latex_preamble = r'''
\usepackage{xcolor}
\definecolor{offwhite}{rgb}{238,238,238}
\everymath{\co
\everydisplay{
'''


# -- Options f                                                                                     ------------

# The theme to
# You can find
# In this project, I wanted to use a non-default theme, and so I downloaded the `graphite` template from here:
#    https://github.com/Cartroo/sphinx-theme-graphite
#    Some adjustments I made to graphite:
#    - I did not use the pygment configuration, and so removed "pygments_style = graphite.GraphiteStyle" from theme.conf
#                                                      and deleted graphite.py
#    - In the static folder, I configured several classes both in graphite.css and in html4css1.css,
```

```
Options for HTML output
```

```
html_sidebars = {
    '**': [
        'globaltoc.html',
        'searchbox.html'
    ]
}
```

```
names.

ve, in the templates_path comment)
tes.
```

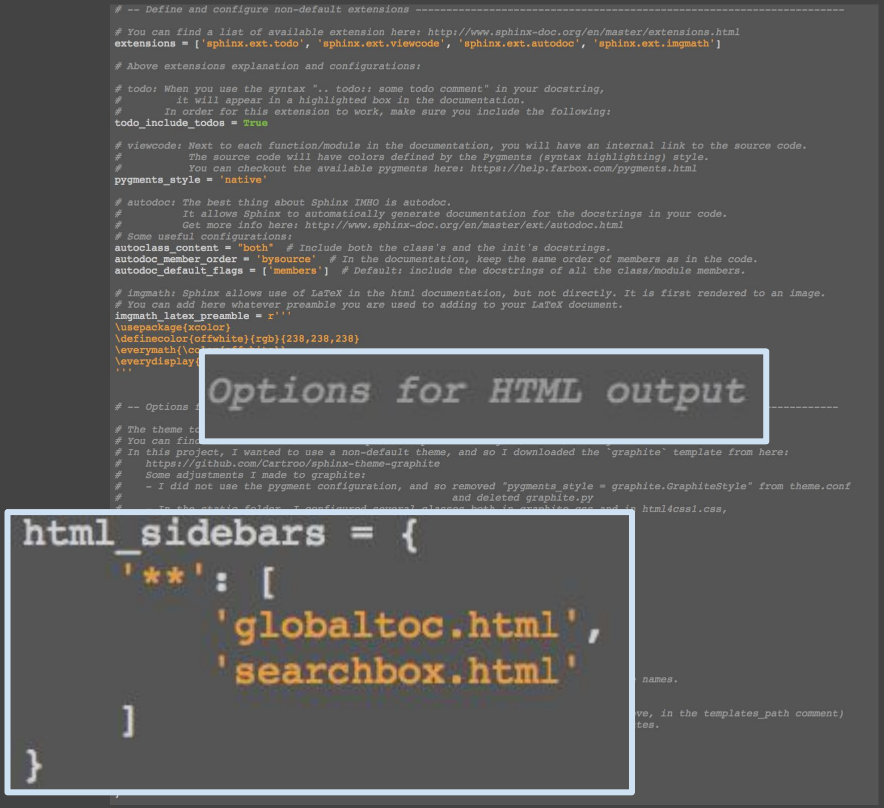
Part II - Main configuration file - conf.py



Sphinx185

- Euler's 185th Riddle
- Solver Runner
- Input Parser
- Naming Utils
- The ILP Manager class
- The main configurations file: conf.py

Quick search



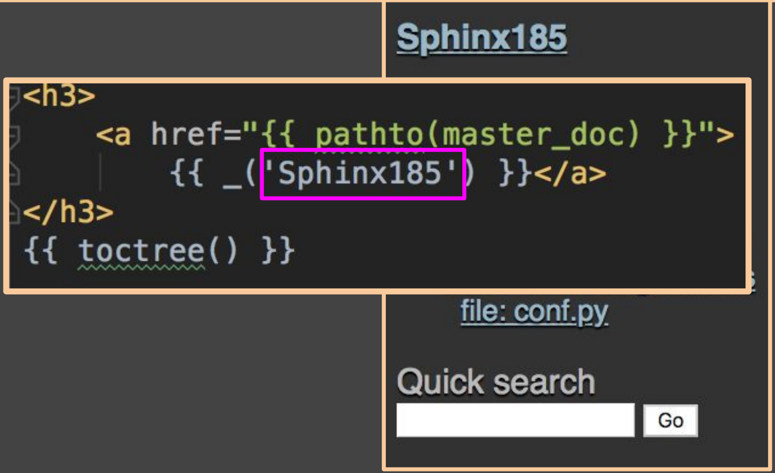
```
# -- Define and configure non-default extensions ------------------------------------------------------------------

