

User facing tutorials as code

reproducible and reliable tutorials with CI/CD

Brigitta Sipőcz

Caltech/IPAC – Scientific Python
BSSw and SSI Fellow

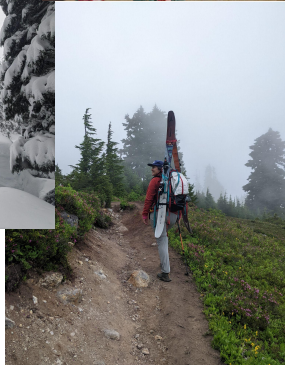
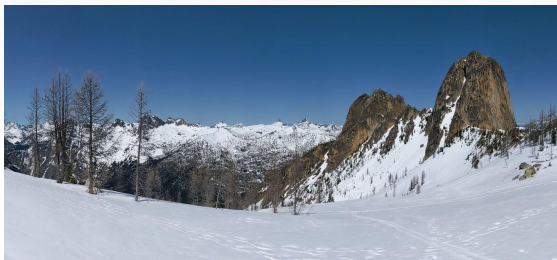


Caltech

Brigitta Sipőcz

astronomer → RSE, staff at Caltech/IPAC – NASA Infrared Science Archive

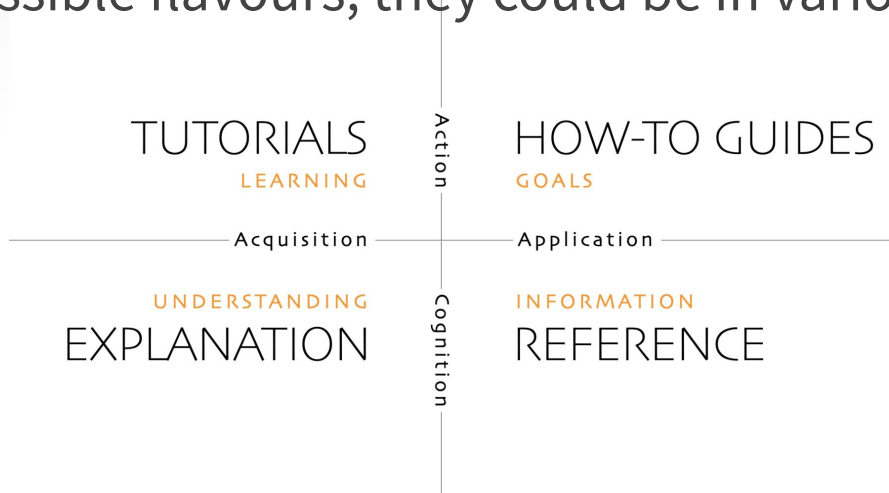
- Python open source maintainer
 - astroquery, numpy-tutorials, astropy, Scientific Python Project,...
- Organizing
 - Scientific Python Dev Summit, Scipy conference, hackweeks, GSoC
- Outdoors – Snow, mountains, cycling, alpine lakes, ...



User facing tutorials as code, reproducible and reliable

- Openly available documentation of a resource; library, API, data archive; including fully working code examples.

Multiple possible flavours, they could be in various [Diátaxis](#) categories



Flavours of tutorials

- Content for individualized/async learning

(Numpy Tutorials, NASA/IPAC Infrared Science Archive Tutorials)

- Long form narrative documentation and examples
- Expected to work in a wide range of user compute environments
- Users most often interact only with a fraction of the material
- Called differently in projects: Guides, How-tos, Tutorials, ...

- Workshop and course materials for synchronous learning

(Data Science Curriculum, Scikit-image Workshop)

- Full engagement during the workshop
- Specific environments often provided for the duration of the course
- May fall out of maintenance between workshops
- May require specific JupyterHub or installs

Flavours of tutorials

- Reproducible science use cases that are expected to work
(*NASA Astrophysics Science Platform demos*)
 - Tend to rely on more complex dependencies
 - Require more resources to run end-to-end
- Something else, e.g. paper appendix
 - Hopefully works at the time of publishing, but most often unmaintained afterwords

User facing tutorials as code, reproducible and reliable

- Code examples are working
- Results are reproducible
- Bonus: ease of maintainability and extendibility

