DOCUMENTATION FOR SCIENTIFIC SOFTWARE

Alexander Moreno Briceño

UNIVERSIDAD ANTONIO NARIÑO

HSF HEP Software Foundation

S³ School — Sustainable Scientific Software School

"Good coding practices to develop better software for your research"

14–21 January 2026 at LAPP, Annecy, France

European Commission
LAPP
eosc
EVERSE
eosc
OSCARS

January 20, 2026

Objective of this lesson

- Understand why documentation is important for sustainable scientific software using Sphinx, building documentation from Python docstrings, Jupyter Notebooks, and publishing automatically using CI/CD

`https://www.sphinx-doc.org/en/master/`

Outline

1. Documentation in Scientific Software

Outline

1. Documentation in Scientific Software

2. Types of Documentation

Outline

1. Documentation in Scientific Software

2. Types of Documentation

3. Sphinx

Outline

1. Documentation in Scientific Software

2. Types of Documentation

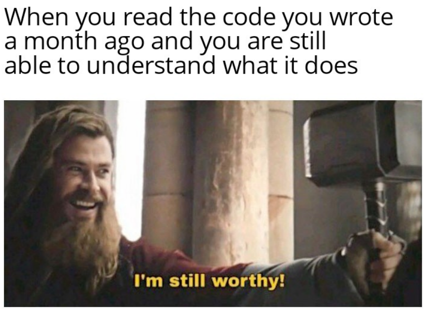
3. Sphinx

4. Hands-on Session

Documentation is important for Sustainable Scientific Software!

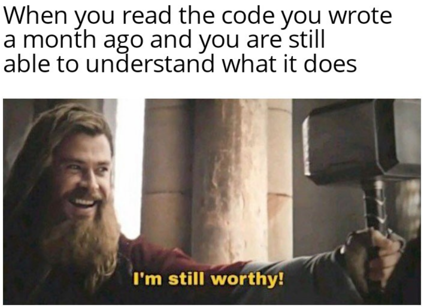
https://www.reddit.com/r/ProgrammerHumor/

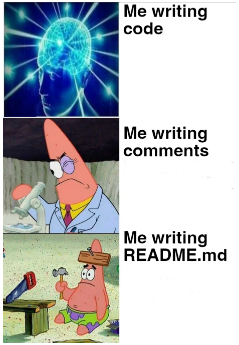
Documentation is important for Sustainable Scientific Software!


When you read the code you wrote a month ago and you are still able to understand what it does

https://www.reddit.com/r/ProgrammerHumor/

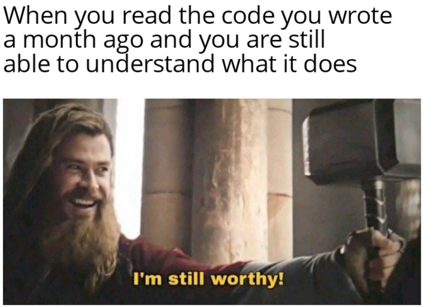
Documentation is important for Sustainable Scientific Software!

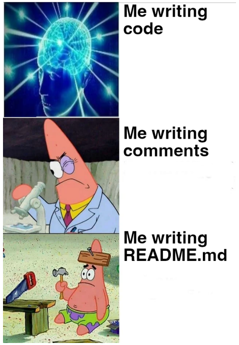

When you read the code you wrote a month ago and you are still able to understand what it does



https://www.reddit.com/r/ProgrammerHumor/

Documentation is important for Sustainable Scientific Software!

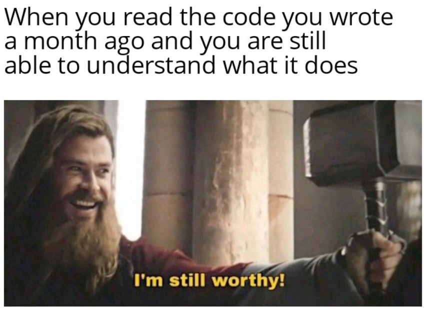

When you read the code you wrote a month ago and you are still able to understand what it does
I'm still worthy!