# You can find a list of available extension here: http://www.sphinx-doc.org/en/master/extensions.html
extensions = ['sphinx.ext.todo', 'sphinx.ext.viewcode', 'sphinx.ext.autodoc', 'sphinx.ext.imgmath']

# Above extensions explanation and configurations:

# todo: When you use the syntax ".. todo:: some todo comment" in your docstring,
#          it will appear in a highlighted box in the documentation.
#        In order for this extension to work, make sure you include the following:
todo_include_todos = True

# viewcode: Next to each function/module in the documentation, you will have an internal link to the source code.
#            The source code will have colors defined by the Pygments (syntax highlighting) style.
#            You can checkout the available pygments here: https://help.farbox.com/pygments.html
pygments_style = 'native'

# autodoc: The best thing about Sphinx IMHO is autodoc.
#           It allows Sphinx to automatically generate documentation for the docstrings in your code.
#           Get more info here: http://www.sphinx-doc.org/en/master/ext/autodoc.html
# Some useful configurations:
autoclass_content = "both"  # Include both the class's and the init's docstrings.
autodoc_member_order = 'bysource'  # In the documentation, keep the same order of members as in the code.
autodoc_default_flags = ['members']  # Default: include the docstrings of all the class/module members.

# imgmath: Sphinx allows use of LaTeX in the html documentation, but not directly. It is first rendered to an image.
# You can add here whatever preamble you are used to adding to your LaTeX document.
imgmath_latex_preamble = r'''
\usepackage{xcolor}
\definecolor{offwhite}{rgb}{238,238,238}
\everymath{\co
\everydisplay{
'''


# -- Options f                                                                                        ------------

# The theme to
# You can find
# In this project, I wanted to use a non-default theme, and so I downloaded the `graphite` template from here:
#    https://github.com/Cartroo/sphinx-theme-graphite
#    Some adjustments I made to graphite:
#    - I did not use the pygment configuration, and so removed "pygments_style = graphite.GraphiteStyle" from theme.conf
#                                                               and deleted graphite.py
#      - In the static folder, I configured several classes both in graphite.css and in html4css1.css,
                                                                                    names.

                                                                                    ve, in the templates_path comment)
                                                                                    tes.
```

