solarsystem

Release 0.1.3

Jan 31, 2020

Contents:

CHAPTER 1

solarsystem package

1.1 Submodules

1.1.1 solarsystem.functions module

solarsystem.functions.**Planet_Sun**(*M*, *e*, *a*, *N*, *w*, *i*)
Helper Function. From planet's trajectory elements to position around sun

Returns position elements

Return type tuple

solarsystem.functions.**demical2arcs**(*num*)
Convert Demical view to Degrees and minutes.

Parameters **num** (*float*) – degrees to be converted.

Returns one string representation in degrees and minutes format.

Return type str

solarsystem.functions.**demical2clock**(*demicaltime*)
Convert demical time view to Hours, Minutes and Seconds.

Parameters **demicaltime** (*float*) – time to be converted.

Returns one string representation in hours, minutes format.

Return type str

solarsystem.functions.**demical2hms**(*degrees*)
Convert degrees to string representation of hours, minutes and seconds.

Parameters **degrees** (*float*) – degrees to be converted.

Returns one string representation in hours, minutes and seconds format.

Return type str

solarsystem.functions.**ecliptic2equatorial**(*xeclip*, *yeclip*, *zeclip*, *oblecl*)
Transform ecliptic to equatorial projection.

Parameters

- **xeclip** – value on x axis of ecliptic plane.
- **yeclip** – value on y axis of ecliptic plane.
- **zeclip** – value on z axis of ecliptic plane.
- **oblecl** – obliquity of the ecliptic, approximately 23.4 degrees for earth

Returns x, y, z equatorial projection

Return type tuple

solarsystem.functions.**equatorial2ecliptic**(*xequat*, *yequat*, *zequat*, *oblecl*)
Transform equatorial to ecliptic projection.

Parameters

- **xequat** – value on x axis of equatorial plane
- **yequat** – value on y axis of equatorial plane
- **zequat** – value on z axis of equatorial plane
- **oblecl** – obliquity of the ecliptic, approximately 23.4 degrees for earth

Returns x, y, z ecliptic projection

Return type tuple

solarsystem.functions.**normalize**(*degrees*)
set degrees always between 0 - 360

Parameters **degrees** (*float*) – degrees to be adjusted

Returns degrees between 0-360

Return type float

solarsystem.functions.**rectangular2spherical**(*x*, *y*, *z*)
Transform rectangular to spherical projection.

From rectangular(x,y,z) coordinates system to spherical (RA,Decl, r) or by replacing x with azimuth and y with altitude we can tranform horizontal coordinates to ecliptic (longitude, latitude).

Parameters

- **x** – value on x axis of a rectangular projection.
- **y** – value on y axis of a rectangular projection.
- **z** – value on z axis of a rectangular projection.

Returns RA, Decl, r spherical coordinate system.

Return type tuple

solarsystem.functions.**spherical2rectangular**(*RA*, *Decl*, *r*)
Transform spherical to rectangular projection.

From spherical (RA,Decl) coordinates system to rectangular(x,y,z) or by replacing RA with longitude and Decl with latitude we can tranform ecliptic coordinates to horizontal (azimuth,altitude).

Parameters

- **RA** – Right Ascension.
- **Decl** – Declination.
- **r** – Distance in astronomical units.

Returns x, y, z rectangular coordinate system.

Return type tuple

solarsystem.functions.**spherical_ecliptic2equatorial**(*long*, *lat*, *distance*, *oblecl*)
Transform eclipitc to spherical projection for given obliquity.

From spherical (RA, Decl, distance) coordinates system to eclipitc(long, lat, distance).

Parameters

- **long** – Longitude.
- **last** – Latitude.
- **distance** – Distance in astronomical units.
- **oblecl** – obliquity (axial tilt).

Returns RA, Decl, distance spherical coordinate system.

Return type tuple

solarsystem.functions.**spherical_equatorial2ecliptic**(*RA*, *Decl*, *distance*, *oblecl*)
Transform spherical to eclipitc projection for given obliquity.

From spherical (RA, Decl, distance) coordinates system to eclipitc(long, lat, distance).

Parameters

- **RA** – Right Ascension.
- **Decl** – Declination.
- **distance** – Distance in astronomical units.
- **oblecl** – obliquity (axial tilt).

Returns long, lat, distance eclipitc coordinate system.

Return type tuple

solarsystem.functions.**sun2planet**(*xeclip*, *yeclip*, *zeclip*, *x*, *y*, *z*)
Helper Function. From Hliocentric to Geocentric position

Returns geocentric view of object.

Return type tuple

1.1.2 solarsystem.geocentric module

class solarsystem.geocentric.**Geocentric**(*year*, *month*, *day*, *hour*, *minute*, *UT=0*, *dst=0*, *plane='ecliptic'*)
Bases: object

Import date data outputs planets positions around Earth.

