

Tutorial for creating an open-source software package on GitHub

Yuxi(Lucy) Lu

Create a new repository and initialize with MIT License

Title and description of the package

Repository name *

Owner lyx12311 / OpenSoftwareTut ✓

Great repository names are short and memorable. Need inspiration? How about [redesigned-octo-sniffle](#)?

Description (optional)

This is a tutorial repository for publishing open software

Public
Anyone can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: None | Add a license: None ⓘ

Licenses

Choose a license

✓ None

Apache License 2.0

GNU General Public License v3.0

MIT License

BSD 2-Clause "Simplified" License

BSD 3-Clause "New" or "Revised" License

Creative Commons Zero v1.0 Universal

Eclipse Public License 2.0

GNU Affero General Public License v3.0

© 2020 GitHub, Inc. Terms Privacy Security Stat Contact GitHub Pricing API Training Blog About

Add MIT License

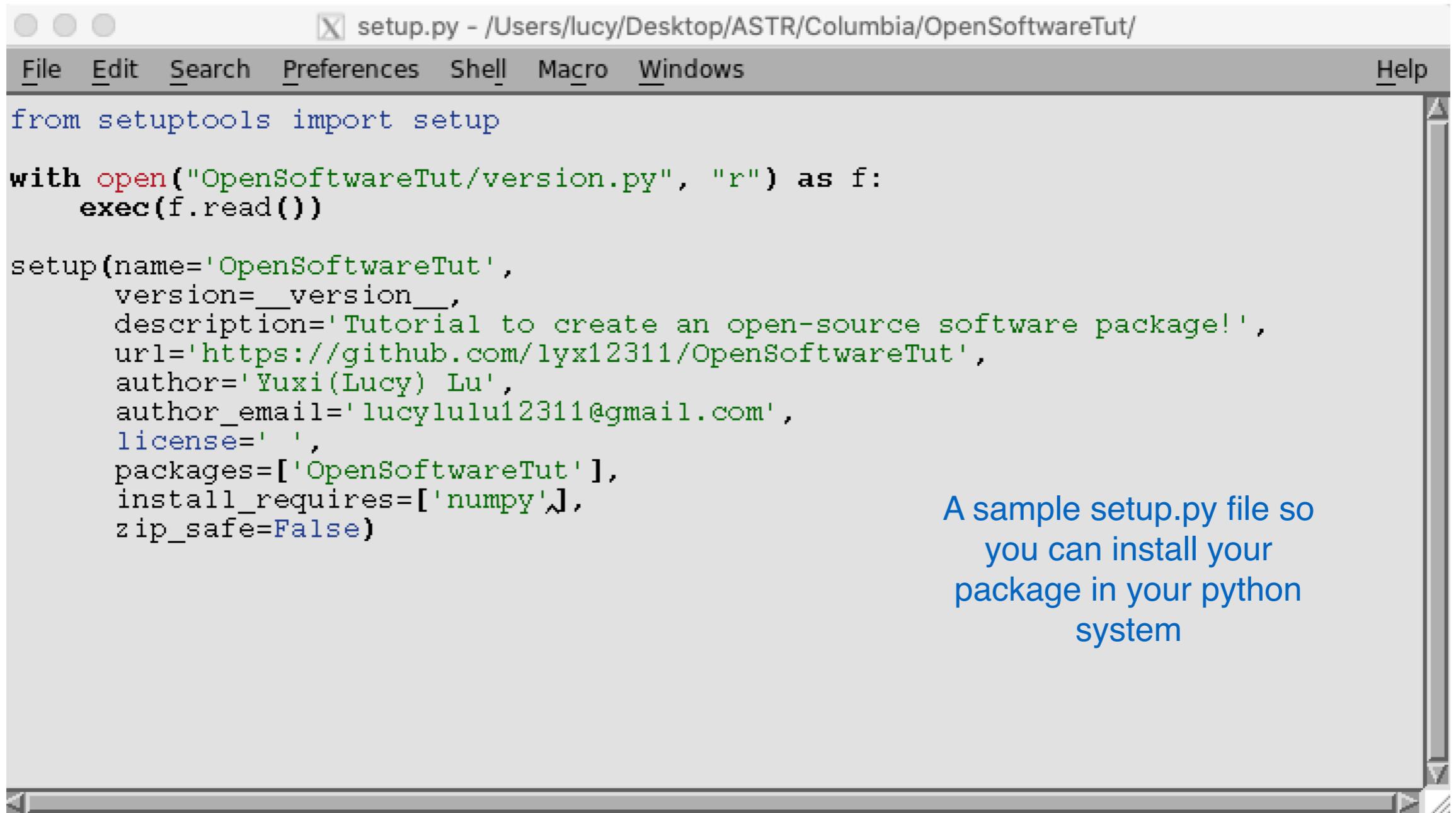
If you want to connect this project with your laptop and work locally

The screenshot shows a GitHub repository page for 'lyx12311/OpenSoftwareTut'. The repository has 2 commits, 1 branch, 0 packages, 0 releases, and 1 contributor. The README.md file contains the text: 'This is a tutorial repository for publishing open software'. The 'Clone or download' button is highlighted with a red box, and the 'Clone with SSH' field contains the URL 'git@github.com:lyx12311/OpenSoftwareTut.git'. Below the clone options, a terminal window shows the command being run: '(base) lucys-MBP-2:Columbia lucy\$ git clone git@github.com:lyx12311/OpenSoftwareTut.git'. The terminal output includes: 'Cloning into 'OpenSoftwareTut'...', 'Warning: untrusted X11 forwarding setup failed: xauth key data not generated', 'remote: Enumerating objects: 6, done.', 'remote: Counting objects: 100% (6/6), done.', 'remote: Compressing objects: 100% (5/5), done.', 'remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0', and 'Receiving objects: 100% (6/6), done.'

Create your code folder

- Under your main folder:

- mkdir OpenSoftwareTut
- touch setup.py



The screenshot shows a code editor window with the title bar "setup.py - /Users/lucy/Desktop/ASTR/Columbia/OpenSoftwareTut/". The menu bar includes File, Edit, Search, Preferences, Shell, Macro, Windows, and Help. The code in the editor is as follows:

```
from setuptools import setup

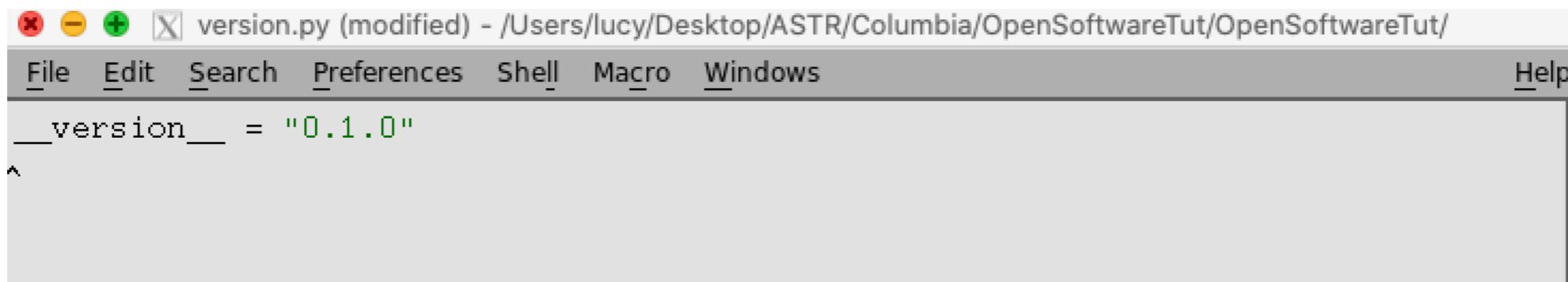
with open("OpenSoftwareTut/version.py", "r") as f:
    exec(f.read())

setup(name='OpenSoftwareTut',
      version=__version__,
      description='Tutorial to create an open-source software package!',
      url='https://github.com/lyx12311/OpenSoftwareTut',
      author='Yuxi(Lucy) Lu',
      author_email='lucylulu12311@gmail.com',
      license='',
      packages=['OpenSoftwareTut'],
      install_requires=['numpy'],
      zip_safe=False)
```

A sample setup.py file so you can install your package in your python system

Create your code folder

- Under your main folder:
 - mkdir OpenSoftwareTut
 - touch setup.py
- Under your code folder (OpenSoftwareTut):
 - Create version.py

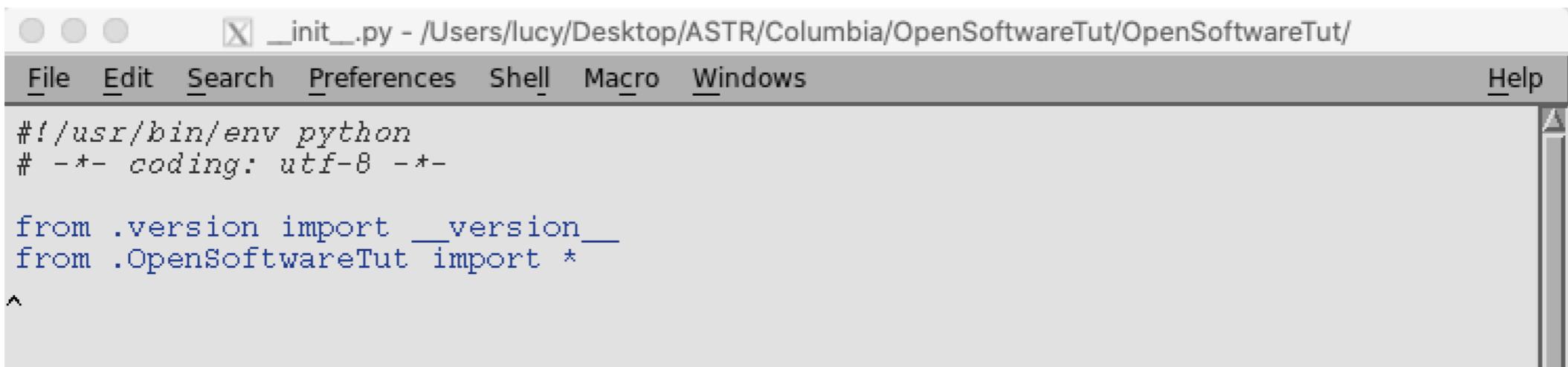


A screenshot of a text editor window titled "version.py (modified) - /Users/lucy/Desktop/ASTR/Columbia/OpenSoftwareTut/OpenSoftwareTut/". The window has a standard OS X-style title bar with close, minimize, and zoom buttons. The menu bar includes "File", "Edit", "Search", "Preferences", "Shell", "Macro", "Windows", and "Help". The main text area contains the following Python code:

```
__version__ = "0.1.0"
```

