Read the Docs Template Documentation

Release 1.0

Read the Docs

Jun 27, 2023

CONTENTS

Contents:

CHAPTER

ONE

WHY PYEARTH?

More than 10 years ago, I relied on both ArcGIS Arc Engine/Object and MATLAB/IDL for various GIS tasks.

The ESRI ArcGIS system is a powerful GIS platform, but it is not free to use. More importantly, it is not easy to run ArcGIS tools in a Linux environment to take the advantage of the parallel computing power of a Linux cluster.

Although MATLAB may be one step ahead of ArcGIS, it is also not free to use. And I had issues with its scaling capability. What's more, MATLAB is not designed to work with GIS data. For example, it is not an easy task to mosaic multiple remote sensing images in MATLAB.

IDL was a clear winner at that time. It supports GIS/RS, it can be run in a Linux environment, and it is easy to scale. By the time, the popular IDL Coyote library (http://www.idlcoyote.com/) is my go-to library for GIS/RS tasks. I even built a library on top of the Coyote library.

IDL solves most of my problems, but it has some limitations other than the license fee. Namely, IDL is mostly conceived as functional programming language, which means it is often not used to build large-scale applications. Meanwhile, I started using Python for other tasks. Linking Python with IDL is possiable but trivial. Since I use HPC, that means I need to use two commands to finish a task instead of one, same for debugging.

Besides, I started to work in the FAIR (fairness, accountability, and transparency) domain once I started using Python.

This is the point I decided to rewrite my IDL library in Python and the birth of PyEarth. PyEarth is still functional-oriented, meaning, each function is designed to do one thing. PyEarth functions are implemented to support real-world tasks such as reading a raster file.

PyEarth becomes the suporting library for my other projects, such as the PyFlowline project. The principle is very straightforward: if a feature is needed more than once, it should be abstracted as a function in PyEarth.

PyEarth uses the Python ecosystem to do lots of simple things easier. For example, I don't want to cope and paste a large chuck of code to plot a time series data. In PyEarth, that is one single function call with various customization options.

Thank you.

CHAPTER

TWO

GETTING STARTED

CHAPTER

THREE

INSTALLATION

You can use the Python package manager pip to install pyearth:

```
pip install pyearth
```

You can also use the Conda package manager to install pyearth:

```
conda install -c conda-forge pyearth
```

CHAPTER

FOUR

USAGE

To use functions from pyearth, you need to import the package or the functions directly:

```
import pyearth
from pyearth.visual.color.create_diverge_rgb_color_hex import create_diverge_rgb_color_hex
```

CHAPTER

FIVE

GIS/RS

5.1 envi

- envi_write_header: generate an ENVI header file from a numpy array

5.2 GDAL

5.2.1 Read

- gdal read ArcGIS ASCII raster
- gdal read shapefile
- gdal geotiff raster
- gdal read envi raster

5.2.2 Write

- gdal write envi file
- gdal geotiff file

5.2.3 Other gdal functions

- reproject_coordinates: reproject coordinates from one projection to another
- reproject_coordinates_batch: reproject coordinates from one projection to another for a batch of points
- obtain_raster_metadata_geotiff: obtain metadata from a geotiff raster
- obtain_shapefile_metadata: obtain metadata from a shapefile

5.2.4 Location related functions

- calculate_distance_based_on_lon_lat: calculate the great circle distance between two points based on their longitude and latitude
- calculate_polygon_area: calcualte the spherical area of a polygon
- convert_360_to_180: convert longitude from 0-360 to -180 to 180

CHAPTER

SIX

SYSTEM

This module mainly defines several system-wide settings.

CHAPTER

SEVEN

TOOLBOX

7.1 Data

7.2 Date

- day_in_month
- day_of_year

7.3 Geometry

7.4 Math

7.5 Reader

- text_reader_string

7.6 Slurm

CHAPTER

EIGHT

VISUALIZATION

8.1 Color

8.2 Formatter

8.3 Barplot

- barplot_data

8.4 Boxplot

- boxplot_data

8.5 Histogram

- histogram_data

8.6 Ladder plot

8.7 Map

8.8 Ridge plot

8.9 Scatter plot

8.10 Surface plot

8.11 Time series

CHAPTER

NINE

API REFERENCE

CHAPTER

TEN

INDICES AND TABLES

- genindex
- modindex
- search