Parameters

- **year** (*int*) – Year (4 digits) ex. 2020
- **month** (*int*) – Month (1-12)

- **day** (*int*) – Day (1-31)
- **hour** (*int*) – Hour (0-23)
- **minute** (*int*) – Minute (0-60)
- **UT** – Time Zone (deviation from UT, -12:+14), ex. for Greece (GMT + 2) enter UT = 2
- **dst** (*int*) – daylight saving time (0 or 1). Wheather dst is applied at given time and place
- **plane** – desired output format. Should be one of: ecliptic, equatorial. Default: ecliptic

objectnames()
Names of solar system objects used.

Returns A list of solar system objects

Return type list

position()
Main method which returns a dictionary of geocentric positions.

Returns

Planet positions around earth: Each row represents a planet and each column the position of that planet.

Return type dictionary

1.1.3 solarsystem.heliocentric module

class solarsystem.heliocentric.**Heliocentric**(*year*, *month*, *day*, *hour*, *minute*, *UT=0*, *dst=0*, *view='horizontal'*)

Bases: object

Import date data outputs planets positions around Sun.

Parameters

- **year** (*int*) – Year (4 digits) ex. 2020
- **month** (*int*) – Month (1-12)
- **day** (*int*) – Day (1-31)
- **hour** (*int*) – Hour (0-23)
- **minute** (*int*) – Minute (0-60)
- **UT** – Time Zone (deviation from UT, -12:+14), ex. for Greece (GMT + 2) enter UT = 2
- **dst** (*int*) – daylight saving time (0 or 1). Wheather dst is applied at given time and place
- **view** – desired output format. Should be one of: horizontal (long in degrees, lat in degrees, distance in AU) or rectangular (x, y, z, all in AU). Default: horizontal.

planetnames()
Names of solar system objects used.

Returns A list of solar system objects.

Return type list

planets()
Main method which returns a dictionary of Heliocentric positions.

Returns Planet positions around sun: Dictionary of tuples. Each row represents a planet and each column the position of that planet.

Return type dictionary

1.1.4 solarsystem.moon module

class solarsystem.moon.**Moon**(*year*, *month*, *day*, *hour*, *minute*, *UT=0*, *dst=0*, *longtitude=0.0*, *latitude=51.48*, *topographic=False*)

Bases: object

Import date and place outputs moons position, phase and rise-set time.

Moon is a class that feeded with date data as well as geocoordicates outputs moons position around Earth, moon phase and moonrise-moonset/

Parameters

- **year** (*int*) – Year (4 digits) ex. 2020
- **month** (*int*) – Month (1-12)
- **day** (*int*) – Day (1-31)
- **hour** (*int*) – Hour (0-23)
- **minute** (*int*) – Minute (0-60)
- **UT** – Time Zone (deviation from UT, -12:+14), ex. for Greece (GMT + 2) enter UT = 2
- **dst** (*int*) – daylight saving time (0 or 1). Wheather dst is applied at given time and place
- **longtitude** (*float*) – longitude of place of Moonrise - Moonset in demical format
- **latitude** (*float*) – latitude of place of Moonrise-Moonset in demical format
- **topographic** (*bool*) – Wheather or not moon's position around earth will be calculated regarding earth surface or center

moonriseset()

Method which returns moon's rise and set time

Returns Moon's time of given date where moon rises and sets

Return type tuple

phase()

Method which returns moon's phase

Returns Moon's phase (percent of illumination)

Return type float

position()

Method which returns moon's position around Earth

Returns Moon's positions around earth in horizontal projection (long, lat and distance in multiple of earth radius)

Return type tuple

1.1.5 solarsystem.sunriseset module

class solarsystem.sunriseset.**Sunriseset**(*year*, *month*, *day*, *UT=0*, *dst=0*, *longtitude=0.0*, *latitude=51.48*)

Bases: object

Import date outputs Sunrise and Sunset time.

Parameters

- **year** (*int*) – Year (4 digits) ex. 2020.
- **month** (*int*) – Month (1-12).
- **day** (*int*) – Day (1-31).
- **UT** – Time Zone (deviation from UT, -12:+14), ex. for Greece (GMT + 2) enter UT = 2.
- **dst** (*int*) – daylight saving time (0 or 1). Wheather dst is applied at given time and place.
- **longtitude** (*float*) – longitude of place of Sunrise - Sunset in demical format.
- **latitude** (*float*) – latitude of place of Sunrise - Sunset in demical format.

riseset()

Get the time of sun rise and set within given date.

Returns Sunrise - Sunset time of given date

Return type tuple

1.2 Module contents

1.2.1 SolarSystem

- Planets positions around Sun.
- Planets positions around Earth.
- Moon positions around Earth.
- Sunrise/Sunset.
- Moonrise/Moonset, Moon Phase.
- Convert between different coordinate systems.

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

S

Index