Temps et coordonnées astronomiques avec ***astropy***

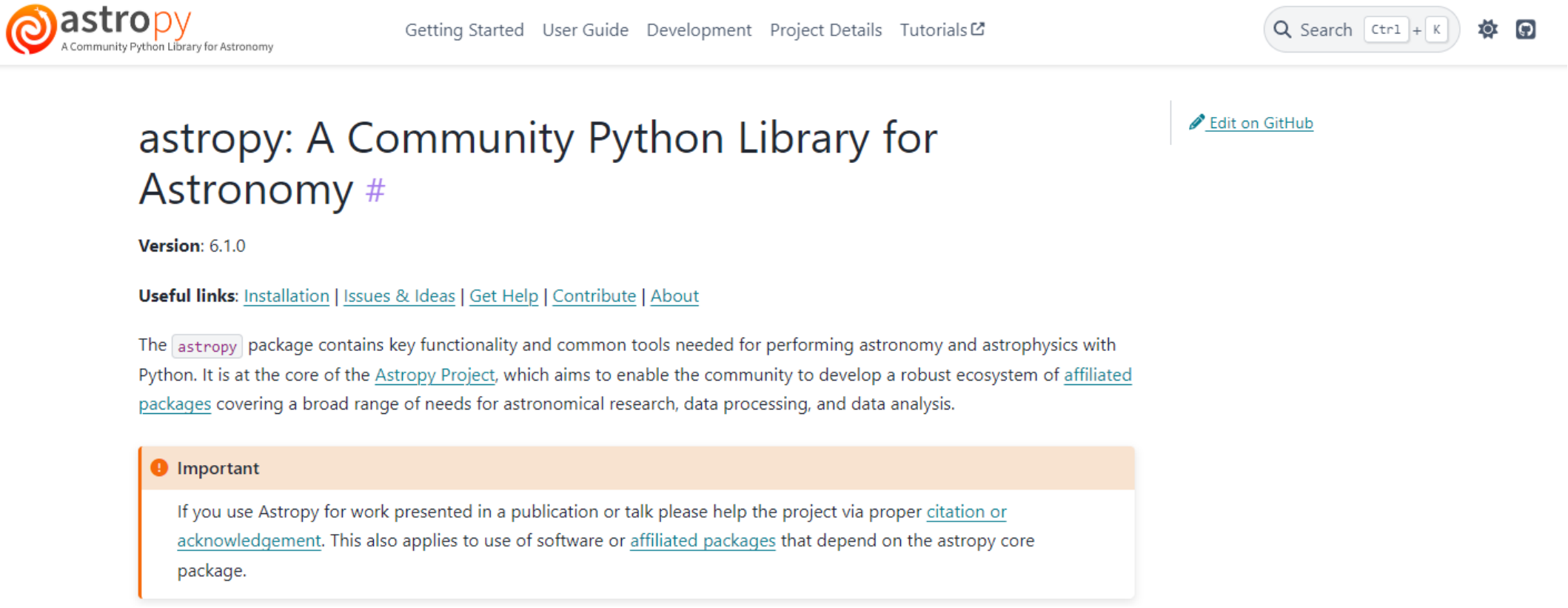
(et sans doute un peu plus...)