Create your code folder

- Under your main folder:
 - mkdir OpenSoftwareTut
 - touch setup.py
- Under your code folder (OpenSoftwareTut):
 - Create version.py
 - Create __init__.py



The screenshot shows a text editor window with the title bar reading "X __init__.py - /Users/lucy/Desktop/ASTR/Columbia/OpenSoftwareTut/OpenSoftwareTut/". The menu bar includes File, Edit, Search, Preferences, Shell, Macro, Windows, and Help. The main content area displays the following Python code:

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

from .version import __version__
from .OpenSoftwareTut import *

^
```

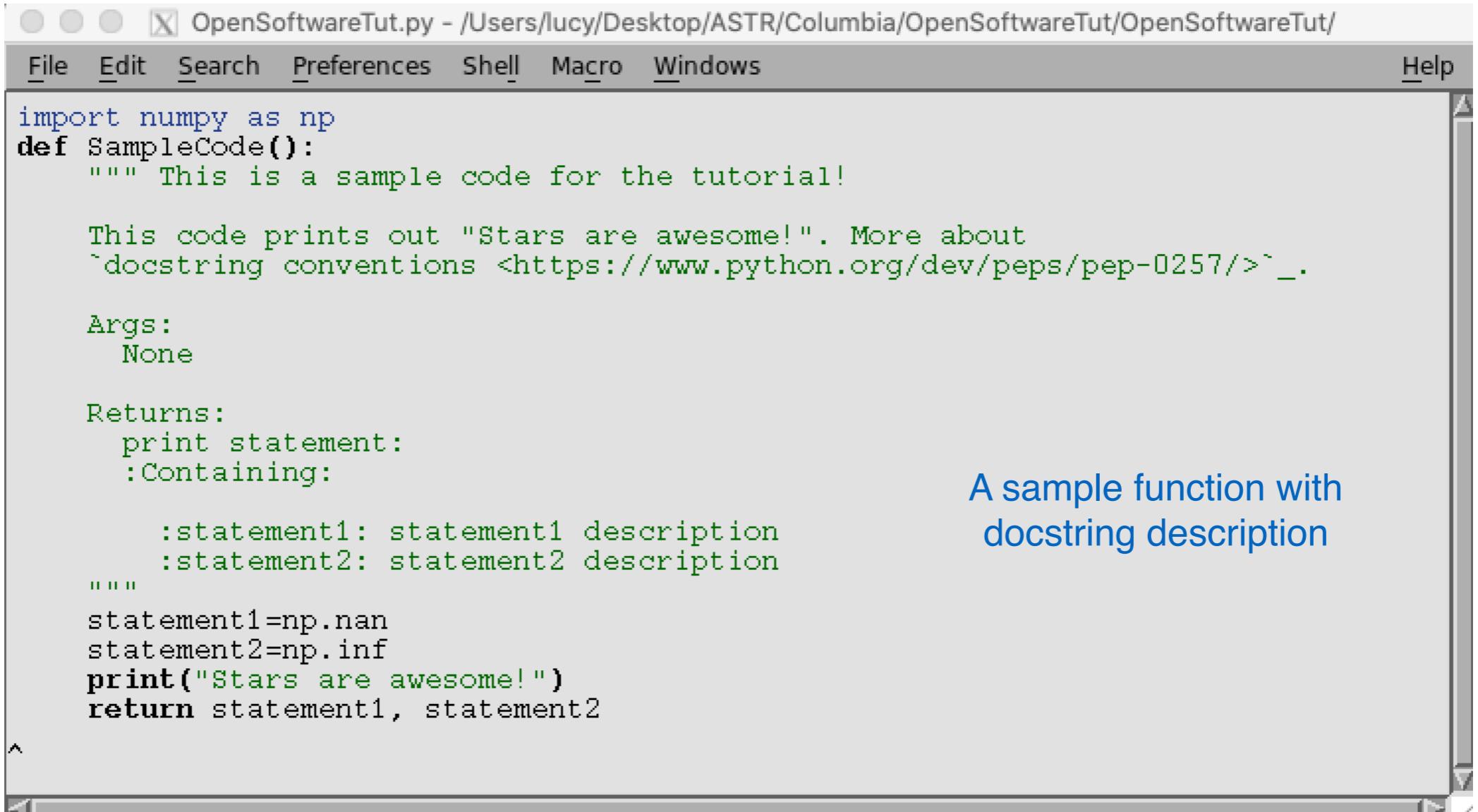
Create your code folder

- Under your main folder:

- mkdir OpenSoftwareTut
- touch setup.py

- Under your code folder (OpenSoftwareTut):

- Create version.py
- Create __init__.py
- Create your code (OpenSoftwareTut.py)



The screenshot shows a window titled "OpenSoftwareTut.py - /Users/lucy/Desktop/ASTR/Columbia/OpenSoftwareTut/OpenSoftwareTut/". The menu bar includes File, Edit, Search, Preferences, Shell, Macro, Windows, and Help. The code editor displays the following Python script:

```
import numpy as np
def SampleCode():
    """ This is a sample code for the tutorial!
        This code prints out "Stars are awesome!". More about
        docstring conventions <https://www.python.org/dev/peps/pep-0257/>.