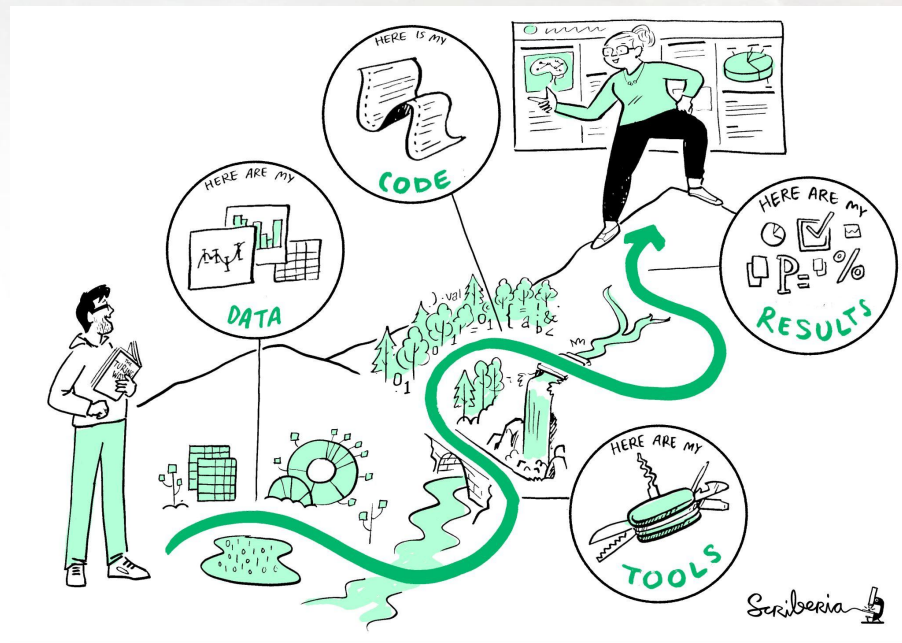


Image by The Turing Way community and Scriberia used under a CC-BY licence.

User facing tutorials as code, reproducible and reliable

CI/CD

- Code examples are working
- Results are reproducible
- Bonus: ease of maintainability and extendibility

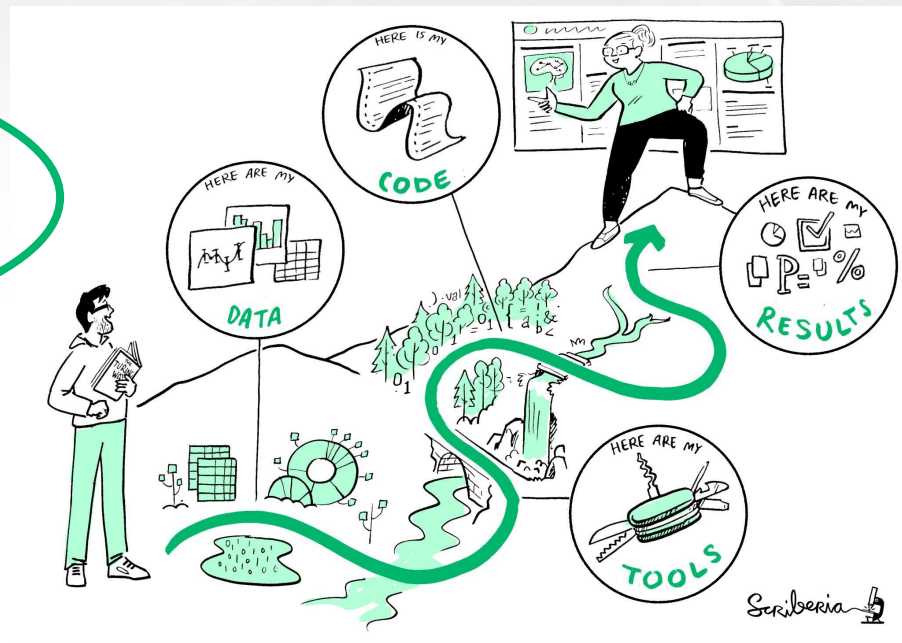


Image by The Turing Way community and Scriberia used under a CC-BY licence.

User facing tutorials as code, reproducible and reliable

CI/CD

- Code examples are working
- Results are reproducible
- Bonus: ease of maintainability and extendibility