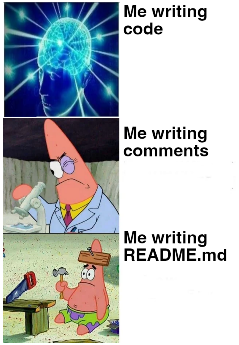


- Code without docs is not:

https://www.reddit.com/r/ProgrammerHumor/

Documentation is important for Sustainable Scientific Software!

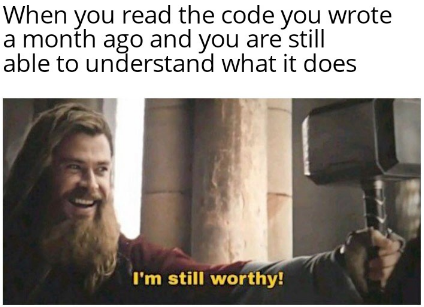

When you read the code you wrote a month ago and you are still able to understand what it does
I'm still worthy!

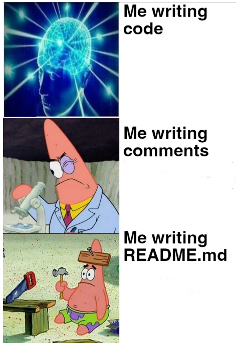


- Code without docs is not:
    - Reproducible

https://www.reddit.com/r/ProgrammerHumor/

Documentation is important for Sustainable Scientific Software!

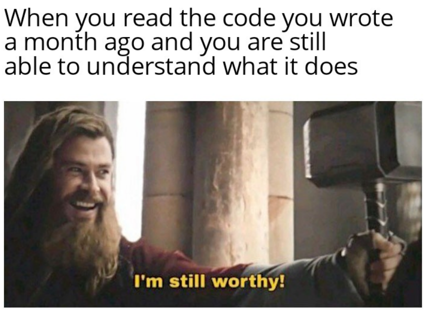

When you read the code you wrote a month ago and you are still able to understand what it does
I'm still worthy!

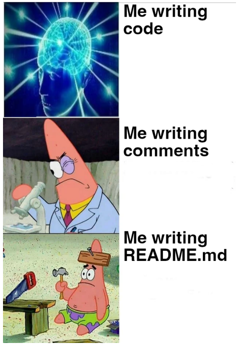


- Code without docs is not:
  - Reproducible
  - Reusable

`https://www.reddit.com/r/ProgrammerHumor/`

Documentation is important for Sustainable Scientific Software!


When you read the code you wrote a month ago and you are still able to understand what it does
I'm still worthy!



- Code without docs is not:
  - Reproducible
  - Reusable
  - Understandable

https://www.reddit.com/r/ProgrammerHumor/

Types of Documentation

Types of Documentation

- Code Documentation (The Reference!)
  - Developers, advanced users, and YOU!
  - APIs, function parameters, return types
  - Docstrings (Autogenerated)
  - Information-oriented ("What does this function do?")

Types of Documentation

- Code Documentation (The Reference!)
    - Developers, advanced users, and YOU!
    - APIs, function parameters, return types
    - Docstrings (Autogenerated)
    - Information-oriented ("What does this function do?")
- User Documentation (The Narrative!)
    - Users, and YOU!
    - Installation, workflows, theory
    - reStructuredText/Markdown
    - Understanding-oriented (""How do I simulate my data?)

Types of Documentation

- Code Documentation (The Reference!)
  - Developers, advanced users, and YOU!
  - APIs, function parameters, return types
  - Docstrings (Autogenerated)
  - Information-oriented ("What does this function do?")
- User Documentation (The Narrative!)
  - Users, and YOU!
  - Installation, workflows, theory
  - reStructuredText/Markdown
  - Understanding-oriented (""How do I simulate my data?)
- Tutorials (The Examples!)
  - Learners, and YOU!
  - Step-by-step lessons, results, visualizations
  - Jupyter Notebooks
  - Learning-oriented ("Show me a result in 2 minutes")

Sphinx in one slide

Sphinx in one slide

- Documentation compiler!