Jean-Baptiste Marquette
SAF - SF2A

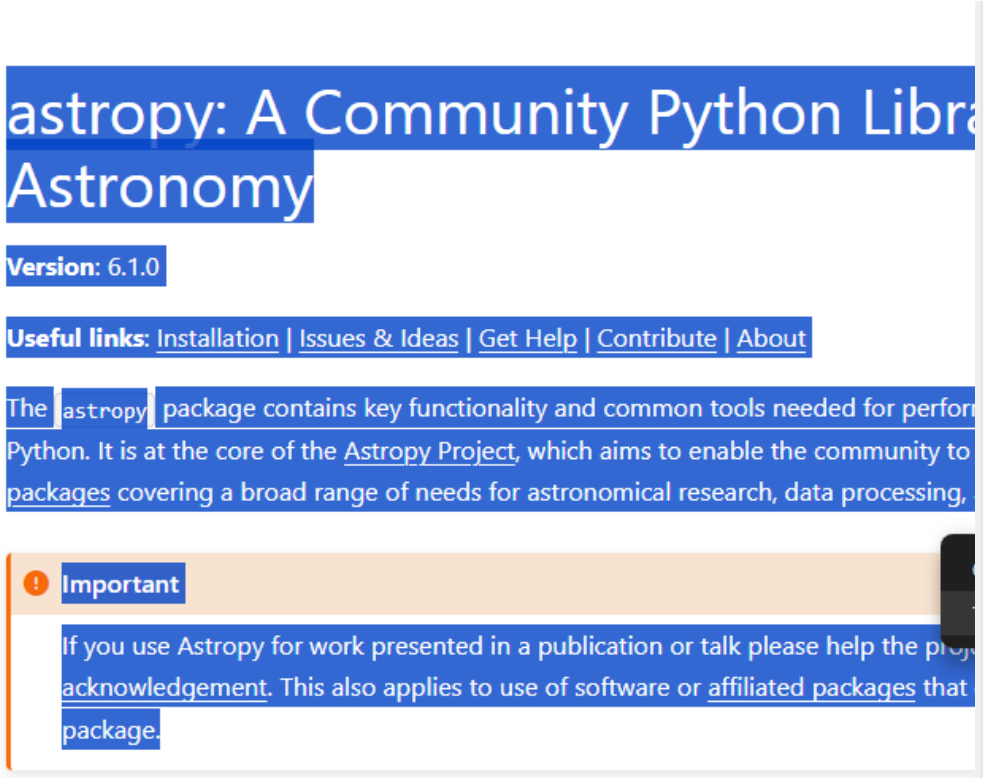
astropy, qu'est-ce que c'est ?

- Langage Python
- Des bibliothèques nombreuses
- Très bien documenté
- Devenu un standard
- Donc très utilisé
- Donc robuste !

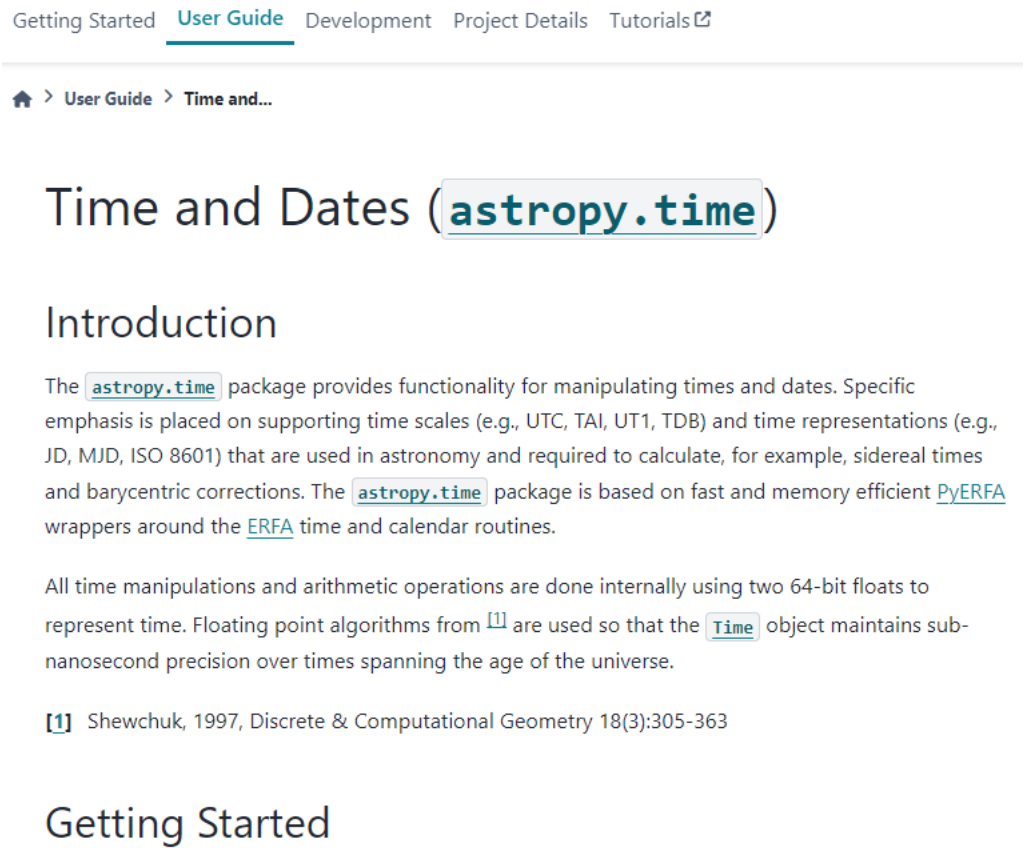
Le site https://docs.astropy.org/en/stable/index.html



Pour les anglophobes… euh, non-anglophones...