**Myst Markdown Jupyter
Notebooks**

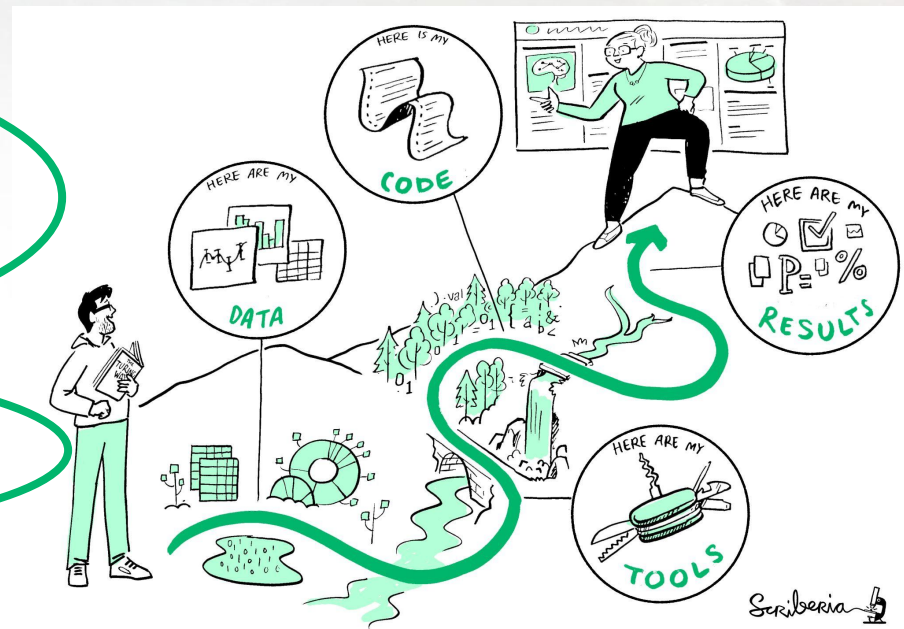


Image by The Turing Way community and Scriberia used under a CC-BY licence.

Continuous integration

- Running tests **automatically** for all proposed code changes.
- Goal: Catch problems as soon as possible.
- Test as widely as possible for the supported use cases
 - Operating systems
 - Supported version numbers
 - Development version of dependencies
 - Ecosystem wide integration tests



Image by The Turing Way community and Scriberia used under a CC-BY licence.

How we CI/CD

- Two main providers: **GitHub Actions** (GHA) and **CircleCI**
 - Other popular options for documentation/websites: **ReadTheDocs** and **Netlify**
- Build a CI Matrix, **same as we do for a library**
 - Cover supported versions (OS/architecture, dependencies, scenarios)
- Run CI matrix for all code changes and regularly from **cron**
 - Proactively looking for regressions, incompatibilities with new releases
- Typical limitations
 - Limited resources (GPU access, smaller memory, limited runtime, limited access to data)

Importance of Downstream testing

- Encourage testing all of
 - Libraries including documentation, use-case examples and notebooks
- Test as widely as possible for the supported use cases
 - OS, supported dependency versions and development versions
- Promote open communication
 - Teach how to write useful bug reports, encourage users to interact on GitHub and report bugs upstream
- Examples for e.g. PyVO downstream testing:
 - astroquery test suite
 - NAVO notebooks, IRSA tutorials, CDS notebooks

Jupyter and Jupyter Notebooks

- Modular, extensible, platform-agnostic ecosystem for interactive computing across programming languages
- Notebooks are an effective and efficient way to distribute tutorial content
- BUT Jupyter is more than JupyterHub and notebooks

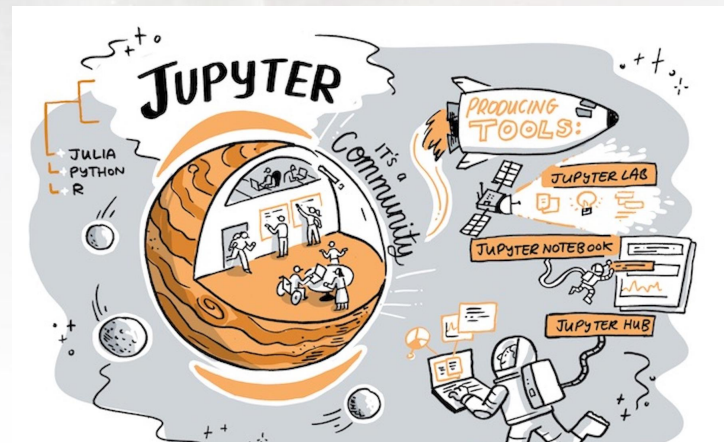


Image by The Turing Way community and Scriberia used under a CC-BY licence.