Sphinx in one slide

- Documentation compiler!
- Reads configuration, looks for docstrings in Python, parses .rst and .md files, and outputs a HTML/PDF

Sphinx in one slide

- Documentation compiler!
- Reads configuration, looks for docstrings in Python, parses .rst and .md files, and outputs a HTML/PDF
- It is the standard for scientific Python (NumPy, SciPy, Astropy)

Sphinx in one slide

- Documentation compiler!
- Reads configuration, looks for docstrings in Python, parses .rst and .md files, and outputs a HTML/PDF
- It is the standard for scientific Python (NumPy, SciPy, Astropy)
- Supports citations, cross-referencing, and math natively

Initializing with Pixi

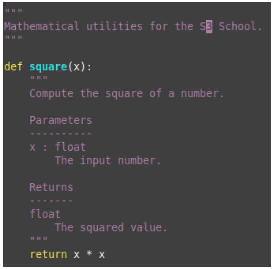
- Initialize a Pixi project (pixi init)
  - mkdir project
  - cd project
  - pixi init .
- Add dependencies (pixi add)
  - pixi add python sphinx sphinx_rtd_theme
- Setup Sphinx
  - sphinx-quickstart docs

Code and Configuration

- cd ..
- mkdir src
- vim calculations.py
- cd ..
- cd project/docs/source/
- vim conf.py
  - extensions = ['sphinx.ext.autodoc', 'sphinx.ext.napoleon', 'sphinx_rtd_theme']
  - html_theme = 'sphinx_rtd_theme'
- Add api.rst in /source
- Add api in index.rst

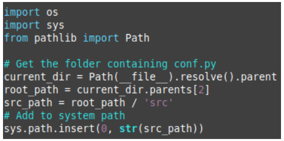
Code and Configuration

calculations.py



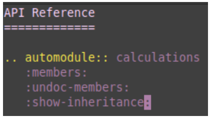
```python
"""
Mathematical utilities for the S3 School.
"""

def square(x):
    """
    Compute the square of a number.

    Parameters
    ----------
    x : float
        The input number.

    Returns
    -------
    float
        The squared value.
    """
    return x * x
```

conf.py



```python
import os
import sys
from pathlib import Path

# Get the folder containing conf.py
current_dir = Path(__file__).resolve().parent
root_path = current_dir.parents[2]
src_path = root_path / 'src'
# Add to system path
sys.path.insert(0, str(src_path))
```

api.rst



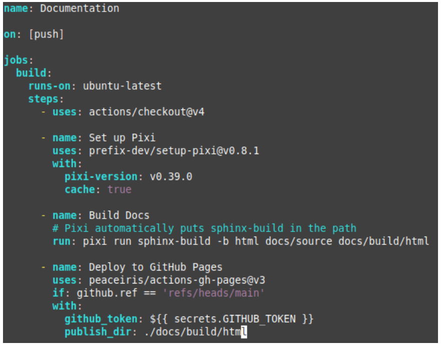
```rst
API Reference
=============

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

Notebooks

- pixi add nbsphinx ipykernel
- touch docs/source/tutorial.ipynb
- Add ’nbsphinx’ to extensions in conf.py
- Add tutorial to index.rst

CI/CD: The workflow file

- mkdir -p .github/workflows
- touch .github/workflows/docs.yml



```yaml
name: Documentation

on: [push]

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4

      - name: Set up Pixi
        uses: prefix-dev/setup-pixi@v0.8.1
        with:
          pixi-version: v0.39.0
          cache: true

      - name: Build Docs
        # Pixi automatically puts sphinx-build in the path
        run: pixi run sphinx-build -b html docs/source docs/build/html

      - name: Deploy to GitHub Pages
        uses: peaceiris/actions-gh-pages@v3
        if: github.ref == 'refs/heads/main'
        with:
          github_token: ${{ secrets.GITHUB_TOKEN }}
          publish_dir: ./docs/build/html
```

HAPPY CODING AND DOCS!