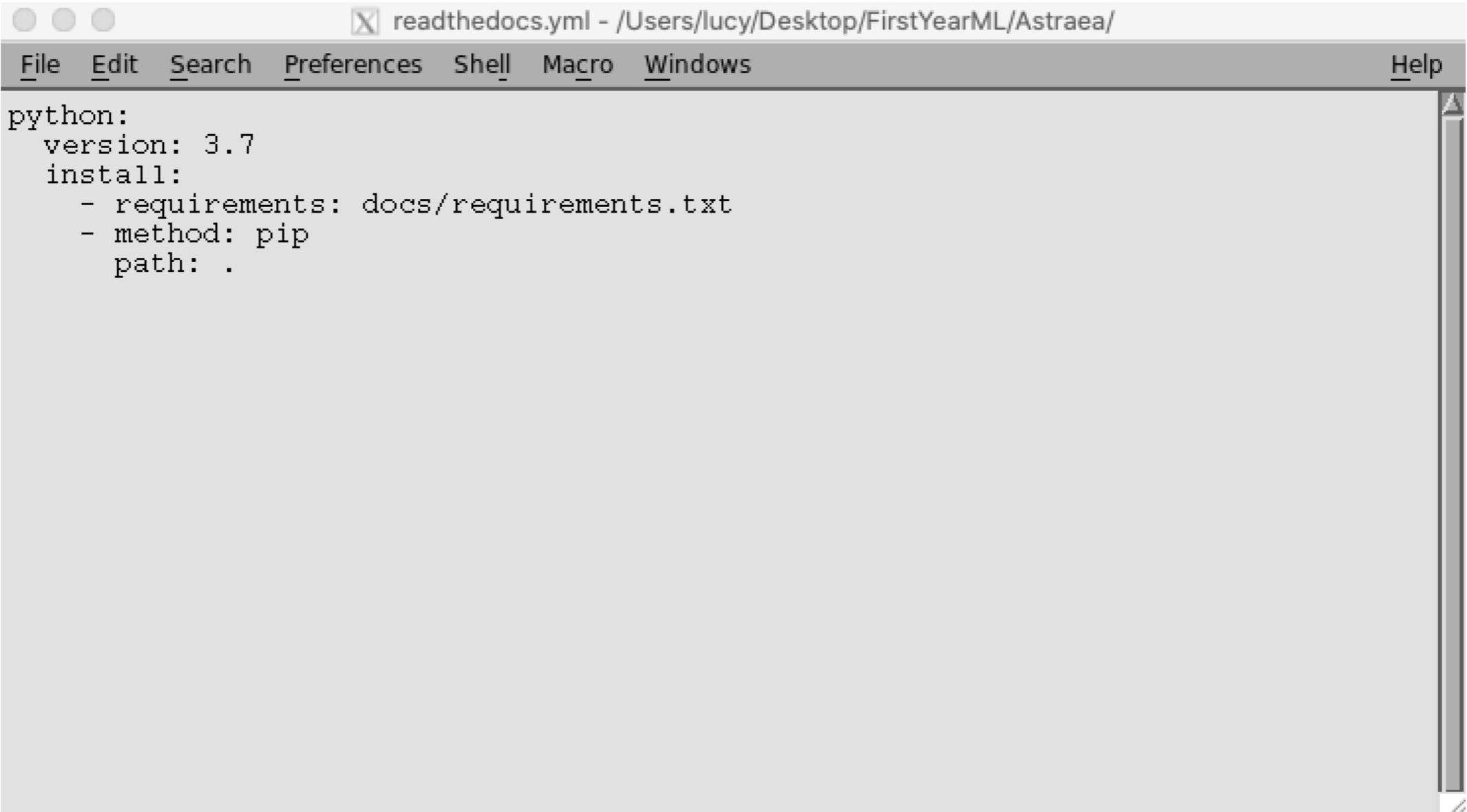
    Args:
        None

    Returns:
        print statement:
        :Containing:
            :statement1: statement1 description
            :statement2: statement2 description
        ...
        statement1=np.nan
        statement2=np.inf
        print("Stars are awesome!")
        return statement1, statement2
```

A sample function with docstring description

Create your documents

- You will need sphinx to create an website on [readthedocs.org](#)
- Make sure your GitHub account is connected with [readthedocs.org](#)
- Under your main folder
 - Create file called readthedocs.yml

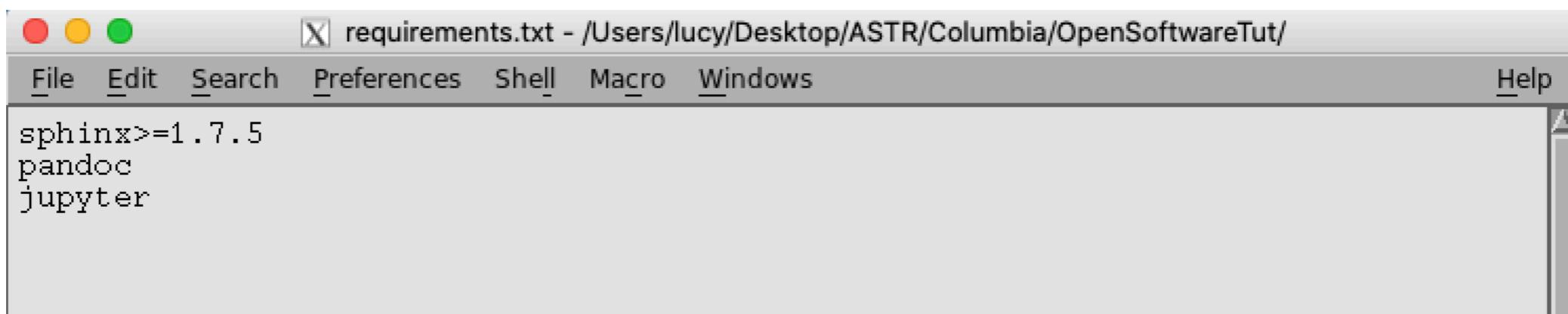


A screenshot of a code editor window titled "readthedocs.yml - /Users/lucy/Desktop/FirstYearML/Astrea/". The menu bar includes File, Edit, Search, Preferences, Shell, Macro, Windows, and Help. The code editor displays the following YAML configuration:

```
python:
  version: 3.7
  install:
    - requirements: docs/requirements.txt
    - method: pip
      path: .
```

Create your documents

- You will need sphinx to create an website on [readthedocs.org](#)
- Make sure your GitHub account is connected with [readthedocs.org](#)
- Under your main folder
 - Create file called readthedocs.yml
 - Create file called requirements.txt



Create your documents

- You will need sphinx to create an website on readthedocs.org
- Make sure your GitHub account is connected with readthedocs.org
- Under your main folder
 - Create file called readthedocs.yml
 - Create file called requirements.txt
 - mkdir docs/
 - I suggest you change your README.md to README.rst and I'll talk about what to do with it later on

Download sphinx and initialize your documents

Quick start

Assuming you have Python already, [install Sphinx](#):

```
$ pip install sphinx
```

Create a directory inside your project to hold your docs:

```
$ cd /path/to/project  
$ mkdir docs
```

Run `sphinx-quickstart` in there:

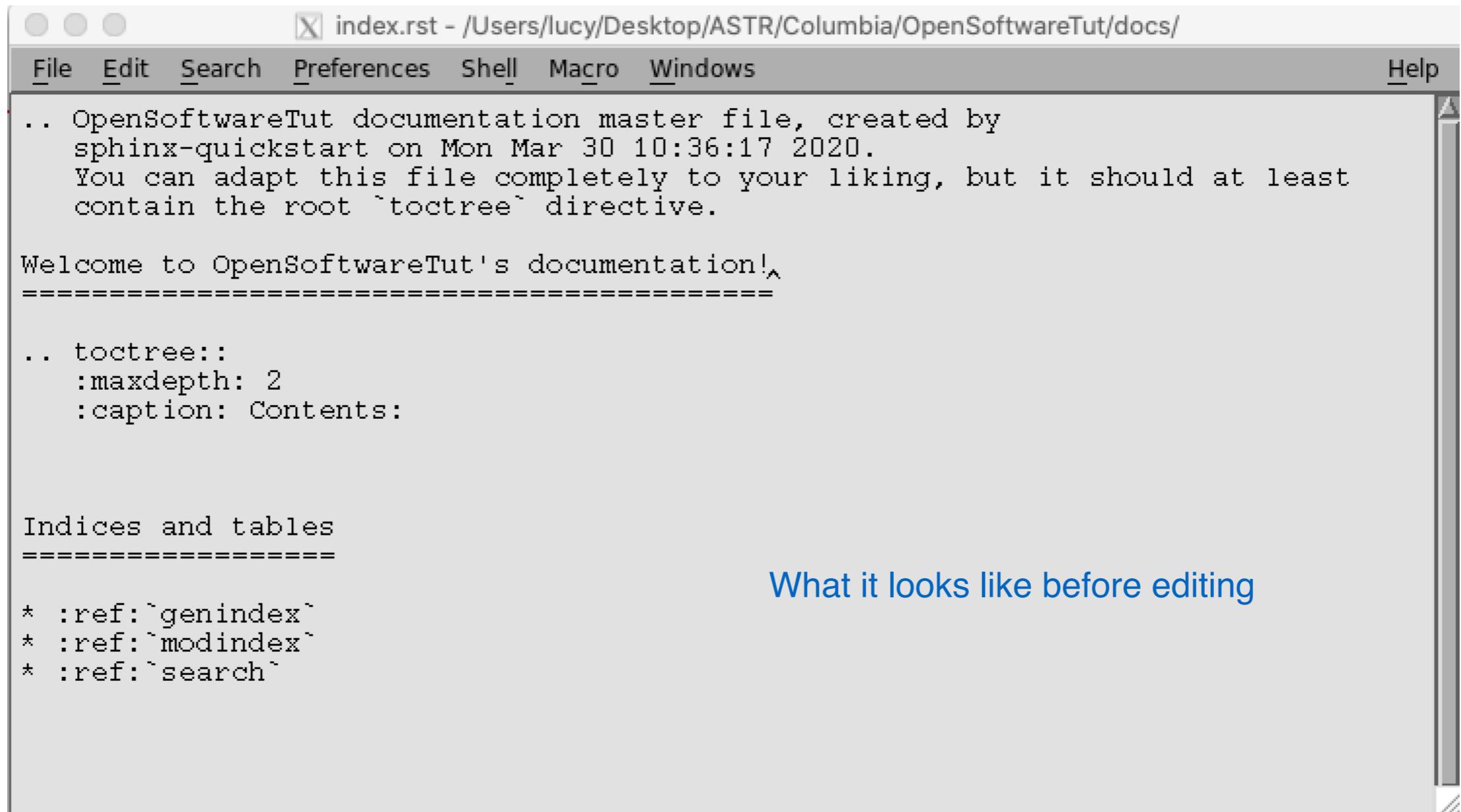
```
$ cd docs  
$ sphinx-quickstart
```

This quick start will walk you through creating the basic configuration; in most cases, you can just accept the defaults. When it's done, you'll have an `index.rst`, a `conf.py` and some other files. Add these to revision control.

Now, edit your `index.rst` and add some information about your project. Include as much detail as you like (refer to the [reStructuredText](#) syntax or [this template](#) if you need help). Build them to see how they look:

```
$ make html
```

Edit your index.rst page (website front page)



The screenshot shows a text editor window titled "index.rst - /Users/lucy/Desktop/ASTR/Columbia/OpenSoftwareTut/docs/". The menu bar includes File, Edit, Search, Preferences, Shell, Macro, Windows, and Help. The main content area contains the following text:

```
.. OpenSoftwareTut documentation master file, created by
sphinx-quickstart on Mon Mar 30 10:36:17 2020.
You can adapt this file completely to your liking, but it should at least
contain the root `toctree` directive.

Welcome to OpenSoftwareTut's documentation!
=====
.. toctree::
:maxdepth: 2
:caption: Contents:

Indices and tables
=====
* :ref:`genindex`
* :ref:`modindex`
* :ref:`search`
```

What it looks like before editing

Edit your index.rst page (website front page)

The screenshot shows a terminal window with the title bar "index.rst - /Users/lucy/Desktop/ASTR/Columbia/OpenSoftwareTut/docs/". The menu bar includes File, Edit, Search, Preferences, Shell, Macro, Windows, and Help. The main text area contains the following content:

```
.. OpenSoftwareTut documentation master file, created by
sphinx-quickstart on Mon Mar 30 10:36:17 2020.
You can adapt this file completely to your liking, but it should at least
contain the root `toctree` directive.

OpenSoftwareTut Project name
=====
*OpenSoftwareTut* is a sample project tutorial to publish open-source softwares.
This is how to add in a `simple link <https://ibb.co/4Sc6wv1>`_.

.. Contents:
User Guide
-----
.. toctree::
:maxdepth: 2
    user/install
    user/tests
    user/api

Tutorials
-----
```

A red box highlights the section starting with "Project name". A red dashed line separates the "Project name" section from the "Contents" section. A red box also highlights the "Contents" section.

Project name

Project description

These dashed lines separate sections

Contents

What it looks like after editing (you can customize it but this is how I like to do it)

Edit your index.rst page (website front page)

when someone finds your
old projects on github



Pls dnt tell anyone

Edit your index.rst page (website front page)

The screenshot shows a text editor window with a red border around the left portion of the content area. The title bar reads "index.rst - /Users/lucy/Desktop/ASTR/Columbia/OpenSoftwareTut/docs/". The menu bar includes File, Edit, Search, Preferences, Shell, Macro, Windows, and Help. The main content area contains the following text:

```
User Guide
-----
.. toctree::
:maxdepth: 2

user/install
user/tests
user/api

Tutorials
-----
.. toctree::
:maxdepth: 2

tutorial/Tutorial

License & attribution
-----
Copyright 2020, Yuxi Lu.
```

A red box highlights the first section, "User Guide". To the right of the highlighted section, the word "Contents" is written in red. At the bottom of the editor window, a message states: "The source code is made available under the terms of the MIT license."

Edit your conf.py (backend)

```
# Configuration file for the Sphinx documentation builder.
#
# This file only contains a selection of the most common options. For a full
# list see the documentation:
# https://www.sphinx-doc.org/en/master/usage/configuration.html

# -- Path setup -----
# If extensions (or modules to document with autodoc) are in another directory,
# add these directories to sys.path here. If the directory is relative to the
# documentation root, use os.path.abspath to make it absolute, like shown here.
#
# import os
# import sys
# sys.path.insert(0, os.path.abspath('.'))