JupyterBook and MyST ecosystem

- Tools for technical communication and documentation from computational content
- JB2 uses the MyST Document Engine
 - Parses, transform, and render markdown/notebooks/etc into HTML/PDF/etc

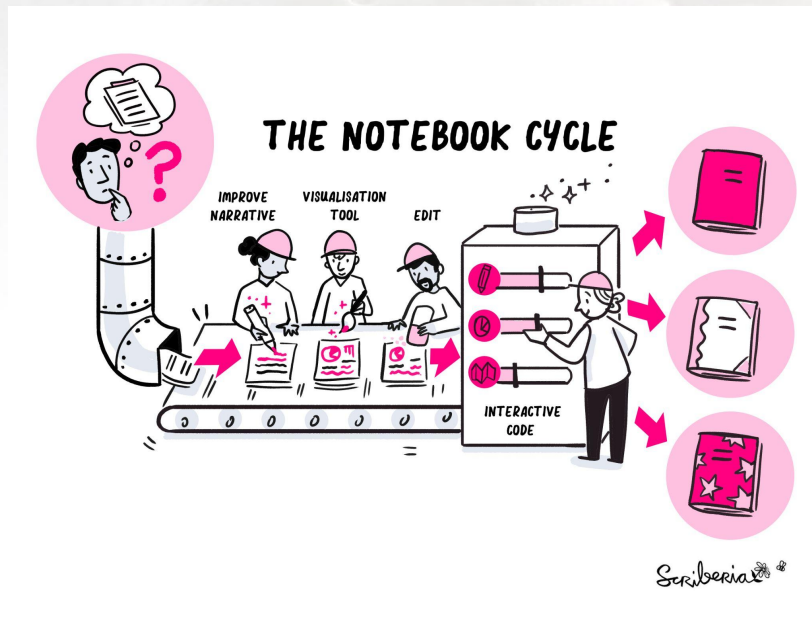


Image by The Turing Way community and Scriberia used under a CC-BY licence.

Jupyter Book users

- The Turing Way

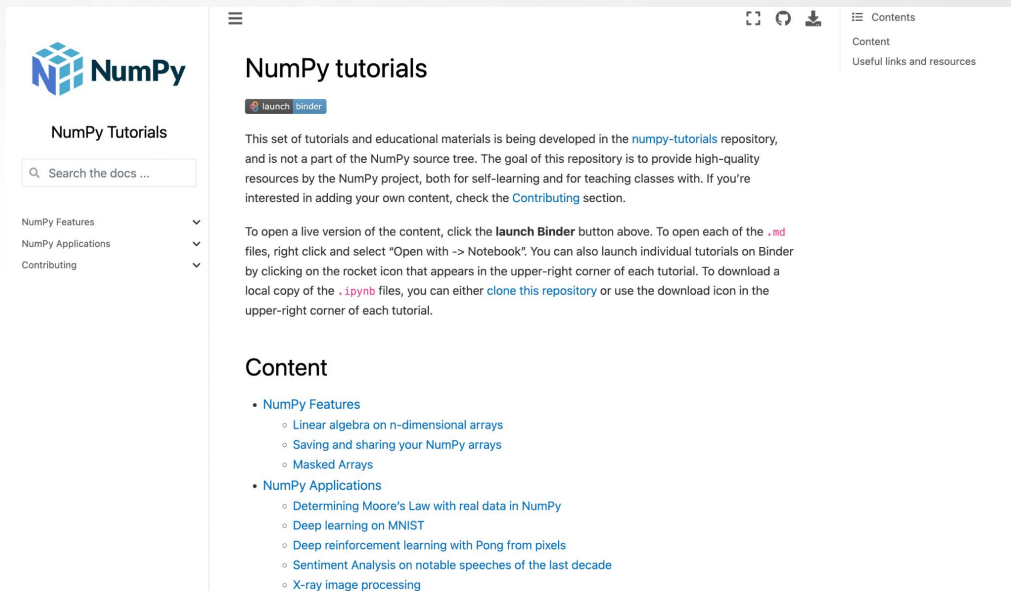
Handbook to reproducible, ethical and collaborative data science.

- Core scientific Python

- NumPy
- Scikit-learn
- NetworkX
- ...

- Astronomy resources

- Fornax and NAVO notebooks
- IRSA Tutorials
- CDS Tutorials
- ...



NumPy Tutorials

Search the docs ...

NumPy Features
NumPy Applications
Contributing

NumPy tutorials

[launch Binder](#)

This set of tutorials and educational materials is being developed in the [numpy-tutorials](#) repository, and is not a part of the NumPy source tree. The goal of this repository is to provide high-quality resources by the NumPy project, both for self-learning and for teaching classes with. If you're interested in adding your own content, check the [Contributing](#) section.

To open a live version of the content, click the **launch Binder** button above. To open each of the `.md` files, right click and select "Open with -> Notebook". You can also launch individual tutorials on Binder by clicking on the rocket icon that appears in the upper-right corner of each tutorial. To download a local copy of the `.ipynb` files, you can either [clone this repository](#) or use the download icon in the upper-right corner of each tutorial.

