

---

# **TLE-tools**

***Release 0.1.1***

**Sep 08, 2019**



---

## Contents:

---

<b>1</b>	<b>Purpose</b>	<b>3</b>
<b>2</b>	<b>Installation</b>	<b>5</b>
<b>3</b>	<b>Links</b>	<b>7</b>
<b>4</b>	<b>Indices and tables</b>	<b>9</b>
4.1	API Documentation . . . . .	9
	<b>Python Module Index</b>	<b>13</b>
	<b>Index</b>	<b>15</b>



**TLE-tools** is a small library to work with [two-line element set](#) files.



# CHAPTER 1

---

## Purpose

---

The purpose of the library is to parse TLE sets them into convenient TLE objects, load entire TLE set files into `pandas.DataFrame`'s, convert TLE objects into `poliastro.twobody.Orbit`'s, and more.

From [Wikipedia](#):

A two-line element set (TLE) is a data format encoding a list of orbital elements of an Earth-orbiting object for a given point in time, the epoch. The TLE data representation is specific to the [simplified perturbations models](#) (SGP, SGP4, SDP4, SGP8 and SDP8), so any algorithm using a TLE as a data source must implement one of the SGP models to correctly compute the state at a time of interest. TLEs can describe the trajectories only of Earth-orbiting objects.

Example:

```
ISS (ZARYA)
1 25544U 98067A   19249.04864348   .00001909   00000-0   40858-4   0   9990
2 25544   51.6464 320.1755 0007999   10.9066   53.2893 15.50437522187805
```





## CHAPTER 2

---

### Installation

---

Install and update using `pip`:

```
pip install -U TLE-tools
```



## CHAPTER 3

---

### Links

---

- Website: <https://github.com/FedericoStra/tletools>
- Documentation: <https://tletools.readthedocs.io/>
- Releases: <https://pypi.org/project/TLE-tools/>
- Code: <https://github.com/FedericoStra/tletools>
- Issue tracker: <https://github.com/FedericoStra/tletools/issues>



- [genindex](#)
- [modindex](#)
- [search](#)

## 4.1 API Documentation

If you are looking for information on a specific function, class, or method, this part of the documentation is for you.

### 4.1.1 API Documentation

This part of the documentation covers all the interfaces of `tle`. For guides on how to use them, please consult the tutorials.

#### TLE Classes

The library offers two classes to represent a single TLE. There is the unitless version `TLE`, whose attributes are expressed in the same units that are used in the TLE format, and there is the unitful version `TLEu`, whose attributes are quantities (`astropy.units.Quantity`), a type able to represent a value with an associated unit taken from `astropy.units`.

**TLE-tools** is a small library to work with [two-line element set](#) files.

**class** `tle.TLE` (*name, norad, classification, int\_desig, epoch\_year, epoch\_day, dn\_o2, ddn\_o6, bstar, set\_num, inc, raan, ecc, argp, M, n, rev\_num*)  
Data class representing a single TLE.

A two-line element set (TLE) is a data format encoding a list of orbital elements of an Earth-orbiting object for a given point in time, the epoch.

All the attributes parsed from the TLE are expressed in the same units that are used in the TLE format.

### Variables

- **name** (*str*) – name of the satellite
- **norad** (*str*) – NORAD catalog number ([https://en.wikipedia.org/wiki/Satellite\\_Catalog\\_Number](https://en.wikipedia.org/wiki/Satellite_Catalog_Number))
- **classification** (*str*) – ‘U’, ‘C’, ‘S’ for unclassified, classified, secret
- **int\_desig** (*str*) – international designator ([https://en.wikipedia.org/wiki/International\\_Designator](https://en.wikipedia.org/wiki/International_Designator))
- **epoch\_year** (*int*) – year of the epoch
- **epoch\_day** (*float*) – day of the year plus fraction of the day
- **dn\_o2** (*float*) – first time derivative of the mean motion divided by 2
- **ddn\_o6** (*float*) – second time derivative of the mean motion divided by 6
- **bstar** (*float*) – BSTAR coefficient (<https://en.wikipedia.org/wiki/BSTAR>)
- **set\_num** (*int*) – element set number
- **inc** (*float*) – inclination
- **raan** (*float*) – right ascension of the ascending node
- **ecc** (*float*) – eccentricity
- **argp** (*float*) – argument of perigee
- **M** (*float*) – mean anomaly
- **n** (*float*) – mean motion
- **rev\_num** (*int*) – revolution number

```
class tle.TLEu(name, norad, classification, int_desig, epoch_year, epoch_day, dn_o2, ddn_o6, bstar,
               set_num, inc, raan, ecc, argp, M, n, rev_num)
    Unitful data class representing a single TLE.
```

This is a subclass of [TLE](#), so refer to that class for a description of the attributes and properties.

The only difference here is that all the attributes are quantities (`astropy.units.Quantity`), a type able to represent a value with an associated unit taken from `astropy.units`.

### Module functions

```
tle.load_dataframe(filename, *, epoch=True)
    Load multiple TLEs from one or more files and return a pandas.DataFrame.
```

### Convenience functions

```
tle.partition(iterable, n)
    Partition an iterable into tuples.
```

The iterable *iterable* is progressively consumed *n* items at a time in order to produce tuples of length *n*.

#### Parameters

- **iterable** (*iterable*) – The iterable to partition.
- **n** (*int*) – Length of the desired tuples.

**Returns** A generator which yields subsequent n-uples from the original iterable.

`tle.add_epoch(df)`

Add a column 'epoch' to a dataframe.

*df* must have columns 'epoch\_year' and 'epoch\_day', from which the column 'epoch' is computed.

**Parameters** *df* (*pandas.DataFrame*) – *pandas.DataFrame* instance to modify.





**t**

tle, 9



## A

`add_epoch()` (*in module tle*), 11

## L

`load_dataframe()` (*in module tle*), 10

## P

`partition()` (*in module tle*), 10

## T

`TLE` (*class in tle*), 9

`tle` (*module*), 9

`TLEu` (*class in tle*), 10