Pour aujourd'hui (1/2) : les temps en astronomie
https://docs.astropy.org/en/stable/time/index.html



Getting Started | User Guide | Development | Project Details | Tutorials

User Guide > Time and...

Time and Dates (`astropy.time`)

Introduction

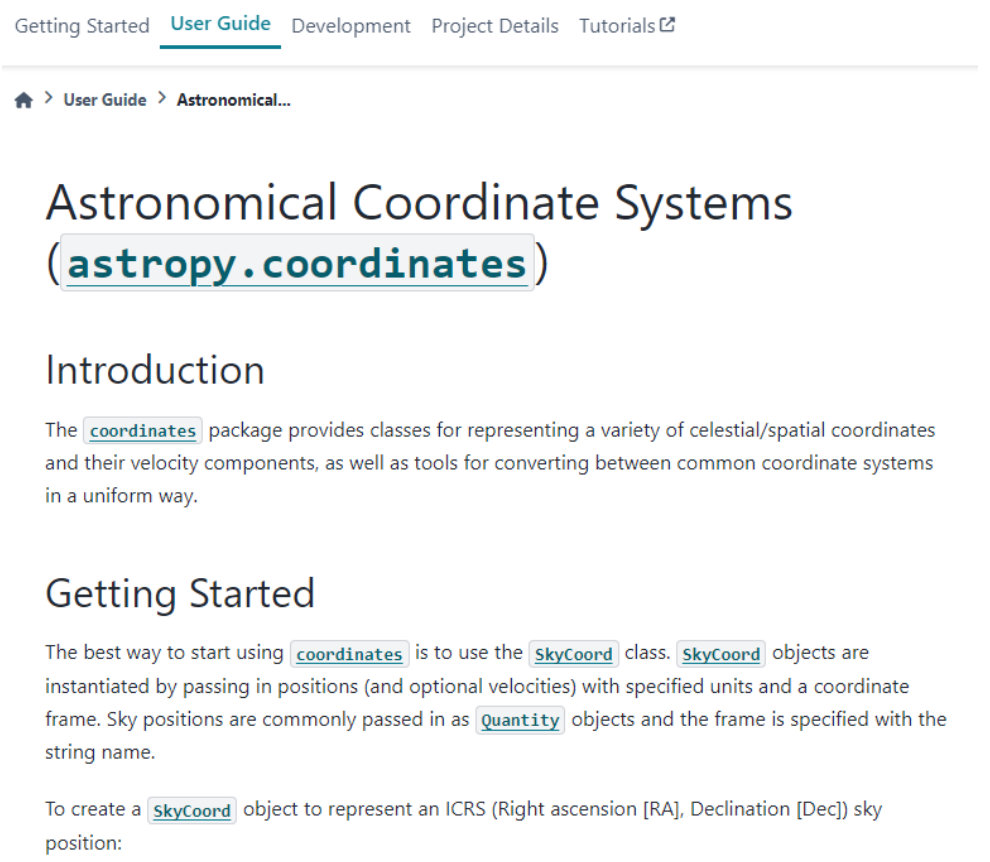
The `astropy.time` package provides functionality for manipulating times and dates. Specific emphasis is placed on supporting time scales (e.g., UTC, TAI, UT1, TDB) and time representations (e.g., JD, MJD, ISO 8601) that are used in astronomy and required to calculate, for example, sidereal times and barycentric corrections. The `astropy.time` package is based on fast and memory efficient PyERFA wrappers around the ERFA time and calendar routines.

All time manipulations and arithmetic operations are done internally using two 64-bit floats to represent time. Floating point algorithms from [1] are used so that the `Time` object maintains sub-nanosecond precision over times spanning the age of the universe.

[1] Shewchuk, 1997, Discrete & Computational Geometry 18(3):305-363

Getting Started

Pour aujourd'hui (2/2) : les coordonnées en astronomie https://docs.astropy.org/en/stable/coordinates/index.html



Getting Started | User Guide | Development | Project Details | Tutorials

User Guide > Astronomical...