Content

- NumPy Features
 - Linear algebra on n-dimensional arrays
 - Saving and sharing your NumPy arrays
 - Masked Arrays
- NumPy Applications
 - Determining Moore's Law with real data in NumPy
 - Deep learning on MNIST
 - Deep reinforcement learning with Pong from pixels
 - Sentiment Analysis on notable speeches of the last decade
 - X-ray image processing

Text-based Notebooks

- Ease of editing and version control
 - Human readable, easy to edit
 - Cell outputs are NOT included
- Myst Markdown
 - A superset of CommonMark, developed for JB/Myst ecosystem
- Dependency: Jupyter package
 - supports various text formats → notebooks are both usable and editable within Jupyter and outside

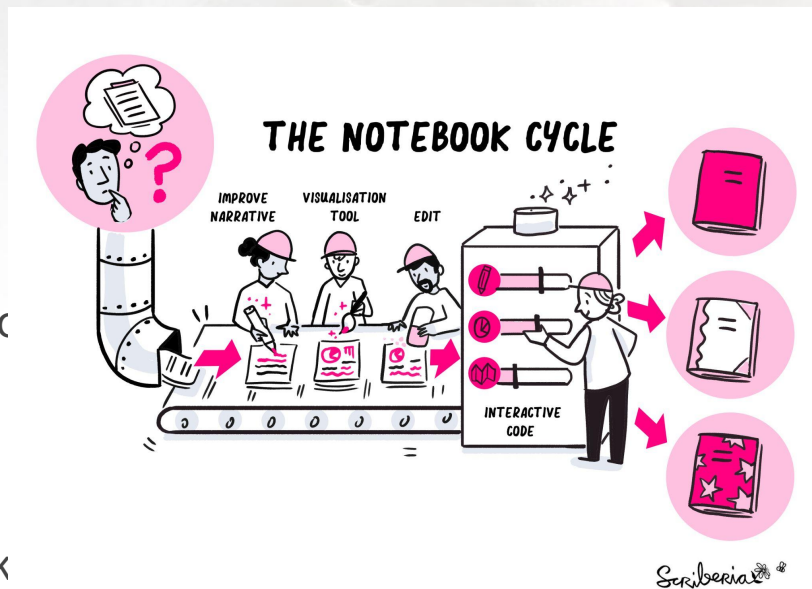


Image by The Turing Way community and Scriberia used under a CC-BY licence.

Text-based Notebooks

```
---
jupyter:
  formats: md:myst
  text_representation:
    extension: .md
    format_name: myst
    format_version: 0.13
    jupyter_text_version: 1.17.2
  kernelspec:
    display_name: Python 3 (ipykernel)
    language: python
    name: python3
  authors:
    - name: Jessica Krick
    - name: Vandana Desai
    - name: Andreas Faisst
    - name: Troy Raen
    - name: Brigitta Sipőcz
---

(spherex-intro)=
# Introduction to SPHEREx Spectral Images

+++
```

1. Learning Goals

- Query for SPHEREx Spectral Image Multi-Extension FITS files (MEFs) that overlap a given coordinate on the sky.
- Read in a SPHEREx Spectral Image MEF using Astropy and understand its structure.

Scientific Python Project and tutorials



- Porting the Scientific Python Lecture notes to JB
- Close collaboration with the Myst team
 - New features for website theme and API documentation
- SPEC recommendation document for tutorials
- An infrastructure repository to support/template tutorial maintenance
 - Builds on the Jupyter Book ecosystem
 - **Collects best practices and the infrastructure used by the big libraries** (numpy-tutorials, networkX-guides, etc)
 - use MyST markdown format to help version control and code review
 - automatize tests for notebooks (including regular cron jobs to test for new regressions)
 - automatize html rendering and deployment
 - workshop support with environment files
 - experiment with JupyterLite interactivity

Conclusions: tutorials as libraries

- **Avoid** reinventing infrastructure wheels: we actively collect best practices and resources to reuse
- Use tools in the ecosystem & contribute back
- Avoid overly restrictive requirements
- There is no one size fits all
 - Be explicit about aims and expectations
- Workflow & automation
 - PR based development
 - Testing
 - Deployment



DEMO slides

- A walked through contributor–maintainer experience at a Caltech/IPAC repository