```
Options for HTML output
```

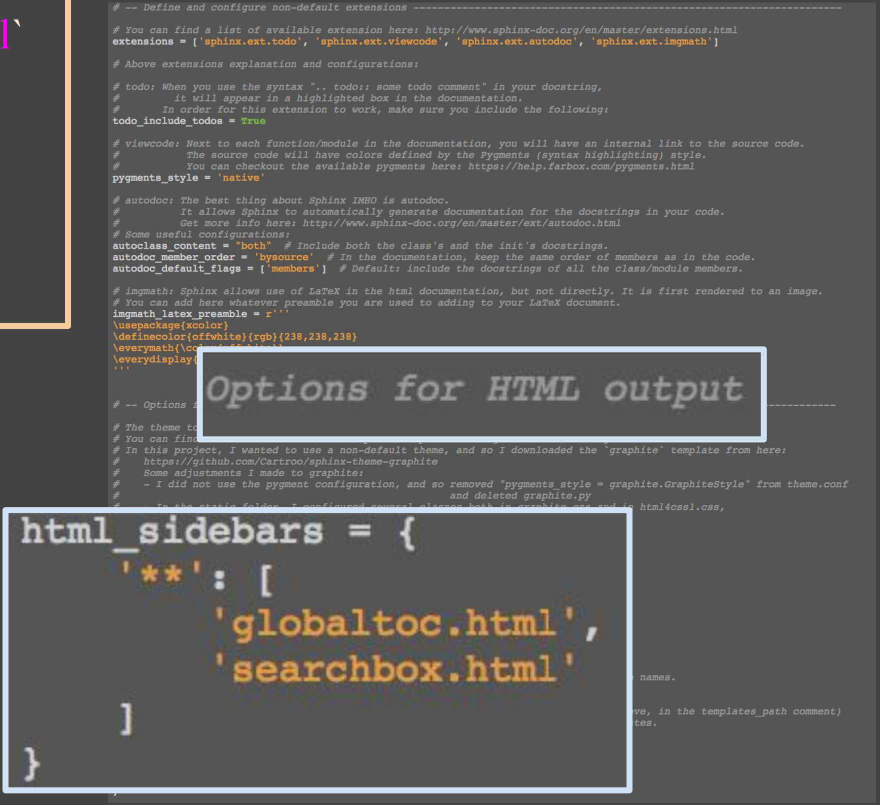
```
html_sidebars = {
    '**': [
        'globaltoc.html',
        'searchbox.html'
    ]
}
```

Part II - Main configuration file - conf.py

copy `<Sphinx install dir>/themes/basic/globaltoc.html`
in to the `_templates` folder in my project,
and replaced 'Table of Content' with 'Sphinx185':



```
<h3>
    <a href="{{ pathto(master_doc) }}">
        {{ _('Sphinx185') }}</a>
</h3>
{{ toctree() }}
```



```
# -- Define and configure non-default extensions -------------------------------------------------

# You can find a list of available extension here: http://www.sphinx-doc.org/en/master/extensions.html
extensions = ['sphinx.ext.todo', 'sphinx.ext.viewcode', 'sphinx.ext.autodoc', 'sphinx.ext.imgmath']

# Above extensions explanation and configurations:

# todo: When you use the syntax ".. todo:: some todo comment" in your docstring,
#         it will appear in a highlighted box in the documentation.
#       In order for this extension to work, make sure you include the following:
todo_include_todos = True

# viewcode: Next to each function/module in the documentation, you will have an internal link to the source code.
#           The source code will have colors defined by the Pygments (syntax highlighting) style.
#           You can checkout the available pygments here: https://help.farbox.com/pygments.html
pygments_style = 'native'

# autodoc: The best thing about Sphinx IMHO is autodoc.
#          It allows Sphinx to automatically generate documentation for the docstrings in your code.
#          Get more info here: http://www.sphinx-doc.org/en/master/ext/autodoc.html
# Some useful configurations:
autoclass_content = "both"  # Include both the class's and the init's docstrings.
autodoc_member_order = 'bysource'  # In the documentation, keep the same order of members as in the code.
autodoc_default_flags = ['members']  # Default: include the docstrings of all the class/module members.

# imgmath: Sphinx allows use of LaTeX in the html documentation, but not directly. It is first rendered to an image.
# You can add here whatever preamble you are used to adding to your LaTeX document.
imgmath_latex_preamble = r'''
\usepackage{xcolor}
\definecolor{offwhite}{rgb}{238,238,238}
\everymath{\co
\everydisplay{
'''


# -- Options f                                                                          ------------

# The theme to
# You can find
# In this project, I wanted to use a non-default theme, and so I downloaded the `graphite` template from here:
#    https://github.com/Cartroo/sphinx-theme-graphite
#    Some adjustments I made to graphite:
#    - I did not use the pygment configuration, and so removed "pygments_style = graphite.GraphiteStyle" from theme.conf
#                                                          and deleted graphite.py
#                                                                     html4css1.css,

                                                                      names.

                                                                      ve, in the templates_path comment)
                                                                      tes.
```

Options for HTML output

```
html_sidebars = {
    '**': [
        'globaltoc.html',
        'searchbox.html'
    ]
}
```