Sphinx Documentation

1. Build up documentation

1.1. Setup Sphinx project

Listing 1. Setup Sphinx project

```
# DO NOT directly copy paste into command line due to PDF formatting issues
$ python -m venv .venv
$ source .venv/bin/activate
(.venv) $ python -m pip install sphinx
(.venv) $ cd YOUR_PROJECT_DIRECTORY
(.venv) $ sphinx-quickstart docs
(.venv) $ cd docs
(.venv) $ make html
```

The above commands will automatically create `/docs` directory with the following contents and structure.

docs
|—- build
|—- make.bat
|—- Makefile
|—-source
    |—-conf.py
    |—-index.rst
    |—-_static
    |—-_templates

The purposes of each of these files are:

- build/:An empty directory (for now) that will hold the rendered documentation
- make.bat and Makefile: Convenience scripts to simplify some common Sphinx operations, such as rendering the content
- source/conf.py: A Python script holding the configuration of the Sphinx project. It contains the project name and release you specified to sphinx-quickstart, as well as some extra configuration keys
- source/index.rst: The root document of the project, which serves as welcome page and contains the root of the "table of contents tree" (or toctree)
- More details of project setups and illustrations

1.2. Next steps

You may start from following the example.

Basically you need to,

1. modify docs/source/conf.py
    - make sure your py file is **importable** by adding the directory to sys.path in docs/source/conf.py and including relevant extension in `extensions` list, e.g. sphinx.ext.autodoc

2. add or modify docs/source/*.rst files

   - Create one separate rst file (Sphinx converts rst file to html page correspondingly) for each module and use `automodule` to document it automatically (see the following example). Feel free to customize your rst file according to reStructuredText Guide section

Listing 2. usage.rst

```
Usage
=====
.. _foo:

mymodule
----------------------

.. automodule:: mymodule
:members:
:undoc-members:
:show-inheritance:
```

   - index.rst is the homepage of your Sphinx project. You need to create links to your module documentation page via referencing, `.. toctree:: [your_module_rst]` (see the following example)

Listing 3. index.rst

```
.. Sphinx Example documentation master file, created by
sphinx-quickstart on Fri Apr 22 14:05:20 2022.
You can adapt this file completely to your liking, but it should at least
contain the root `toctree` directive.

Welcome to a Sphinx Example!
====================================

Check out the :doc:`usage` section for further information, including how to use
:ref:`foo <foo>`

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

Contents
=========

.. toctree::

usage


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

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

3. After you add/modify conf.py, *.rst files, run `make html` under the doc directory to render the changes

2. Additional Reference

2.1. reStructuredText Guide

- reStructuredText is an easy-to-read, what-you-see-is-what-you-get plaintext markup syntax and parser system. It is useful for in-line program documentation (such as Python docstrings), for quickly creating simple web pages, and for

standalone documents. A short example and a comprehensive example

- You are recommended to start by following the examples, and look up further details from the reference below or online when needed
  - Basic Structured Text, Paragraphs: must have the same indentation
  - italics with `*italics*` or bold with `**bold**` or backquotes ``` ``text`` ``` for code samples
  - Enumerated lists: start a line off with a number or letter followed by a period `.`, right bracket `)` or surrounded by brackets `( )` e.g. `1.`, `1)`, `(1)`
  - Bulleted lists: start the line off with a bullet point character - either −, + or ⋆
  - Field Lists: `:fieldname:Field content`
  - Preformatting: to just include a chunk of preformatted, never-to-be-fiddled-with text, finish the prior paragraph with `::`. The preformatted block is finished when the text falls back to the same indentation level as a paragraph prior to the preformatted block
  - Sections: a single line of text (one or more words) with adornment: an underline alone, or an underline and an overline together, in dashes `-----`, equals `======`, tildes ∼∼∼∼∼ or any of the non-alpha numeric characters = − ´ : ′ ” ∼ ∧ _ ⋆ + # < > that you feel comfortable with. Be consistent, since all sections marked with the same adornment style are deemed to be at the same level
  - Images: `.. image::images/biohazard.png`
  - Doctest blocks: interactive Python sessions cut-and-pasted into docstrings. They are meant to illustrate usage by example, and provide an elegant and powerful testing environment via the doctest module in the Python standard library. Doctest blocks are text blocks which begin with >>>, the Python interactive interpreter main prompt, and end with a blank line
  - Grid tables: described with a visual grid made up of the characters −, =, |, and +. The hyphen (−) is used for horizontal lines (row separators). The equals sign (=) may be used to separate optional header rows from the table body. The vertical bar (|) is used for vertical lines (column separators). The plus sign (+) is used for intersections of horizontal and vertical lines
  - Hyperlinks: ``` `Link text <https://domain.invalid/>`_ ```
  - Cross-references: in index page, use `.. toctree::` or `:doc:` to link other pages or ``` :ref:`install <installation>` ``` to point to specifically the ”Installation” section, `.. _installation:` to reference the “Installation” subsection within the same page. More illustrations
- More reStructuredText syntax

2.2. Sphinx Extensions

- sphinx.ext.autodoc: import the modules you are documenting, and pull in documentation from docstrings
- sphinx.ext.todo: support for todo items/lists
- sphinx.ext.coverage: collect doc coverage stats
- sphinx.ext.mathjax: Math support for HTML outputs in Sphinx
- sphinx.ext.viewcode: add links to highlighted source code
- sphinx.ext.napoleon: enables Sphinx to parse both NumPy and Google style docstrings

2.3. Further Reading: Deploying a Sphinx project online

See this link for more details. DO NOT deploy your course project materials online