

Welcome to The Carpentries Etherpad!

This pad is synchronized as you type, so that everyone viewing this page sees the same text. This allows you to collaborate seamlessly on documents.

Use of this service is restricted to members of The Carpentries community; this is not for general purpose use (for that, try <https://etherpad.wikimedia.org>).

Users are expected to follow our code of conduct: https://docs.carpentries.org/topic_folders/policies/code-of-conduct.html

All content is publicly available under the Creative Commons Attribution License: <https://creativecommons.org/licenses/by/4.0/>

Links

Etherpad: <https://pad.carpentries.org/2023-ADACS-ECR-Workshop>

Workshop page: https://adacs-australia.github.io/2023_ASA_ECR_Python_Workshop/

Github for workshop: https://github.com/ADACS-Australia/2023_ASA_ECR_Python_Workshop

Webex link: <https://curtin.webex.com/curtin/j.php?MTID=m2eef0ab441aa81ed67e3908c45aaf7c4>

Pauls' email for feedback / questions: paul.hancock@curtin.edu.au

Pre-workshop survey: <https://forms.gle/cpoHF72b4Ah4k8co9>

Post-workshop survey: <https://forms.gle/HJR4ERAJRDEHZiFc6>

Sydney

Day 1

Packages:

Nick: numpy, matplotlib, pandas, astropy, psrqpy, pulsar_spectra (which I made muahahha)

Priyam: Astropy, matplotlib, scipy, mpdaf

Oguzhan: numpy, scipy, matplotlib, smplotlib, pandas

Rami: astropy, numpy, matplotlib, os, scipy, PSRCHIVE

Henry: numpy, scipy, astropy, extinction, multiprocessing, pathlib, tqdm, sys, os, pandas, matplotlib

Evans: numpy, pandas, matplotlib, astropy, os, scipy, warnings

Sam: numpy, matplotlib, tqdm

Srivardini - numpy, pandas, matplotlib, astropy, photutils, os

Claudia: cv2, scipy, pandas, sklearn

Daniel: numpy, matplotlib, astropy, scipy, photutils

Hina-Scipy, Astropy

Jahang- scipy, astropy

Joel: PSRCHIVE, scipy, numpy

Aashique: matplolib, numpy

Jaime A. Alvarado-Montes - scipy, spicepy, astroplan, ccdproc, astropy, BeautifulSoup

Mina Pak: numpy, astropy, scipy, matplotlib

Where to find out how to make and install python packages: <https://packaging.python.org/en/latest/>

Google *will* give you the wrong answer

Blog post by pip developer explaining the difference between setup.py and requirements.txt:

<https://web.archive.org/web/20130723114307/https://caremad.io/blog/setup-vs-requirements/>

To get the "locked" or "frozen" dependencies, you can do

...

```
pip list --format=freeze
```

...

Problems/questions:

Put any problems or questions here!

Your module README.md:

I'm waiting.....

<https://github.com/daniel-lyon/zfinder/blob/main/README.md>

Group projects

Jahang

Aashique Unnikrishnan

Sam Gordon

Priyam

https://github.com/samg525/constellation_idenitfier

Personal projects

Rami Mandow

Liroy Lourenço

Oguzhan Cakir

Mina Pak

Srivardini

Henry Zovaro (polishing my python package that I'm releasing soon:

<https://github.com/hzovaro/spaxelsleuth>)

Evans

Joel

Jaime A. Alvarado-Montes
Daniel
Hina

Discussions/study

Rami Mandow (general help with some packages I am stuck with)
Oguzhan Cakir -- quick discussion on "class"
Evans
Jaime A. Alvarado-Montes
Daniel

Other/
Claudia: Sorry, I have to leave at 4pm
Nick: All good Claudia

Version control question:

Rami - Overleaf with items that are no longer used % out. Also, Google docs for long docs when I get a feeling that my supervisor is going to get loads of comments on there. Overleaf overloaded with comments = triggering nightmare.
Henry - Pray that I have a Time Machine backup from when the good paragraph existed... (but I use git+github for all my papers irl)
Sam - Overleaf, comment out unused items or use history
Hina- Github version control/ Google docs versioning
Joel - Overleaf history/comment out unused items. Link Overleaf with dropbox and github.
Evans - Overleaf history to restore deleted items (figures) and % to comments paragraph that are considered unimportant
Oguzhan - Overleaf; commenting out stuff

Youtube video with a physical demo of the git repository: <https://www.youtube.com/watch?v=1ffBJ4sVUb4>

DAY 2

<https://swcarpentry.github.io/git-novice/index.html> is an introduction to git which goes at a slower pace, in case there are parts which are confusing you.