Astronomical Coordinate Systems (`astropy.coordinates`)

Introduction

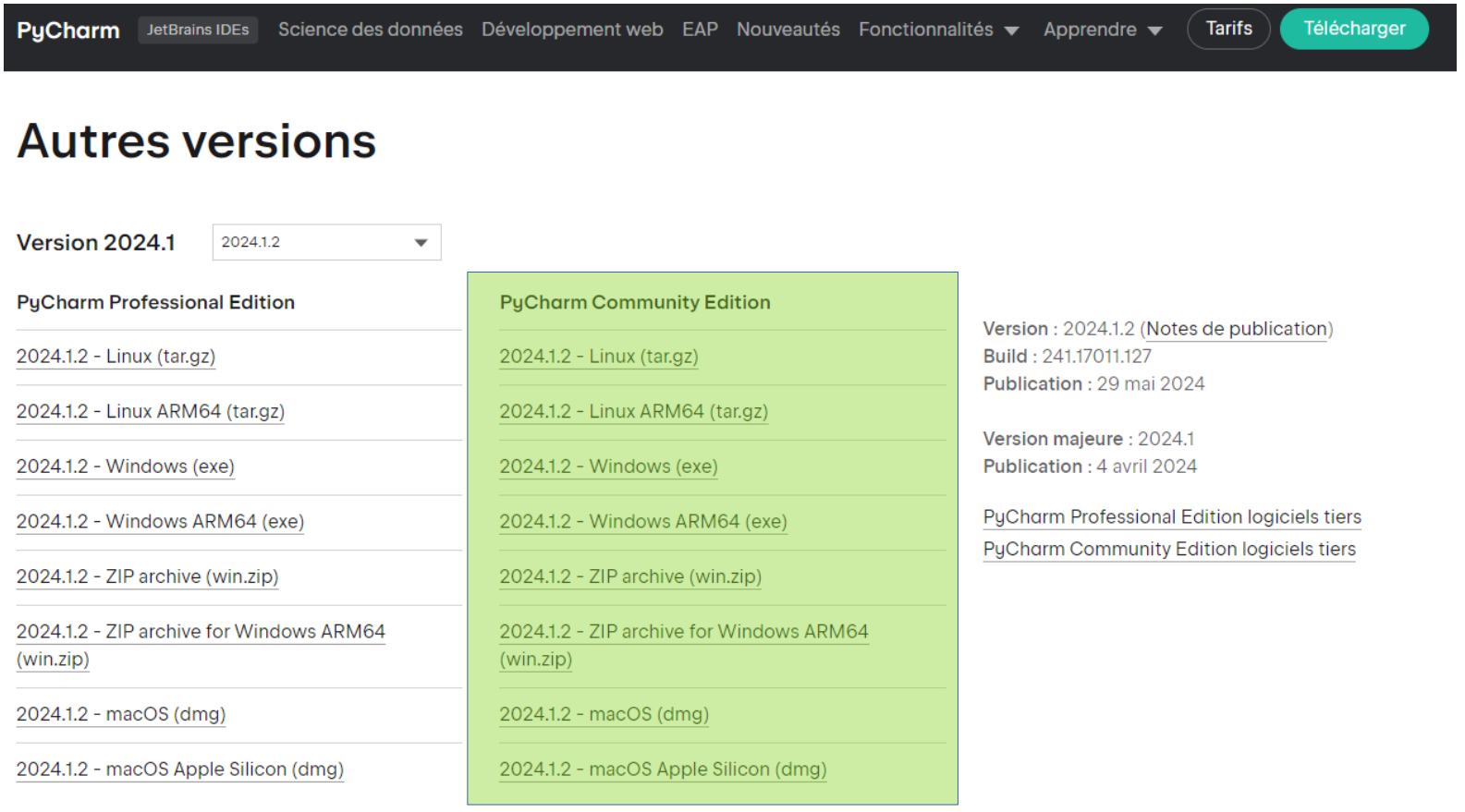
The `coordinates` package provides classes for representing a variety of celestial/spatial coordinates and their velocity components, as well as tools for converting between common coordinate systems in a uniform way.

Getting Started

The best way to start using `coordinates` is to use the `SkyCoord` class. `SkyCoord` objects are instantiated by passing in positions (and optional velocities) with specified units and a coordinate frame. Sky positions are commonly passed in as `Quantity` objects and the frame is specified with the string name.