# -- Project information -----
project = 'OpenSoftwareTut'
copyright = '2020, Lucy Lu'
author = 'Lucy Lu'

# -- General configuration -----
# Add any Sphinx extension module names here, as strings. They can be
# extensions coming with Sphinx (named 'sphinx.ext.*') or your custom
# ones.
extensions = [
]

# Add any paths that contain templates here, relative to this directory.
templates_path = ['_templates']

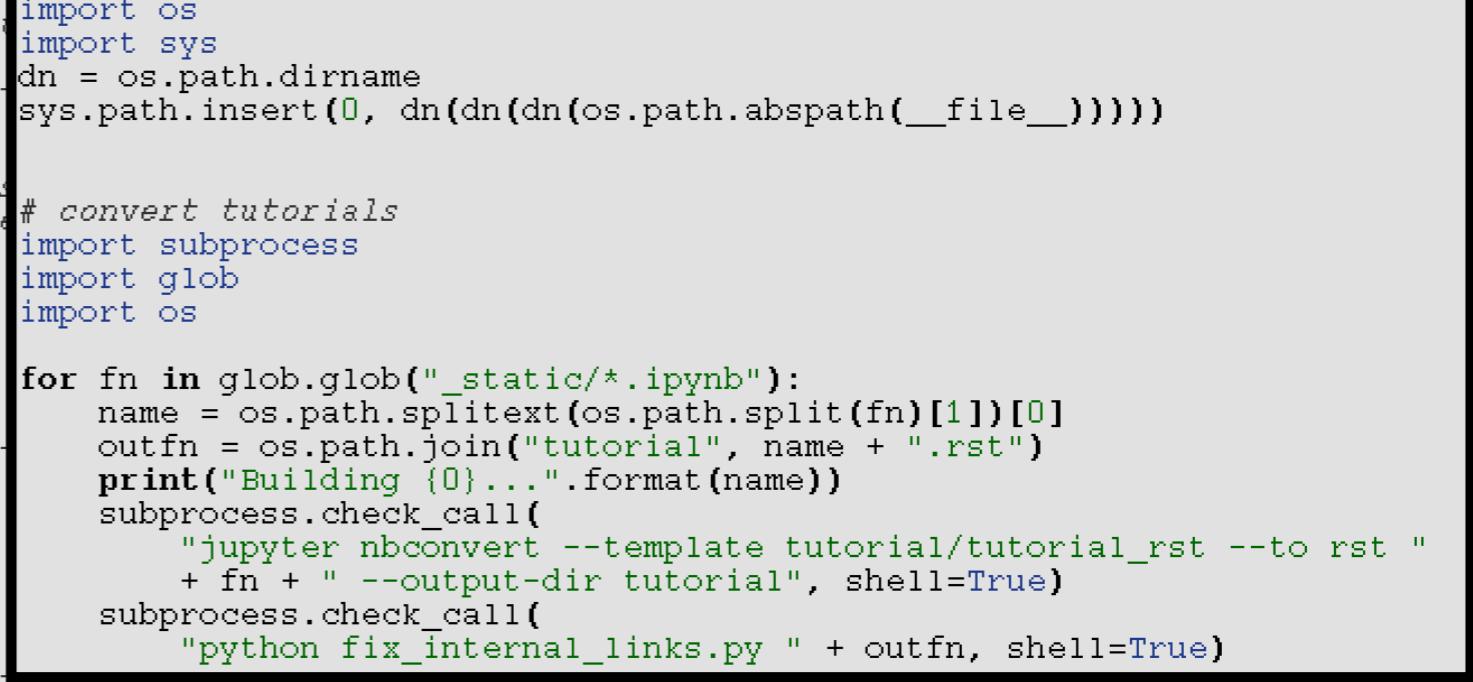
# List of patterns, relative to source directory, that match files and
# directories to ignore when looking for source files.
# This pattern also affects html_static_path and html_extra_path.
exclude_patterns = ['_build', 'Thumbs.db', '.DS_Store']

# -- Options for HTML output -----
# The theme to use for HTML and HTML Help pages. See the documentation for
# a list of builtin themes.
#
html_theme = 'alabaster'

# 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']
```

What it looks like before editing

Edit your conf.py (backend)



```
# Configuration file for the Sphinx documentation builder.
#
# This file only contains a selection of the most common options. For a full
# list see the documentation:
# https://www.sphinx-doc.org/en/master/usage/configuration.html
#
# -- Path setup -----
#
# If extensions (or modules to document with autodoc) are in
# add these directories to sys.path here. If the directory is
# documentation root, use os.path.abspath to make it absolute
#
# import os
# import sys
# sys.path.insert(0, os.path.abspath('.'))
#
# -- Project information -----
#
project = 'OpenSoftwareTut'
copyright = '2020, Lucy Lu'
author = 'Lucy Lu'
#
# -- General configuration -----
#
# Add any Sphinx extension module names here, as strings. They can be
# extensions coming with Sphinx (named 'sphinx.ext.*') or your custom
# ones.
extensions = [
]
#
# Add any paths that contain templates here, relative to this directory.
templates_path = ['_templates']
#
# List of patterns, relative to source directory, that match files and
# directories to ignore when looking for source files.
# This pattern also affects html_static_path and html_extra_path.
exclude_patterns = ['_build', 'Thumbs.db', '.DS_Store']
#
# -- Options for HTML output -----
#
# The theme to use for HTML and HTML Help pages. See the documentation for
# a list of builtin themes.
#
html_theme = 'alabaster'
#
# 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']
```

import os
import sys
dn = os.path.dirname
sys.path.insert(0, dn(dn(os.path.abspath(__file__))))

convert tutorials
import subprocess
import glob
import os

for fn in glob.glob("_static/*.ipynb"):
 name = os.path.splitext(os.path.split(fn)[1])[0]
 outfn = os.path.join("tutorial", name + ".rst")
 print("Building {}".format(name))
 subprocess.check_call(
 "jupyter nbconvert --template tutorial/tutorial_rst --to rst " + fn + " --output-dir tutorial", shell=True)
 subprocess.check_call(
 "python fix_internal_links.py " + outfn, shell=True)

Insert these lines (this gives you
the ability to convert your Jupiter
notebook into a html page. Useful
for Tutorials)

Edit your conf.py (backend)

```
X conf.py - /Users/lucy/Desktop/ASTR/Columbia/OpenSoftwareTut/docs/
File Edit Search Preferences Shell Macro Windows Help

# Configuration file for the Sphinx documentation builder.
#
# This file only contains a selection of the most common options. For a full
# list see the documentation:
# https://www.sphinx-doc.org/en/master/usage/configuration.html

# -- Path setup ----

# If extensions (or modules to document with autodoc) are in another directory,
# add these directories to sys.path here. If the directory is relative to the
# documentation root, use os.path.abspath to make it absolute, like shown here.
#
# import os
# import sys
# sys.path.insert(0, os.path.abspath('.'))

# -- Project information ----

project = 'OpenSoftwareTut'
copyright = '2020, Lucy Lu'
author = 'Lucy Lu'

# -- General configuration ----

# Add any Sphinx extension module names here, as strings. They can be
# extensions coming with Sphinx (named 'sphinx.ext.*') or your custom
# ones.
extensions = [  ←.....]
]

# Add any paths that contain templates here, relative to this directory.
templates_path = ['_templates']

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

# -- Options for HTML output ----

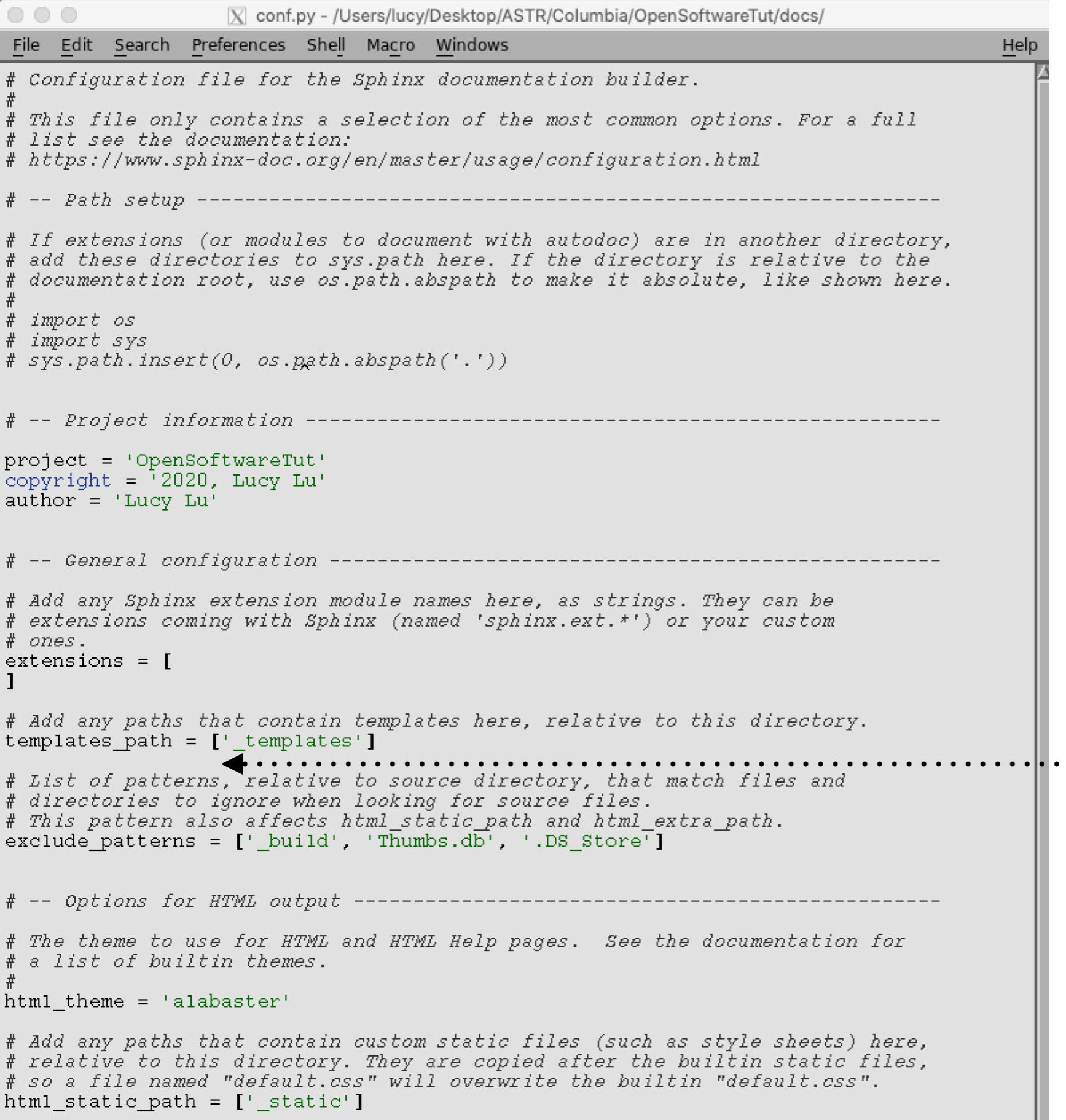
# The theme to use for HTML and HTML Help pages. See the documentation for
# a list of builtin themes.
#
html_theme = 'alabaster'