IRSA Tutorials: <https://caltech-ipac.github.io/irsa-tutorials/>

IRSA Python tutorials

- Tutorial notebooks to access various NASA/IPAC Infrared Science Archive (IRSA) data and services using Python
- Hosted on github, open source but not community project
 - Follows the majority of community open source best practices
 - PR-based workflow, branch protections, CI/CD, open (public) issues and discussions
- Builds on the Jupyter Book ecosystem
 - **utilizes the infrastructure used by the big libraries** (numpy, networkX, etc)
 - use MyST markdown format to help version control and code review
 - automatize tests for notebooks (including testing with dev versions!)
 - regular cron jobs to test for new regressions (server side and dependencies)
 - automatize html rendering and deployment

IRSA

<https://caltech.edu/irsa/>

- Tutorial Archive

- Builds
 - util
 - use l
 - auto
 - regu
 - auto

irsa-tutorials Public

Unwatch

main 4 Branches 0 Tags

Go to file T Add file <>

DeekshaMohanty Merge pull request #365 from Caltech-IPAC/361-hidden-... b633852 · 15 hours ago 911 Commits

.binder	MAINT: don't hide requirements list in a hidden directory	7 months ago
.circleci	CI: circleci config cleanup	6 months ago
.github	Bump the actions group with 2 updates	2 weeks ago
_static	Move all cards to 1 grid, make header appear clickable	6 months ago
ignore_tutorials	Remove notebook from ignore list	2 weeks ago
plugins	Remove test from euclid and start a readme for plugins/ dir	4 months ago
tutorials	Fixing the hidden code and figure cells in the euclid q1 hat...	16 hours ago
.gitignore	Ignore numpy data files	5 months ago
.gitmodules	Add missing FSPS submodule for OpenUniverse SED note...	last month
.mailmap	Adding mailmap to remove name duplicates	6 months ago
LICENSE	MAINT: update LICENSE years	11 months ago
README.md	DOC: adding myst markdown to the readme	last year
deploy_to_fornax_manifest.in	Add ZTF light curves tutorial to notebook metadata and m...	last month
index.md	Remove generic author lists	6 months ago
jupytertext.toml	Adding jupytertext config to ensure metadata is not cleared o...	6 months ago
myst.yml	Add a notebook-gallery plugin	4 months ago

ance

workX, etc)

ew

rs!)

dependencies)

	main ▾	irsa-tutorials / tutorials / spherex /	Go to file T	Add file ▾	...
	bsipocz hiding all cells both in jupyterlab and renderring ✕	02df340 · last month History			
Name	Last commit message	Last commit date			
..					
spherex_source_discovery	Move author list into frontmatter (spherex sdt)	6 months ago			
spherex.md	Adding tutorial to gallery	3 months ago			
spherex_cutouts.md	Minor style fix to ensure we have a notebook triggered for testing	last month			
spherex_intro.md	Clean up SPHEREx notebook authors	5 months ago			
spherex_mosaic.md	hiding all cells both in jupyterlab and renderring	last month			
spherex_psf.md	small fixes: author, code-cell	5 months ago			

IRSA Python tutorials

The screenshot shows a GitHub pull request (PR) titled "Upgrade lsdb to >=0.8 #331" in the repository "Caltech-IPAC / irsa-tutorials". The PR is merged and was created by [troynaen](#) on July 1. It targets the `main` branch and is linked to the commit `jaladh-singhal:lsdb-upgrade`. The PR includes 16 conversations, 3 commits, 6 checks, and 3 files changed. The net change is +10 lines and -7 deletions.

A comment by [jaladh-singhal](#) on June 3 explains the choice of version 0.8.1, noting it was a hotfix for problems in 0.8.0 and is used in `foranx-demo-notebooks`. The comment links to the `astronomy-commons/lsdb@v0.8.1` release and indicates it closes issue #197.