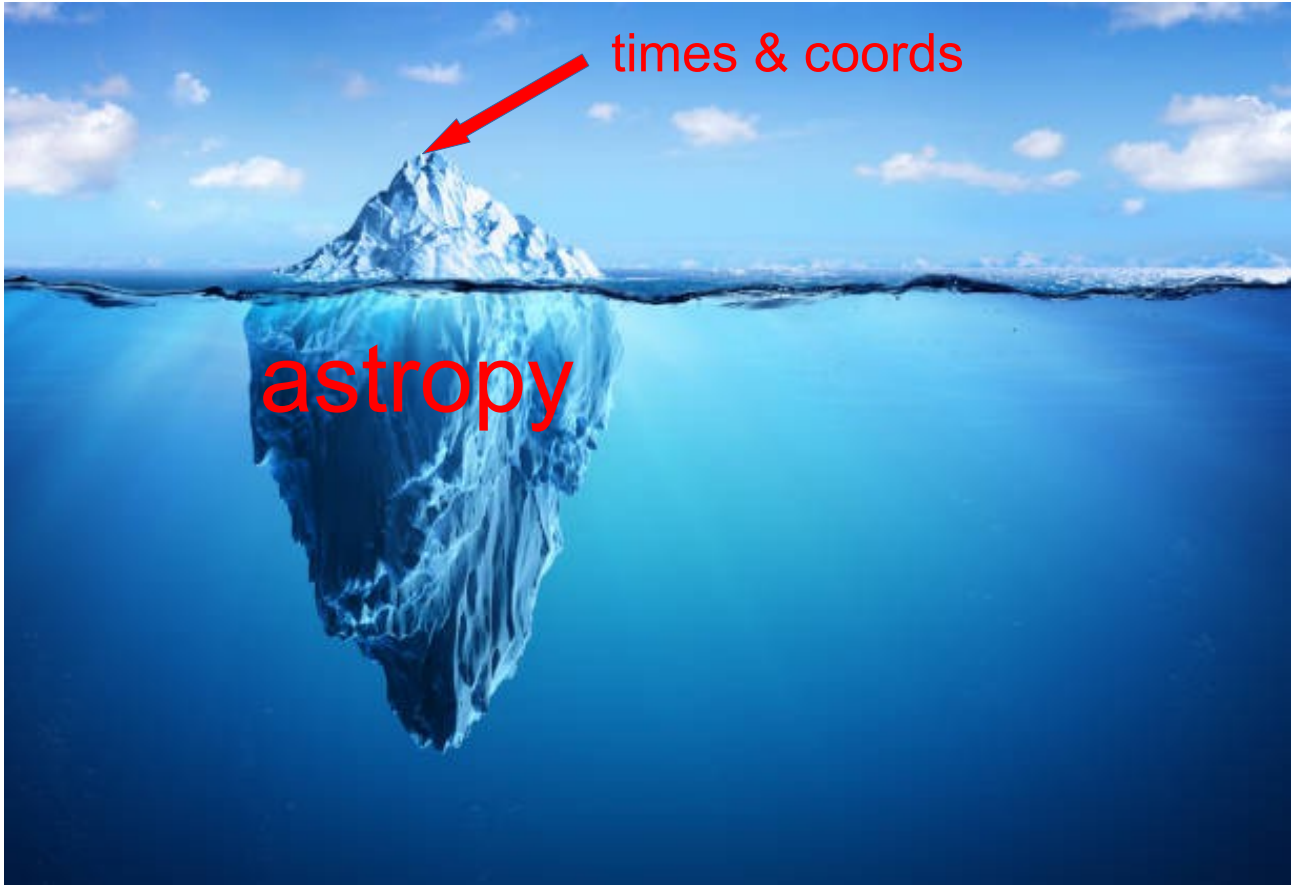
To create a `SkyCoord` object to represent an ICRS (Right ascension [RA], Declination [Dec]) sky position:

Un peu du “*un peu plus*”…
https://www.jetbrains.com/fr-fr/pycharm/download/other.html

