Outline

Web: https://github.com/nickalaskreynolds/nkrpy
Author: Nickalas Reynolds <nickalaskreynolds@gmail.com>
Author Web: reynolds.oucreate.com

Date: 2018-12-10 23:23:34.625807

Description: This file fully explores all directories of the module *nkrpy*.

- **nkrpy/**
    - .rst_pdf.json <--
    - README.md <--
    - outline.rst <--
    - setup.py <--
    - outline.html5 <--
    - outline.pdf <--
    - makefile <--
    - **bin/**
        - template <--
        - outlinegen.py <--"""This file fully explores all directories of the module *nkrpy*."""
        - docgen.sh <--
        - **templates/**
            - template.py <--""".."""
            - template.md <--
            - template.rst <--
            - template.sh <--
        - **misc/**
            - paul_bootstrap.py <--
            - arcsat_nightlog_creator.sh <--
            - submit_jobs.py <--
            - matplotlib_colors.py <--
            - QL_ARCSAT.py <--
            - fft_h370_example.ipynb <--
            - **tspec_analysis/**
                - template_analysis.ipynb <--
                - README.md <--
    - **nkrpy/**
        - constants.py <--
        - coordinates.py <--
        - error.py <--
        - functions.py <--"""Just generic functions that I use a good bit."""
        - linelist.py <--"""Main linelist for various wavelength bands. The main
        - astro.py <--
        - atomicline.py.new <--
        - colours.py <--

- files.py <--"""."""
- load.py <--"""."""
- __info__.py <--
- keplerian.py <--"""orbital_params(lsma,usma,le,ue,li,ui,mass,size). Use orbital_params or orbital_2_xyz as the main function call.
- config.py <--
- check_file.py <--"""."""
- sorting.py <--
- atomiclines.py <--
- sizeof.py <--
- miscmath.py <--
- decorators.py <--"""Generalized decorators for common usage."""
- stdio.py <--
- **dustmodels/**
    - oh1994.tsb <--
    - README.md <--
    - kappa.py <--"""Just generic functions that I use a good bit."""
- **plot/**
    - styles.py <--
- **mercury/**
    - orbit.py <--"""This packages tries to be fairly robust and efficient, utilizing the speedups offered via numpy where applicable and multicore techniques. To get started, simply need a config file and call orbit.main(config). Inside the config should be mostly 3 things: files<input file list> out_dir<outputdirectory> and out_name<unique output name>. A lot of files will be generated (sometimes tens of thousands). The end goal is matplotlib libraries are ineffient for animation creation, so static thumbnails are created and then a imagmagick shell script is created to utilize a more efficient program."""
    - config_orbit.py <--
    - config_plotting.py <--
    - file_loader.py <--
    - plotting.py <--
- **image/**
    - image_interp.py <--
    - image_reproj.py <--

- **apo/**
    - combined_orders_template.ipynb <--
    - fits.py <--"""."""
    - guidecam_thumbnail.py <--"""Just call this module as a file while inside the directory of guidecam images."""
    - reduction.py <--
    - apoexpcal.pro <--
    - generate_ipynb.sh <--
    - **arcsat/**
        - template_config.py <--
        - arcsat_file.py <--"""."""
        - reduction.py <--"""Handles bulk reduction for ARCSAT. Must have a config file defined and tries to do basic reduction quickly."""
        - arcsat_mosaic.py <--"""."""
- **check_file_templates/**
    - default.py <--
    - sh.py <--
    - python.py <--