`git log --graph --decorate --oneline` is the command which draws the nice graph

The original git flow article: <https://nvie.com/posts/a-successful-git-branching-model/>

The "standard" GitHub workflow: <https://docs.github.com/en/get-started/quickstart/github-flow>

<https://jonas.github.io/tig/> - a terminal-based viewer for git

<https://meldmerge.org/> - a GUI to make the merges easier

<https://tox.wiki/> - a testing tool which handles installing your package, your test tools, and their dependencies

How to install conda:

<https://docs.conda.io/projects/conda/en/stable/user-guide/install/index.html>

Conda cheat sheet:

<https://docs.conda.io/projects/conda/en/stable/user-guide/cheatsheet.html>

Fill between (use a low alpha value):

https://matplotlib.org/stable/gallery/lines_bars_and_markers/lines_bars_and_markers_fill_between_demo.html#sphx-glr-gallery-lines-bars-and-markers-fill-between-demo-py

<https://datacarpentry.org/astronomy-python/> - covers a bunch of complementary topics to this workshop

Day 3

https://github.com/NickSwainston/asa_workshop

Example python package template which shows how to use sphinx and how to automatically set them up with github actions and pages

https://github.com/ADACS-Australia/python_project_template/tree/main

Some other documentation on how to make automated documentation with sphinx:

https://adacs-australia.github.io/2023-07-21_OzGrav_Python_Training/Docs/index.html

sky_sim options suggestions:

Output the version:

```
parser.add_argument('--version', action='version', version=f'%(prog)s  
{mymodule.__version__}')
```

Site for browsing all the sphinx themes: <https://sphinx-themes.org/>

ReadTheDocs: <https://readthedocs.org>

Jaime A. Alvarado-Montes: <https://github.com/JAAlvarado-Montes/ploonetide>

```
.. autoclass:: ploonetide.ploonetide.TidalSimulation
```

:members:

utils

.. automodule:: ploonetide.utils.functions

:members:

https://matplotlib.org/stable/tutorials/introductory/animation_tutorial.html

https://adacs-australia.github.io/research_project_management_training/03-trello/index.html

Day 4

<https://missing.csail.mit.edu/>

<https://tom.preston-werner.com/2009/05/19/the-git-parable.html>

<https://think-like-a-git.net/>

<https://mybinder.org/>

What will be slow:

Sam - The make_positions function

Oguzhan - clip_to_radius (due to for + if) and make_stars

Priyam: for _ in range(nsrc): Because of a large NSRC value and the slowest for loop

Henry: writing to the .csv file

Evans - I agreed with Oguzham and i think writing to CSV file.

Numbers you should know:

- <https://gist.github.com/jboner/2841832>
- <https://www.freecodecamp.org/news/must-know-numbers-for-every-computer-engineer/>
- https://colin-scott.github.io/personal_website/research/interactive_latency.html
- <https://softwareengineering.stackexchange.com/questions/312485/how-can-jeff-deans-latency-numbers-every-programmer-should-know-be-accurate-i>
- <https://nitter.net/skimbrel/status/893337218101555200#m>

Profilers:

- profile/cProfile: <https://docs.python.org/3/library/profile.html> - part of the standard library, should work on all your systems. There is some discussion on the different types of profilers at <https://docs.python.org/3/library/profile.html?highlight=profile#what-is-deterministic-profiling>.
- py-spy: <https://github.com/benfred/py-spy> - James' preferred profiler, can "attach" to an existing process, making it easy to see why an already-running process is slow

SQL/ADQL:

- TOPCAT: <https://www.star.bris.ac.uk/~mbt/topcat/>
- ADQL introduction from ESA: <https://www.cosmos.esa.int/web/gaia-users/archive/writing-queries>
- ADQL tutorial from CDS (who run Vizier, Simbad, Aladin): <https://simbad.u-strasbg.fr/simbad/tap/help/adqlHelp.html>
- Video tutorial from the IVOA Education group: <https://www.youtube.com/watch?v=jHKcGtpyeE8>
- The Data Aggregation Service: <https://das.datacentral.org.au>

Day 5

Nick's example of how to make a DF:

https://github.com/NickSwainston/all_pulsar_spectra/blob/main/fit_all_pulsars.py

Tidy Data:

An R-based intro to tidy data: <https://tidyr.tidyverse.org/articles/tidy-data.html>

The original paper: <https://vita.had.co.nz/papers/tidy-data.html>

Other ADACS workshops that may be useful:

<https://adacs.org.au/adacs-training-vision/workshops/>

And some more courses (mostly video based):

<https://adacs.org.au/adacs-training-vision/lms-courses/>

Please do the post-survey before you leave:

<https://forms.gle/HJR4ERAJRDEHZiFc6>