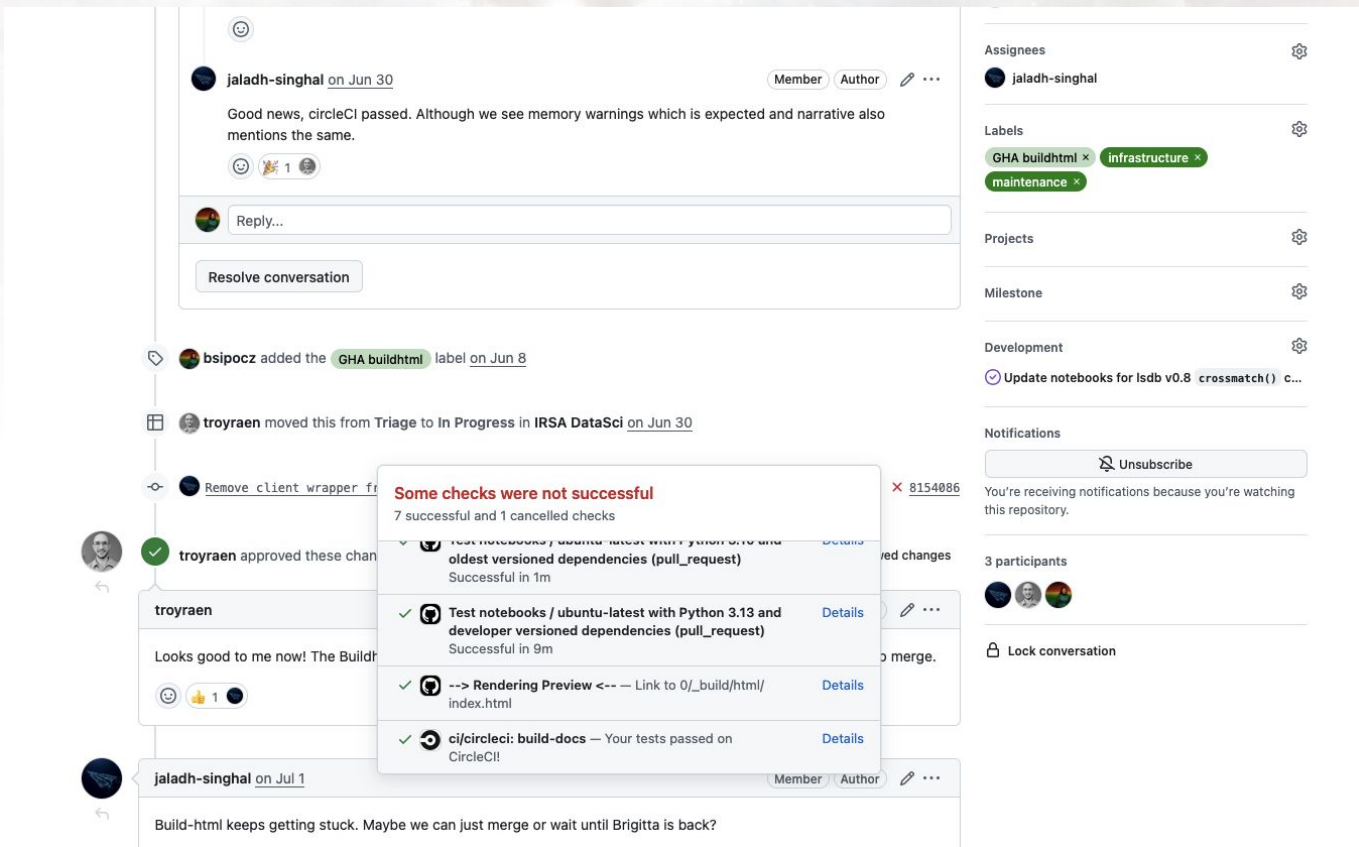
The PR history shows the following actions:

- `Upgrade lsdb to >=0.8.1` (commit 9c81b48) was added to the IRSA DataSci project on June 3 by `github-project-automation`.
- The PR was moved to the "Triage" state in the IRSA DataSci project on June 3 by `github-project-automation`.
- [jaladh-singhal](#) requested a review from [bsipocz](#) and [troynaen](#) two months ago.
- [jaladh-singhal](#) self-assigned the PR on June 3.
- [jaladh-singhal](#) added the `maintenance` and `infrastructure` labels on June 3.
- The PR was updated with the commit `Add suffix_method kwarg explicitly` (commit b2060e1).
- [jaladh-singhal](#) mentioned this PR on June 3.

The right sidebar shows the PR's metadata:

- Reviewers:** [bsipocz](#) (reviewed), [troynaen](#) (approved).
- Assignees:** [jaladh-singhal](#).
- Labels:** `GHA buildhtml`, `infrastructure`, `maintenance`.
- Projects:** IRSA DataSci.
- Milestone:** None.
- Development:** Update notebooks for lsdb v0.8 (crossmatch) c...
- Notifications:** Unsubscribe button. Message: "You're receiving notifications because you're watching this repository."
- Participants:** 3 participants ([bsipocz](#), [troynaen](#), [jaladh-singhal](#)).

IRSA Python tutorials



 **jaladh-singhal** on Jun 30 Member Author

Good news, circleCI passed. Although we see memory warnings which is expected and narrative also mentions the same.

  1 

Reply...

Resolve conversation

 **bsipocz** added the **GHA buildhtml** label on Jun 8

 **troyraen** moved this from Triage to In Progress in IRSA DataSci on Jun 30

 **Remove client wrapper from build-html**

 **troyraen** approved these changes

troyraen

Looks good to me now! The Build...