# 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']
```

```
extensions = [
    'sphinx.ext.autodoc',
    'sphinx.ext.mathjax',
    'sphinx.ext.viewcode',
    'sphinx.ext.napoleon',
]
```

Insert these extensions (these are used to convert document to html)

Edit your conf.py (backend)



```
# Configuration file for the Sphinx documentation builder.
#
# This file only contains a selection of the most common options. For a full
# list see the documentation:
# https://www.sphinx-doc.org/en/master/usage/configuration.html

# -- Path setup ----

# If extensions (or modules to document with autodoc) are in another directory,
# add these directories to sys.path here. If the directory is relative to the
# documentation root, use os.path.abspath to make it absolute, like shown here.
#
# import os
# import sys
# sys.path.insert(0, os.path.abspath('.'))

# -- Project information ----

project = 'OpenSoftwareTut'
copyright = '2020, Lucy Lu'
author = 'Lucy Lu'

# -- General configuration ----

# Add any Sphinx extension module names here, as strings. They can be
# extensions coming with Sphinx (named 'sphinx.ext.*') or your custom
# ones.
extensions = [
]

# Add any paths that contain templates here, relative to this directory.
templates_path = ['_templates']

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

# -- Options for HTML output ----

# The theme to use for HTML and HTML Help pages. See the documentation for
# a list of builtin themes.
#
html_theme = 'alabaster'

# 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']
```

master_doc = 'index'▲

Insert this line (tells the document where the front page document is named)

Edit your conf.py (backend)

```
# Configuration file for the Sphinx documentation builder.
#
# This file only contains a selection of the most common options. For a full
# list see the documentation:
# https://www.sphinx-doc.org/en/master/usage/configuration.html

# -- Path setup --
# If extensions (or modules to document with autodoc) are in another directory,
# add these directories to sys.path here. If the directory is relative to the
# documentation root, use os.path.abspath to make it absolute, like shown here.
#
# import os
# import sys
# sys.path.insert(0, os.path.abspath('.'))

# -- Project information --
project = 'OpenSoftwareTut'
copyright = '2020, Lucy Lu'
author = 'Lucy Lu'

# -- General configuration --
# Add any Sphinx extension module names here, as strings. They can be
# extensions coming with Sphinx (named 'sphinx.ext.*') or your custom
# ones.
extensions = [
]

# Add any paths that contain templates here, relative to this directory.
templates_path = ['_templates']

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

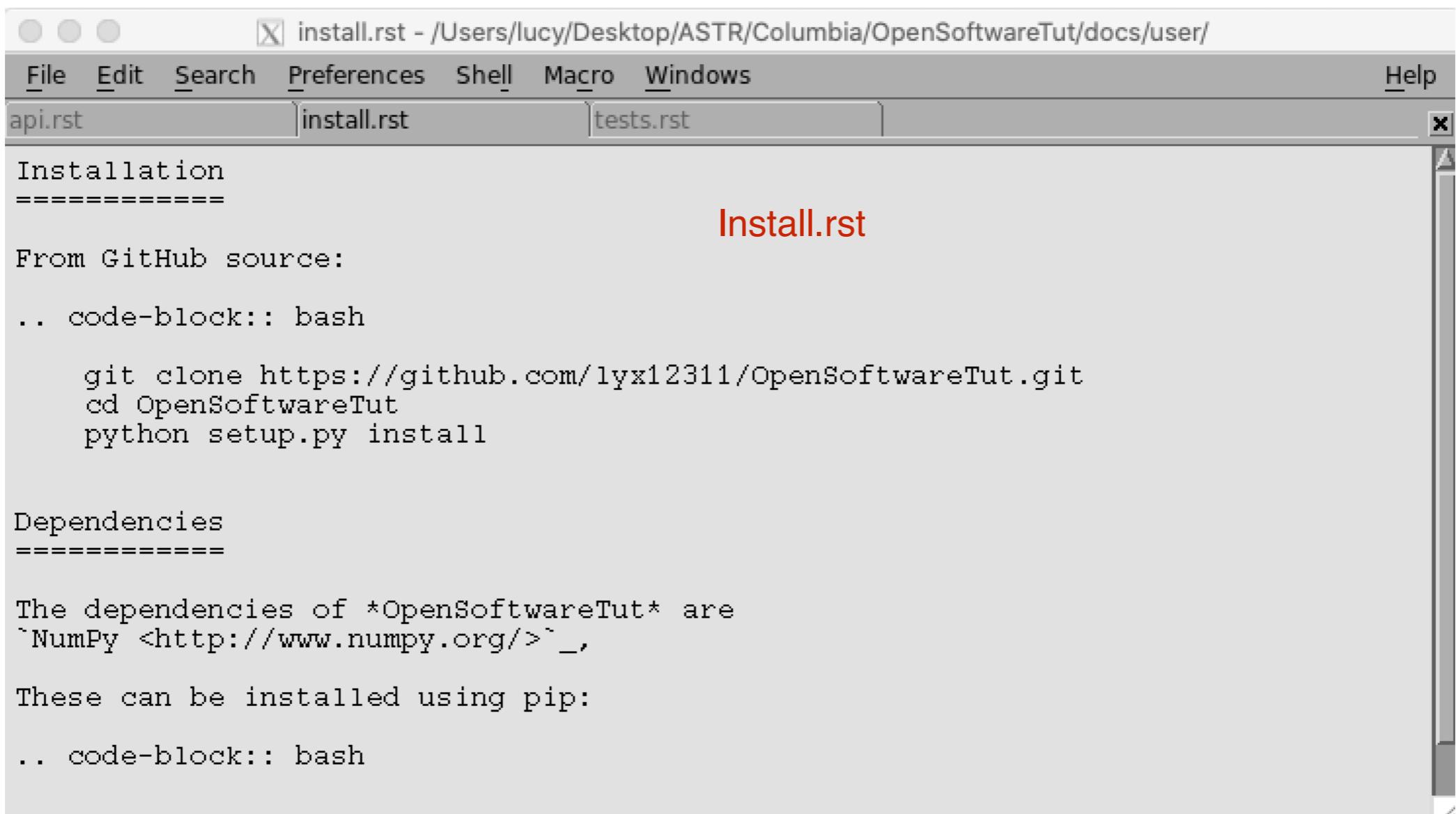
# -- Options for HTML output --
# The theme to use for HTML and HTML Help pages. See the documentation for
# a list of builtin themes.
#
# pygments_style = 'sphinx'
#
# import os
# on_rtd = os.environ.get("READTHEDOCS", None) == "True"
# if not on_rtd:
#     import sphinx_rtd_theme
#     html_theme = "sphinx_rtd_theme"
#     html_theme_path = [sphinx_rtd_theme.get_html_theme_path()]

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

Replace this line with these (set the document styles)

Edit your user guide

- In your docs/ folder:
 - mkdir user
- In your docs/user/ folder:
 - Create api.rst install.rst tests.rst



The screenshot shows a window titled "install.rst - /Users/lucy/Desktop/ASTR/Columbia/OpenSoftwareTut/docs/user/". The window has a menu bar with File, Edit, Search, Preferences, Shell, Macro, Windows, and Help. Below the menu is a toolbar with three buttons: api.rst, install.rst, and tests.rst. The main text area contains the following content:

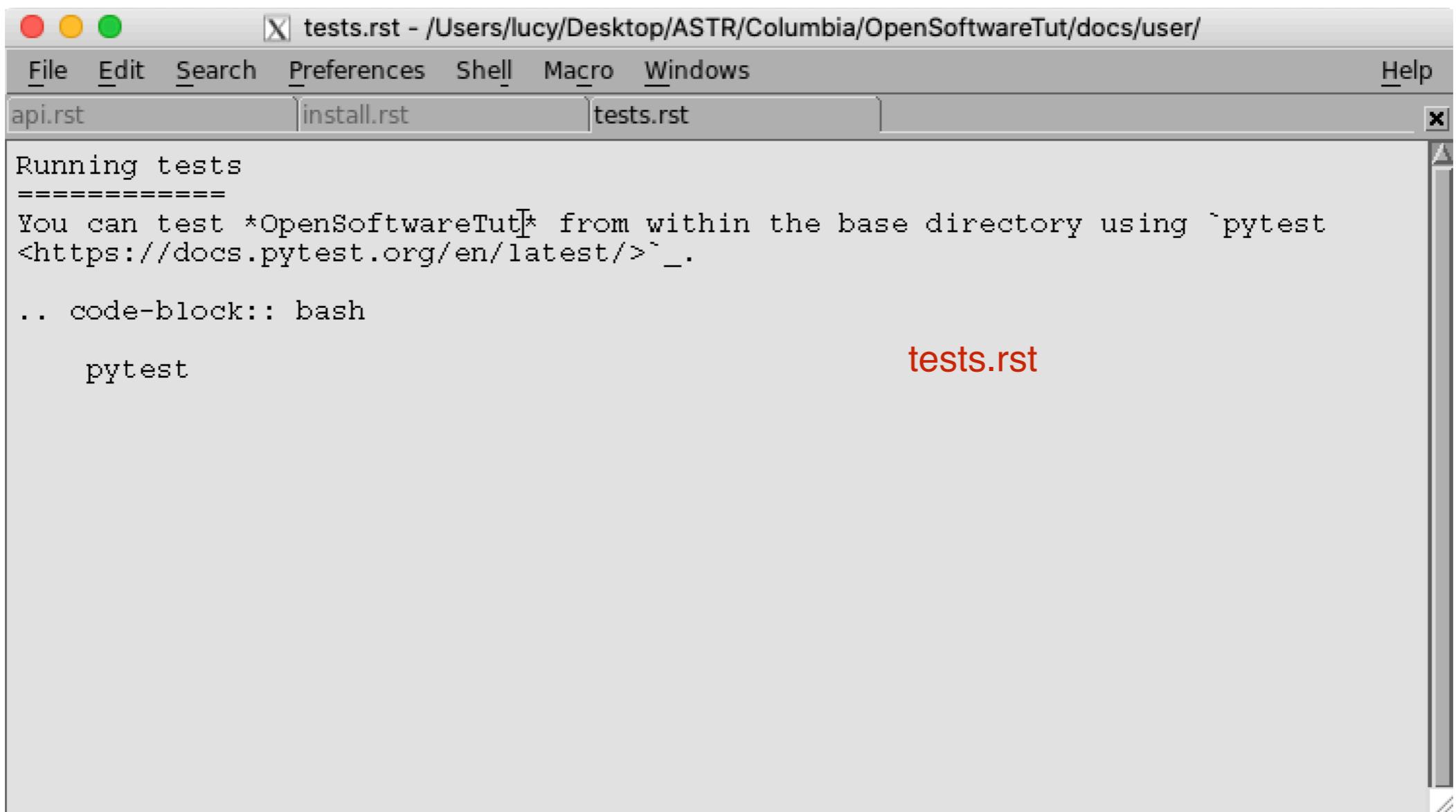
```
Installation
=====
Install.rst
From GitHub source:
... code-block:: bash
    git clone https://github.com/lyx12311/OpenSoftwareTut.git
    cd OpenSoftwareTut
    python setup.py install

Dependencies
=====
The dependencies of *OpenSoftwareTut* are
`NumPy <http://www.numpy.org/>`_,

These can be installed using pip:
... code-block:: bash
```

Edit your user guide

- In your docs/ folder:
 - mkdir user
- In your docs/user/ folder:
 - Create api.rst install.rst tests.rst



The screenshot shows a terminal window with the title bar "tests.rst - /Users/lucy/Desktop/ASTR/Columbia/OpenSoftwareTut/docs/user/". The menu bar includes File, Edit, Search, Preferences, Shell, Macro, Windows, and Help. Below the menu is a tab bar with three tabs: "api.rst", "install.rst", and "tests.rst", where "tests.rst" is the active tab. The main pane displays the following text:

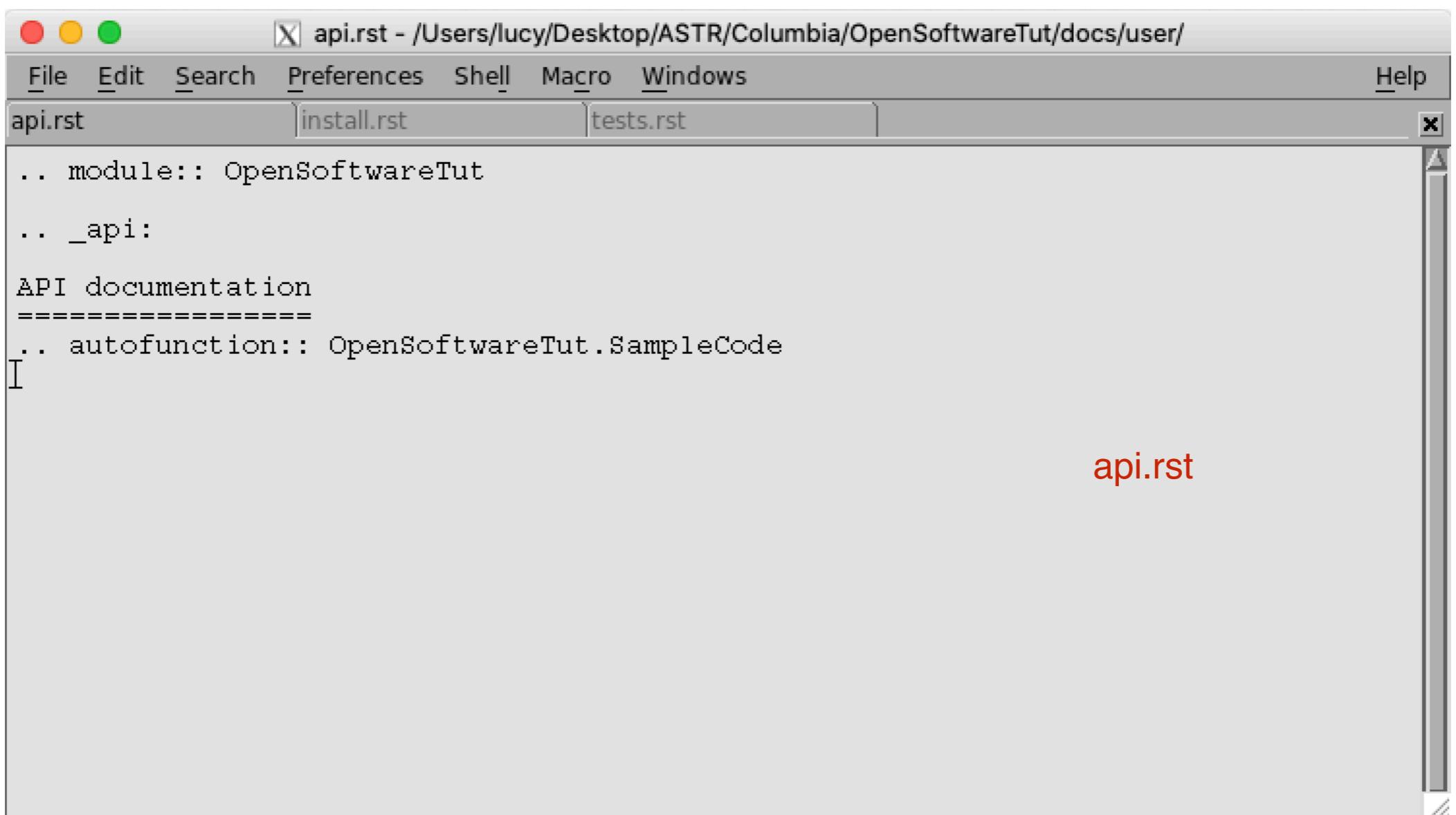
```
Running tests
=====
You can test *OpenSoftwareTut* from within the base directory using `pytest
<https://docs.pytest.org/en/latest/>`_.

... code-block:: bash
    pytest
```

tests.rst

Edit your user guide

- In your docs/ folder:
 - mkdir user
- In your docs/user/ folder:
 - Create api.rst install.rst tests.rst



The screenshot shows a terminal window titled "api.rst - /Users/lucy/Desktop/ASTR/Columbia/OpenSoftwareTut/docs/user/". The window has a menu bar with File, Edit, Search, Preferences, Shell, Macro, Windows, and Help. Below the menu is a tab bar with three tabs: "api.rst", "install.rst", and "tests.rst", where "api.rst" is the active tab. The main pane displays the following RST content:

```
.. module:: OpenSoftwareTut
.. _api:
API documentation
=====
.. autofunction:: OpenSoftwareTut.SampleCode
```

A red label "api.rst" is overlaid on the bottom right of the terminal window.

