astroph

Release v0.0

Mateus Piovezan Otto

Dec 07, 2019

CONTENTS:

CHAPTER

ONE

GETTING STARTED TO *SPHINX*

The syntax is pretty much like *Markdown*. Are them related?

1.1 Documenting objects

enumerate(*sequence*[, *start=0*])
Return an iterator that yields tuples of an index and an item of the *sequence*. (And so on.)

the *enumerate()* function can be used for ...

Referencing `io.open()`

1.1.1 Testing some headings

class xtalphases.data.preprocess.**CIFParser**(*filename, cif_columns=['index_h', 'index_k', 'index_l', 'FOBS', 'SIGFOBS', 'UOBS', 'SIGUOBS', 'FC', 'PHI', 'FOM', 'RESOL', 'pdbx_r_free_flag']*)

get_pdb_name()
Get PDB_ID of the file being parsed.

Returns **name**

Return type PDB_ID of the file being parsed.

header_parse()
Parse numeric and textual info on .cif header.

header_refln_df(*headercols=None*, *reflncols=None*, *phierror=False*)
Convert header and reflection into a single DataFrame. Header and reflections columns can be specified via headercols and reflncols arguments.

Parameters

- **headercols** (*header columns to be included in the final DataFrame.*) –
- **reflncols** (*reflection columns to be included in the final DataFrame.*) –

Returns **header_refln_df**

Return type DataFrame containing both header and reflection info.

header_to_df(*columns=None*)
Convert header dictionary to a pandas DataFrame.

Returns

- **cifheader_df** (*pandas DataFrame containing all information*)
- *stored on the header dictionary.*

header_to_series(*columns=None*)
Convert header dictionary to a pandas Series.

Returns

- **cifheader_series** (*pandas Series containing all information*)
- *stored on the header dictionary.*

parse(*integer_indexes=['index_h', 'index_l', 'index_k', 'pdbx_r_free_flag']*)
Parse both header and reflections.

reflections_parse(*integer_columns=['index_h', 'index_l', 'index_k', 'pdbx_r_free_flag']*)
Parse reflections on .cif file.

Parameters

- **integer_indexes** (*reflection columns containing*) –
- **entries.** (*integer*) –

reflections_to_df(*columns=None*)
Convert reflection dictionary to a pandas DataFrame.

Returns

- **refln_df** (*pandas DataFrame containing all information*)
- *stored on the reflection dictionary.*

class astroph.random_code.**FaradayRotation**(*polarization_angles*)
A class for Faraday Rotation measurements to determine interstellar magnetic fields.

Parameters **polarization_angles** (*array*) – Polarization angle measurements.

astroph.random_code.FaradayRotation.**_get_magnetic_field_map**(*self*)
Testing math docstring.

Returns **map** – Array representing the magnetic field map. Here $\alpha < \beta$.

Return type array

$$\mathbf{B} = \frac{\mu_0}{4\pi}$$

```
>>> import math
>>> math.sqrt(2.)
1.4142135623730951
```

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

INDEX