  1 

 **jaladh-singhal** on Jul 1 Member Author

Build-html keeps getting stuck. Maybe we can just merge or wait until Brigitta is back?

Some checks were not successful
7 successful and 1 cancelled checks

- ✓  **Test notebooks / ubuntu-latest with Python 3.13 and oldest versioned dependencies (pull_request)**
Successful in 1m [Details](#)
- ✓  **Test notebooks / ubuntu-latest with Python 3.13 and developer versioned dependencies (pull_request)**
Successful in 9m [Details](#)
- ✓  **--> Rendering Preview <-- -- Link to 0/_build/html/index.html** [Details](#)
- ✓  **ci/circleci: build-docs** — Your tests passed on CircleCI! [Details](#)

Assignees 

 **jaladh-singhal**

Labels 

GHA buildhtml  **infrastructure**  **maintenance** 

Projects 

Milestone 

Development 

 **Update notebooks for Isdb v0.8 crossmatch() c...**

Notifications

 **Unsubscribe**

You're receiving notifications because you're watching this repository.

3 participants

 **Lock conversation**

IRSA Python tutorials

Upgrade lsdb to >=0.8 #331

Code

Preview

Merged by troyraen on Jul 1 main ← jaladh-singhal:lsdb-upgrade

Conversation 16

Commits 3

Checks 6

Files changed 3

+10 -7

All commits

0 / 3 viewed

Submit comments

Settings

Info

2 Comments

Reactions

Filter files...

tutorials/techniques-and-tools

irsa-hats-with-lsdb.md

tox.ini

tutorial_requirements.txt

1 Comment

tutorials/techniques-and-tools/irsa-hats-with-lsdb.md

+7 -5

2 Comments

Reactions

Viewed

Comments

More

```
@@ -56,7 +56,7 @@ We will use lsdb to leverage HATS partitioning for performing fast spatial queri
56 56
57 57 ```{code-cell} ipython3
58 58 # Uncomment the next line to install dependencies if needed.
59 59 # !pip install s3fs "lsdb>=0.6.6,<0.8" pyarrow pandas numpy astropy dask matplotlib
60 60
61 61
62 62 ```{code-cell} ipython3
@@ -392,9 +392,10 @@ Since ZTF objects are defined per band, setting `n_neighbors=1` means this is on
392 392 ```{code-cell} ipython3
393 393 euclid_x_ztf = euclid_cone.crossmatch(
394 394     ztf_cone,
395 395     suffixes=("_euclid", "_ztf"), # to distinguish columns from the two catalogs
396 396     n_neighbors=1, # default is 1 too, can be tweaked
397 397     radius_arcsec=1, # default is 1 arcsec too, can be tweaked
398 398     suffixes=("_euclid", "_ztf"), # to distinguish columns from the two catalogs
399 399     suffix_method="all_columns",
400 400 )
401 401 euclid_x_ztf
```

IRSA Python tutorials

 **Merged**

Update firefly tutorials as per the latest firefly-client API #36

bsipocz merged 3 commits into [Caltech-IPAC:main](#) from [jaladh-singhal:update-firefly-tut](#) 2 weeks ago

  **bsipocz** approved these changes 2 weeks ago [View reviewed changes](#)

bsipocz left a comment Member ...

Given the comment above I go ahead and merge this now. Thanks for the review.



  **bsipocz** merged commit [f3e1bd9](#) into [Caltech-IPAC:main](#) 2 weeks ago Hide details Revert

8 checks passed

  ubuntu-latest with Python 3.12 [Details](#)

  macos-latest with Python 3.12 [Details](#)

  windows-latest with Python 3.12 [Details](#)

  ubuntu-latest with Python 3.9 and oldest versioned dependencies [Details](#)

  ubuntu-latest with Python 3.12 and developer versioned dependencies [Details](#)

  Buildhtml testing [Details](#)

  ci/circleci: build-docs Your tests passed on CircleCI! [Details](#)

  ci/circleci: build-docs artifact Link to [0/_build/html/index.html](#) [Details](#)

 **github-actions** bot pushed a commit that referenced this pull request 2 weeks ago

 Merge pull request [#36](#) from [jaladh-singhal/update-firefly-tut](#) ... ✗ 0fb2924

Conclusions: tutorials as libraries

- **Avoid** reinventing infrastructure wheels: we actively collect best practices and resources to reuse
- Use tools in the ecosystem & contribute back
- There is no one size fits all
 - Be explicit about aims and expectations
 - Avoid overly restrictive requirements
- Workflow & automation
 - Easy to edit source files in VC
 - PR based development
 - Testing
 - Deployment