Edit your user guide

- In your docs/ folder:
 - mkdir user
- In your docs/user/ folder:
 - Create api.rst install.rst tests.rst
- In your docs/_static/ folder:
 - Create your Jupyter notebook tutorial: Tutorial.ipynb

This is a main title

This is a subtitle

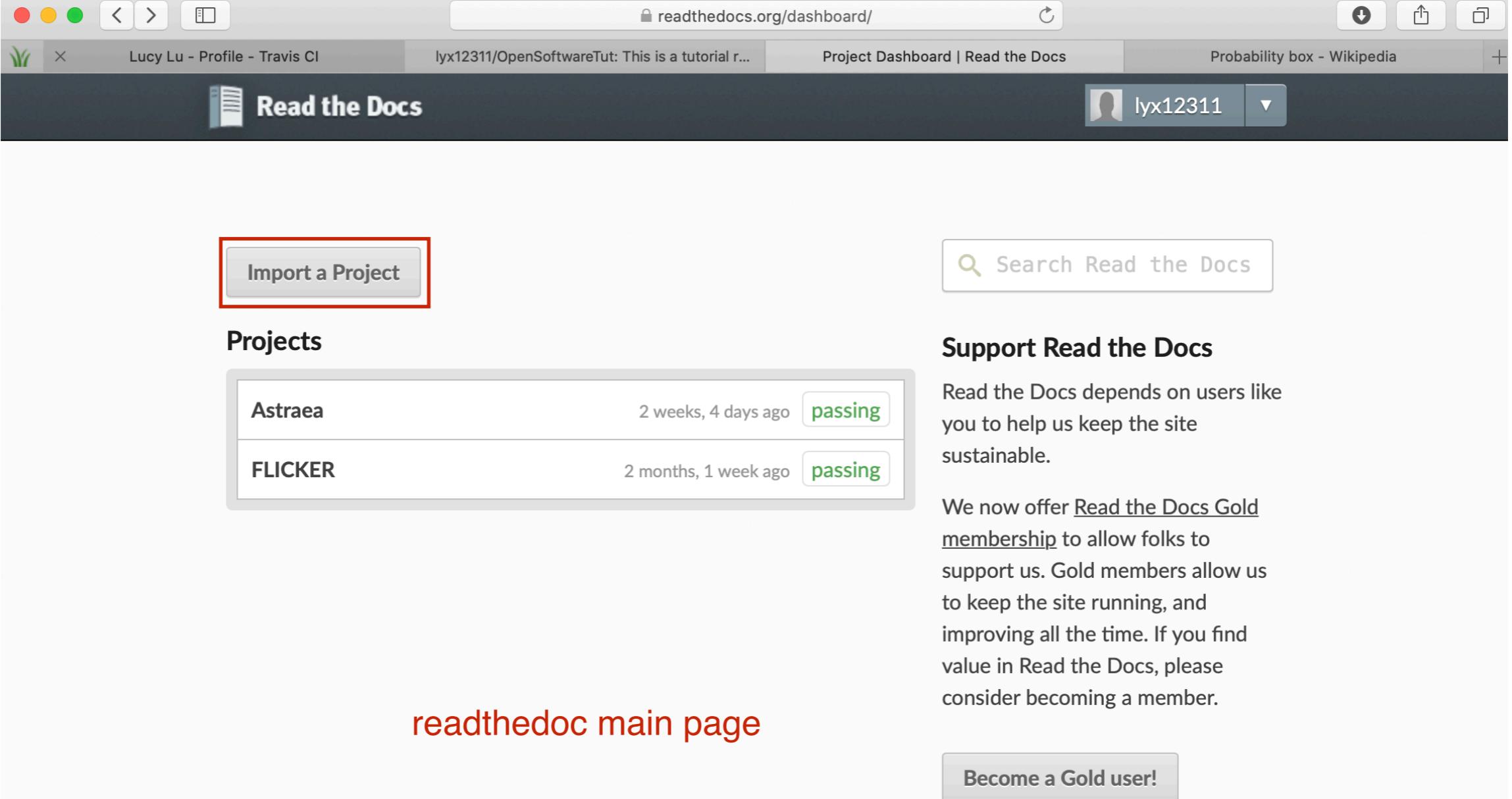
```
In [1]: 1 import numpy as np
          2 import OpenSoftwareTut
          3
          4 s1, s2=OpenSoftwareTut.SampleCode()
          5 print(s1,s2)
```

Stars are awesome!
nan inf

Edit your user guide

- In your docs/ folder:
 - mkdir user
- In your docs/user/ folder:
 - Create api.rst install.rst tests.rst
- In your docs/_static/ folder:
 - Create your Jupyter notebook tutorial: Tutorial.ipynb
- In your docs/ folder:
 - mkdir tutorial
 - copy fix_internal_links.py in docs/
 - copy tutorial_rst.tpl in docs/tutorial/

Create your readthedocs page



The screenshot shows a web browser window with the URL `readthedocs.org/dashboard/` in the address bar. The title bar includes tabs for "Lucy Lu - Profile - Travis CI", "lyx12311/OpenSoftwareTut: This is a tutorial r...", "Project Dashboard | Read the Docs", and "Probability box - Wikipedia". The main content area is the Read the Docs dashboard. At the top left is a button labeled "Import a Project" with a red box drawn around it. To the right is a search bar labeled "Search Read the Docs". Below the search bar is a section titled "Support Read the Docs" with text about the site's dependency on user support and a "Become a Gold user!" button. In the center, there is a "Projects" section displaying two projects: "Astrea" (2 weeks, 4 days ago, passing) and "FLICKER" (2 months, 1 week ago, passing). A large red watermark "readthedoc main page" is overlaid across the bottom left of the dashboard.

Import a Project

Projects

Astrea 2 weeks, 4 days ago passing

FLICKER 2 months, 1 week ago passing

readthedoc main page

Search Read the Docs

Support Read the Docs

Read the Docs depends on users like you to help us keep the site sustainable.

We now offer [Read the Docs Gold membership](#) to allow folks to support us. Gold members allow us to keep the site running, and improving all the time. If you find value in Read the Docs, please consider becoming a member.

Become a Gold user!

Learn More

Check out the [documentation for Read the Docs](#). It contains lots of information about how to get the most out of RTD.

Create your readthedocs page

Lucy Lu - Profile - Travis CI lyx12311/OpenSoftwareTut: This is a tutorial r... Import a Remote Repository | Read the Docs Probability box - Wikipedia

We're only showing public repositories. For private projects and many of features, please use [Read the Docs for Business](#).

manually if it isn't listed here or connected to one of your accounts.

[Import Manually](#)

Filter repositories

lyx12311

lyx12311/ASTR310 https://github.com/lyx12311/ASTR310.git	[+]
lyx12311/Astrea https://github.com/lyx12311/Astrea.git	[x]
joehahn/epi_int_lite https://github.com/joehahn/epi_int_lite.git	[+]
lyx12311/epi_int_lite https://github.com/lyx12311/epi_int_lite.git	[+]
lyx12311/FLICKER https://github.com/lyx12311/FLICKER.git	[x]
lyx12311/HJspinup https://github.com/lyx12311/HJspinup.git	[+]
lyx12311/Mdwarp https://github.com/lyx12311/Mdwarp.git	[+]
lyx12311/NarrowRing https://github.com/lyx12311/NarrowRing.git	[+]
lyx12311/OpenSoftwareTut https://github.com/lyx12311/OpenSoftwareTut.git	[+]
lyx12311/SaturnRing https://github.com/lyx12311/SaturnRing.git	[+]
lyx12311/spokes https://github.com/lyx12311/spokes.git	[+]

Create your readthedocs page

The screenshot shows a web browser window for the project `OpenSoftwareTut` on [readthedocs.org](https://readthedocs.org/projects/opensoftwaretut/builds/10726972/). The URL in the address bar is `readthedocs.org/projects/opensoftwaretut/builds/10726972/`. The browser's top navigation bar includes icons for back, forward, and search, along with tabs for the current project and other sites like Wikipedia.

The main content area shows the project's documentation type as `Sphinx Html`, requirements file as empty, Python interpreter as `CPython 3.x`, and the `Install Project` checkbox selected. A note on the right says `Make sure to save afterwards`.

Project Navigation: Projects > OpenSoftwareTut

Project Settings:

- Overview
- Downloads
- Search
- Builds
- Versions
- Admin** (highlighted with a red box)

Documentation type*: Sphinx Html

Type of documentation you are building. [More info on sphinx builders.](#)

Requirements file: (empty input field)

A [pip requirements file](#) needed to build your documentation. Path from the root of your project.

Python Interpreter*: CPython 3.x

The Python interpreter used to create the virtual environment.

Install Project (checkbox checked, highlighted with a red box)

Install your project inside a virtualenv using `setup.py install`

Use system packages (checkbox unselected)

Give the virtual environment access to the global site-packages dir.

Note: Make sure to save afterwards

Create your readthedocs page

The screenshot shows a web browser window with the URL `readthedocs.org/projects/opensoftwaretut/builds/10726972/` in the address bar. The title bar includes standard OS X icons (red, yellow, green) and a tab labeled "Lucy Lu - Profile - Travis CI". The main content area displays the "OpenSoftwareTut" project page. At the top, there's a navigation bar with "Read the Docs" and a user profile for "lyx12311". Below this, the project name "OpenSoftwareTut" is shown with a "View Docs" button. A navigation menu at the bottom includes "Overview", "Downloads", "Search", "Builds" (which is highlighted with a red box), "Versions", and "Admin". In the lower half of the page, a section titled "Recent Builds" features a dropdown menu for "Build Version:" with "latest" selected. The entire screenshot is framed by a thick black border.

Create your readthedocs page

The screenshot shows the readthedocs project dashboard for a project named "OpenSoftwareTut". At the top left, there's a breadcrumb navigation "Projects > OpenSoftwareTut". On the right side, there's a prominent green button with the text "View Docs" inside a red-bordered box. Below the header, there are six navigation tabs: "Overview" (which is highlighted in white), "Downloads", "Search", "Builds" (which is dark grey), "Versions", and "Admin". The main content area displays a build summary for "Build #10727358". It includes the build status "Completed March 30, 2020. 5:30 p.m.", the duration "Build took 47 seconds", and two links: "View docs" and "View raw". A green button labeled "Build completed" is also present. At the bottom, there's a note about configuring documentation builds with a ".readthedocs.yml" file.

