ESOpy 2.0
8-9-10 January 2018
ESO Vitacura

LOC: R. Thomas, B. Dias, A. Razza, E. Sedaghati

Projects description for afternoon activities

The two afternoon sessions at second and third days will be dedicated to some projects. The plan is that each of you work on one project, or if you wish to finish a project in one afternoon, you may move to a second project.

We will split everyone into six groups following your preferences. Please follow the link below to give a priority from 1 to 6 to each of the six projects, where 1 is the top priority. We would like to have your answer before the starting of the event, therefore we appreciate if you **answer by Sunday, Jan 7**:

https://goo.gl/forms/EZkyYqY2xv7jrEax2

Please find below the details of each of the six projects offered.

Project 1 - coordinated by Romain Thomas

Title:
Where do we all look at? A real time overview of the ESO telescopes during the night

Abstract:
This project aims at providing a real time position map of all the ESO Telescope during the night. Trainees will be creating a GUI in python that allows to plot at a single moment all the ESO telescope position over the sky.

It will make you practice some interesting features in Python:

-Create a simple GUI with embedded plots and interaction between plot and GUI
-Retrieve real time OB information from all the Telescope (coordinates, OB time...)
-Plot their trajectories in a 3D half sphere representing the sky over the observatories

Required background, skills:
None

Required python packages:
- python 3.5+ and the ususal (numpy, scipy, etc…)
- matplotlib
- PyQt5
- astropy

Project 2 - coordinated by Romain Thomas

Title:
Paranal Error Database

Abstract:
In Paranal we often have errors on the telescops and instruments. The error messages are in general obscure for sciops department and the team staff might take some time to retrieve the good PPRS or the troubleshooting operation that allows to go back to the sky.
The idea here is to create an error database that would link error message and troubleshooting and would increase the efficiency of the team crew.

The principle would be to be able to fetch the error message from the log monitor into a program that display information about that error
-# of occurences,
-last occurence,
-associated PPRSs
-Troubleshooting
If the error has never happened before, a new entry in the database would be created.
The ideal would be to make a GUI (optionnal for the project) and one particular DB for each instrument and a common for the UTs.

It will make the trainee to work with:
-Reading from file
-Create database and reading from it with queries
-Create a GUI

Required background, skills:
None to expert

Required python packages:
- Python 3.5+
- sqlite3
- PyQt5 (if GUI)

Project 3 - coordinated by Frédéric Vogt

Title:
Finding charts 101

Abstract:
Finding charts are not only a useful device for planning observations - they are also the perfect excuse to discover several nifty features of Python. In this project, I propose to write from scratch a code to create finding charts automatically, and in so-doing explore (based on the interest of each participant) some of the following elements:

- FITS+ WCS plotting with aplpy and gridspec
- automated queries to imaging surveys (DSS2, GAIA)
- dealing with Sky Coordinates in astropy
- code structuring and packaging [intermediate]
- code documentation with sphinx [intermediate-advanced]
- interacting with p2 [intermediate-advanced]

Participants will be free to start from whatever angle they so-desire, to explore the aspects closest-to-their-heart: plotting, web-queries, astropy, sphinx, p2api

Required background, skills:
None to expert, depending on the chosen path.

Required python packages:
- python 3.6 and the ususal (numpy, scipy, etc…)
- matplotlib
- aplpy
- astropy
- astroquery
- Montage (wrapper + base, optional, strongly recommended, but not a show killer)

Project 4 - coordinated by Cristian Herrera and Eleonora Sani

Title:
IDL2PYTHON

Abstract:
So, in principle the idea behind this Project is to convert these IDL scripts to Python one´s, this way most of the Astronomer currently programming in IDL can move to Python Programming.
Some example IDL codes are:
- Adaptively spatial bin two-dimensional data to a constant signal-to-noise ratio per bin (Voronoi Binning)
- Extract the stellar or gas kinematics from galaxy spectra via full spectrum fitting (pPXF)

Required background, skills:
Knowledge on IDL and Python to intermediate.

Required python packages:
- ANACONDA (Python 3.6 and astropy, numpy, scipy, etc)
- Spyder 3.0 or Jupyter (IPython)
- Matplotlib or Orange 3 (Optional)
- IDL to Python Bridge
- IPython IDL Kernel
- Another packages to manipulate IDL scripts (TBD)

Project 5 - coordinated by Elyar Sedaghati

Title:
Star Visibility Plots

Abstract:
This project will involve writing a python code that will calculate the visibility of an object, given its coordinates, for a given location on earth. At the end, the participants should have a code that plots the airmass of the object as a function of time, as well as the moon.

Required background, skills:

Required python packages:
- numpy
- matplotlib
- astropy

Project 6 - coordinated by Joe Anderson

Title:
An image quality python script for quick-look QC0 for FORS2 imaging mode

Abstract:

Required background, skills:
None

Required python packages:
- python 3.5+ and the ususal (numpy, scipy, etc…)
- pyraf