

Proposal: Creating a central entry point for all HEP Software Training Material

One place for students to find study material, similar to <https://learn.astropy.org>

Contact: Kilian Lieret (kilian.lieret@posteo.de)

Overview.....	1
Pitch GSoC.....	2
Requirements.....	3
Tiles.....	4
Side panels.....	5
Technical details.....	5

Overview

- **Problem to be solved:**
 - **Students:**
 - Feel overwhelmed because of the vast amount of prerequisite software knowledge
 - Not easy to find good training material of the appropriate level and context
 - **Package maintainers:**
 - Want people to find their documentation, and understand that their packages are used and supported
 - It has been brought up before that the scikit-hep/coffea cosmos only shows small demos for singular packages, but that it is hard for students to see the “big picture”
 - **Advisors:**
 - Want to be able to quickly point students to a list of prerequisites rather than explaining everything themselves
- **Status quo:**
 - At HSF Training, we have created a [curriculum](#). **However,**
 - Different students have different needs ⇒ The concept of a “linear” curriculum isn’t really applicable, would need different “tracks”
 - Even with only ~15 training modules, the curriculum starts to become crowded ⇒ Very small selection of material with a narrow focus on a specific *kind* of teaching style
- **Solution:**
 - Create a **single entry point** with options to filter training material by various criteria
 - See below for details

- The HSF curriculum and other pages become obsolete
- **Inspirations:**
 - Learn.astropy.org does this for the astropy community
- **Existing frameworks to leverage:** Can fork learn.astropy.org, greatly reducing development effort
- **Proposed Domain:** learn.hepsoftwarefoundation.org
 - But maybe we could get a shorter domain? This is really clunky...
- **Work time estimate:**
 - ~1 week work for someone already somewhat familiar with JS/TS
 - Could also be project for GSoC or for IRIS-HEP fellow

Pitch GSoC

Project Name

A New Training Center for HSF Training

Description

There is a large number of training resources for newcomers in the field of High Energy Physics. The [HSF Training group](#) has started to compile a curriculum of such training modules that helps to get beginners up to speed quickly. However, the [current listing in the form of a static table](#) is quickly becoming overwhelming, and we cannot include many resources because of space limitations.

This project is about creating a new Training Center that turns the static page into a dynamic list of training content that can be filtered by attributes such as programming language, common tasks, type of training, HEP experiment, etc. The candidate can start with the code of <https://learn.astropy.org/> (an unrelated training website of a popular astrophysics package) and change the underlying data sources: rather than being based on a series of Jupyter notebooks, configuration files should allow listing arbitrary content.

Requirements

Typescript, HTML, web development basics

Task ideas

- Run a local version of learn.astropy.org
- Adapt the data sources to list arbitrary content
- Adapt the design to that of HSF
- Help to get a first version of the website live
- Add custom sorting algorithms
- Work on integrating a fulltext search engine
- Collect user feedback

- Add code to collect the number of visitors and searches

Expected results

A deployed training center webpage with basic filters and a full text search.

Mentors

TBD

Additional information

- Difficulty level (low/high/medium): low
- Estimated effort: medium (175 hours).
- Mentor availability: TBD

Corresponding Project

HSF Training

Participating Organizations

TBD

Requirements

Learn Astropy

Learn how to use Python for astronomy through tutorials and guides that cover Astropy and other packages in the astronomy Python ecosystem.

Format

- ☐ tutorial (22)
- ☐ guide (1)

Astropy packages

Search here...

- ☐ coordinates (10)
- ☐ astroquery (9)
- ☐ units (8)
- ☐ modeling (4)
- ☐ wcs (4)
- ☐ spectral cube (3)
- ☐ table (3)
- ☐ synphot (2)
- ☐ constants (1)
- ☐ convolution (1)

Show more

Python packages

Search here...

- ☐ matplotlib (14)
- ☐ numpy (2)
- ☐ scipy (1)

Tasks

Search here...

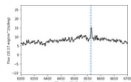
Search by  algolia

GUIDE CCD Data Reduction Guide



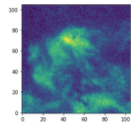
The purpose of this text is to walk through image reduction and photometry using Python, especially Astropy and its affiliated packages. It assumes some basic familiarity with astronomical images and with Python. The inspiration for this work is a pair of guides written for IRAF, "A User's Guide to CCD Reductions with IRAF" (Massey 1997) and "A User's Guide to Stellar CCD Photometry with IRAF" (Massey and Davis 1992).

TUTORIAL Modeling 2: Create a User Defined Model using astropy.modeling



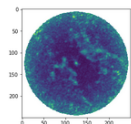
In this tutorial, we will learn how to define a new model in two ways: with a compound model and with a custom model.

TUTORIAL Extracting and Plotting Position-Velocity Diagrams



In this tutorial, we will extract position-velocity (PV) diagrams from a spectral cube and plot them.

TUTORIAL Cube Reprojection Tutorial



Spectroscopic cube observations taken at different wavelength can trace the motion of gas or stars using spectral lines, but often lines at different wavelengths give different information. For example, one might observe a galaxy in the 21cm line of HI and the 115 GHz line of CO, or a protoplanetary disk in a line of N2H+ and a line of CO, or a galactic disk in the H-alpha and H-beta lines (in absorption or emission). In order to compare these data sets pixel-by-pixel, they must be placed onto a common grid with common resolution.

We keep a similar layout to learn.astropy.org but customize various things.

Tiles

- In contrast to learn.astropy.org, tiles should be configured manually, not always correspond to a notebook.
- The following attributes need to be configured
 - Target: This is what happens when you click the tile
 - Title
 - Description
 - Image
 - All the information for the side panels above
 - Prerequisites: Should link to other tiles
- Ideally, it should also be easy to add the tiles to other pages with a simple include statement?

Side panels

- Type:
 - Tutorial
 - Documentation
- Level:
 - Beginner
 - Advanced
- Language:
 - C++
 - Python
 - Julia
- Package: Listing packages of the language
- Tasks: Physics tasks (e.g., histogramming, visualization, data access,)
- Experiment:
 - Prerequisite (software prerequisites, entirely independent of HEP, e.g., git, python, bash)
 - Cross experiment
 - ATLAS
 - CMS
 - Belle
 - ...
- Status:
 - Ready (selected by default)
 - Beta (selected by default)
 - In development (deselected by default)
- Curriculum: (this is for grouping some related modules together)
 - Different “tracks” for different experiments?
 - Maybe this is less important for students to select, but if we can apply the selector in the URL, so that e.g., ATLAS supervisors send people to central.entry.point/atlas-analysis and find a nice selection there, this would be cool

Technical details