Projects >
OpenSoftwareTut

[View Docs](#)

[Overview](#) [Downloads](#) [Search](#) [Builds](#) [Versions](#) [Admin](#)

Build #10727358

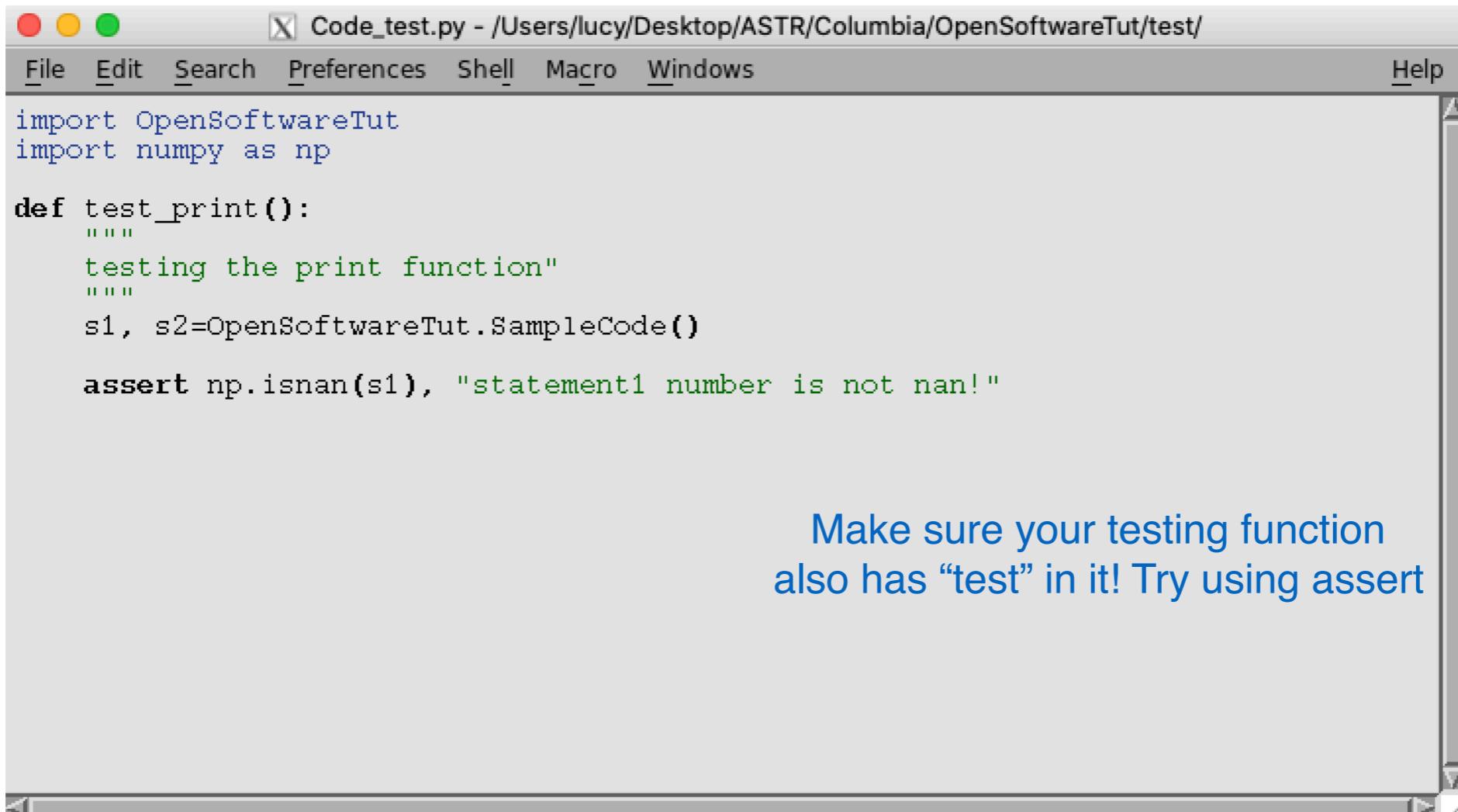
Completed March 30, 2020. 5:30 p.m.
Build took 47 seconds
[View docs](#)
[View raw](#)

Build completed

Configure your documentation builds! Adding a [.readthedocs.yml](#) file to your project is the recommended way to configure your documentation builds. You can declare dependencies, set up submodules, and many other great features.

Writing tests for your code

- Under your main folder:
 - mkdir test/
- Under your test/ folder:
 - Write all your tests with “test” in the filename. I wrote mine called Code_test.py



```
Code_test.py - /Users/lucy/Desktop/ASTR/Columbia/OpenSoftwareTut/test/
File Edit Search Preferences Shell Macro Windows Help
import OpenSoftwareTut
import numpy as np

def test_print():
    """
    testing the print function
    """
    s1, s2=OpenSoftwareTut.SampleCode()

    assert np.isnan(s1), "statement1 number is not nan!"
```

Make sure your testing function also has “test” in it! Try using assert

Writing tests for your code

- Under your main folder:
 - mkdir test/
- Under your test/ folder:
 - Write all your tests with “test” in the filename. I wrote mine called Code_test.py
- Now you can go back to your main folder:
 - python setup.py install
 - pytest

```
(base) lucys-MBP-2:OpenSoftwareTut lucy$ pytest
=====
          test session starts
=====
platform darwin -- Python 3.7.3, pytest-5.2.2, py-1.8.0, pluggy-0.13.0
rootdir: /Users/lucy/Desktop/ASTR/Columbia/OpenSoftwareTut
plugins: arraydiff-0.3, remotedata-0.3.2, doctestplus-0.4.0, openfiles-0.4.0
collected 1 item

test/Code_test.py .

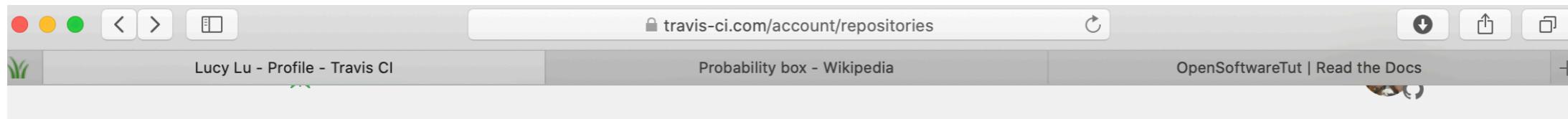
=====
      1 passed in 0.04s =====
```

Output [100%]

Writing tests for your code

- Under your main folder:
 - mkdir test/
- Under your test/ folder:
 - Write all your tests with “test” in the filename. I wrote mine called Code_test.py
- Now you can go back to your main folder:
 - python setup.py install
 - pytest
- Under your main folder:
 - Copy file .travis.yml

Test on Travis



MY ACCOUNT



Lucy Lu

[Sync account](#)

ORGANIZATIONS

You are not currently a member of any organization.

MISSING AN ORGANIZATION?

Review and add your authorized organizations.

 Lucy Lu
@lyx12311

[Repositories](#) [Settings](#) [Subscription](#) [Migrate](#)

GitHub Apps Integration

[Manage repositories on GitHub](#)

[Filter GitHub repositories](#)

 ASTR310	Settings
 Astraea	Settings
 epi_int_lite	Settings
 FLICKER	Settings
 HJspinup	Settings
 Mdwarf	Settings
 NarrowRing	Settings
 OpenSoftwareTut	Settings
 SaturnRing	Settings
 spokes	Settings

Test on Travis

Travis CI



Dashboard

Changelog

Documentation

Help



Search all repositories



lyx12311 / OpenSoftwareTut



build unknown

My Repositories

Running (3/3)



✓ lyx12311/Astrea

23

⌚ Duration: 11 min 58 sec

⌚ Finished: 18 days ago

○ lyx12311/HJspinup



⌚ Duration: -

○ lyx12311/Mdwarf



⌚ Duration: -

○ lyx12311/spokes



⌚ Duration: -

○ lyx12311/epi_int_lite



⌚ Duration: -

○ lyx12311/SaturnRing



⌚ Duration: -

○ lyx12311/ASTR310



⌚ Duration: -

Current

Branches

Build History

Pull Requests

More options



○ master travis file

-o #1 started

-o Commit 1b483fb ↗

⌚ Running for 39 sec

↳ Compare a97880b..1b483fb ↗

↳ Branch master ↗

Lucy Lu

Build jobs

View config

○ # 1.1



</> Python: 3.5

cube no environment variables set

⌚ 34 sec

○ # 1.2



</> Python: 3.6

cube no environment variables set

⌚ 39 sec

○ # 1.3



</> Python: 3.7

cube no environment variables set

⌚ 33 sec

Test on Travis

Travis CI



Dashboard

Changelog

Documentation

Help



Search all repositories



lyx12311 / OpenSoftwareTut



build unknown

My Repositories

Running (3/3)



✓ lyx12311/Astrea

23

⌚ Duration: 11 min 58 sec

⌚ Finished: 18 days ago

○ lyx12311/HJspinup



⌚ Duration: -

○ lyx12311/Mdwarf



⌚ Duration: -

○ lyx12311/spokes



⌚ Duration: -

○ lyx12311/epi_int_lite



⌚ Duration: -

○ lyx12311/SaturnRing



⌚ Duration: -

○ lyx12311/ASTR310



⌚ Duration: -

Current

Branches

Build History

Pull Requests

More options



○ master travis file

-o #1 started

-o Commit 1b483fb ↗

⌚ Running for 39 sec

↳ Compare a97880b..1b483fb ↗

↳ Branch master ↗

Lucy Lu

Build jobs

View config

○ # 1.1



</> Python: 3.5

cube no environment variables set

⌚ 34 sec

○ # 1.2



</> Python: 3.6

cube no environment variables set

⌚ 39 sec

○ # 1.3



</> Python: 3.7

cube no environment variables set

⌚ 33 sec

Adding badges on README.rst

On readthedocs.org

On [travis](https://travis-ci.com)

The screenshot shows a Mac desktop with two browser windows open. The left window is the 'OpenSoftwareTut' project page on readthedocs.org. It features a 'Builds' section with a 'docs passing' badge. The right window is the Travis CI dashboard for the same repository. It shows a 'Build History' section with a 'build passing' badge. Both badges are highlighted with red boxes.

```
File Edit Search Preferences Shell Macro Windows Help
=====
OpenSoftwareTut
=====
.. image:: https://travis-ci.com/lyx12311/OpenSoftwareTut.svg?branch=master
:target: https://travis-ci.com/lyx12311/OpenSoftwareTut

.. image::
https://readthedocs.org/projects/opensoftwaretut/badge/?version=latest
:target: https://opensoftwaretut.readthedocs.io/en/latest/?badge=latest
:alt: Documentation Status

This is a tutorial repository for publishing open software
Copy into README.rst
```