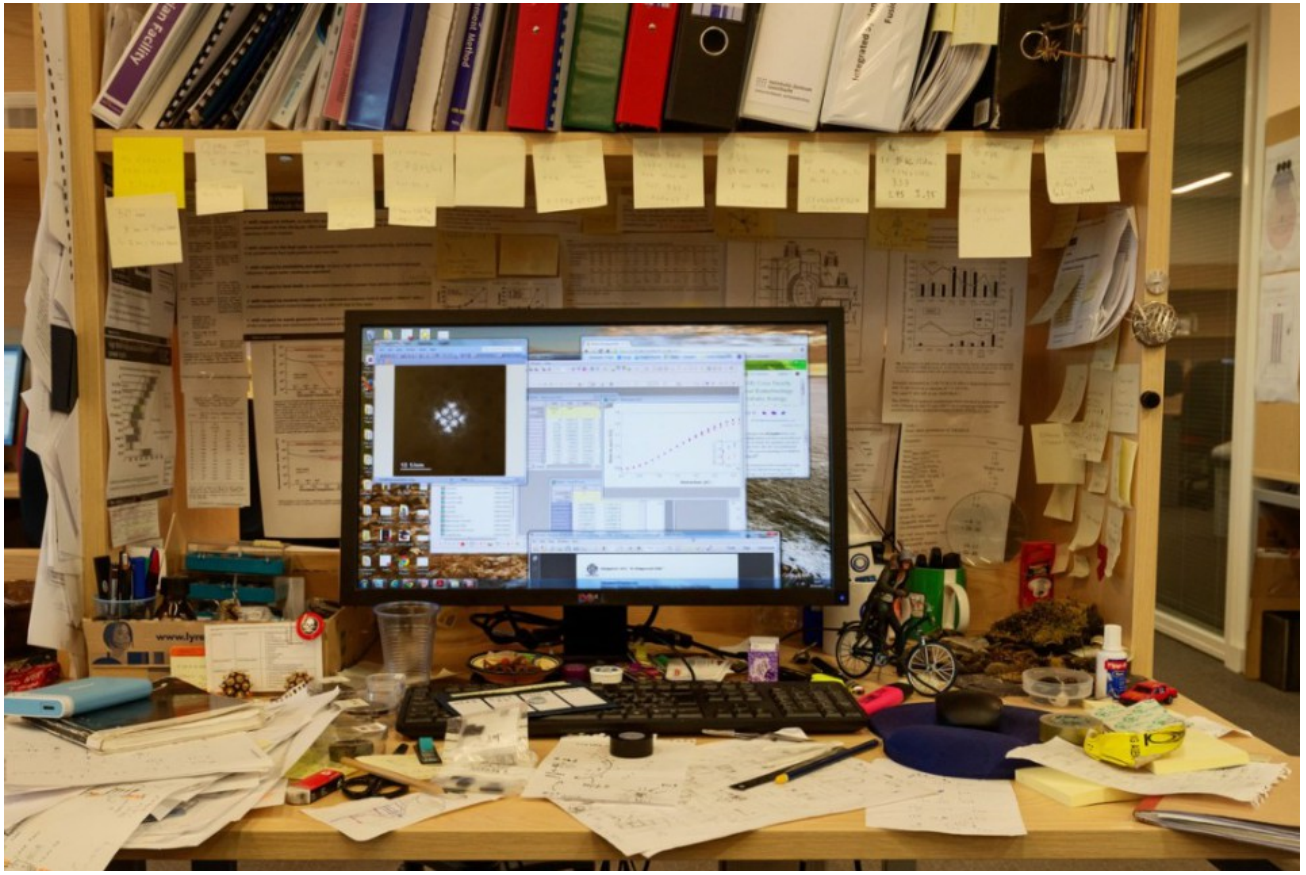



File Edit View Navigate Code Refactor Run Tools VCS Window Help
EcoleMarseille2024
Current File
Project
EcoleMarseille2024 C:\Users\astro\
External Libraries
Scratches and Consoles
AIE CONFIANCE
dans
personne ne vous entend crier.
Python Console
Install AI Assistant
Notifications
Bookmarks
Structure
Version Control
Python Packages
TODO
Coverage
Microsoft Defender configuration: Project paths were su
C:\Users\astro\anaconda3

Juste un minuscule bout de la pointe...



Démo interactive ! Avec ordre et clarté...



On continue ?...

Coordonnées ↔ WCS

Getting Started | User Guide | Development | Project Details | Tutorials

User Guide > World...

World Coordinate System (`astropy.wcs`)

Introduction

World Coordinate Systems (WCSs) describe the geometric transformations between one set of coordinates and another. A common application is to map the pixels in an image onto the celestial sphere. Another common application is to map pixels to wavelength in a spectrum.

`astropy.wcs` contains utilities for managing World Coordinate System (WCS) transformations defined in several elaborate FITS WCS standard conventions. These transformations work both forward (from pixel to world) and backward (from world to pixel).

Nouvelle démo interactive !
Avec calme et sérénité...


C'EST TOUT
POUR AUJOUR
D'HUI


QUESTIONS