Table of Contents

[Package Structure: 1](#_Toc102319744)

[horoscope module - functions 1](#_Toc102319745)

[Required imports 1](#_Toc102319746)

[Lambda functions 1](#_Toc102319747)

[get\_ayanamsa\_value (jd) 1](#_Toc102319748)

[set\_ayanamsa\_mode (ayanamsa\_mode = "KP", ayanamsa\_value=None, jd=None) 1](#_Toc102319749)

[nakshatra\_pada (longitude) 1](#_Toc102319750)

[sidereal\_longitude (jd, planet) 1](#_Toc102319751)

[sunrise(jd, place, as\_string=False) 1](#_Toc102319752)

[sunset(jd, place, as\_string=False) 1](#_Toc102319753)

[moonrise(jd, place, as\_string=False) 1](#_Toc102319754)

[moonset(jd, place, as\_string=False) 1](#_Toc102319755)

[tithi(jd, place, as\_string=False) 1](#_Toc102319756)

[raasi(jd, place, as\_string=False) 1](#_Toc102319757)

[nakshatra(jd, place, as\_string=False) 1](#_Toc102319758)

[yogam(jd, place, as\_string=False) 1](#_Toc102319759)

[karana(jd, place, as\_string=False) 1](#_Toc102319760)

[vaara(jd, as\_string=False) 1](#_Toc102319761)

[maasa(jd, place, as\_string=False) 1](#_Toc102319762)

[elapsed\_year(jd, maasa\_index) 1](#_Toc102319763)

[samvatsara(jd, maasa\_index, as\_string=False, north\_indian\_tradition=False) 1](#_Toc102319764)

[ritu(maasa\_index) 1](#_Toc102319765)

[gauri\_chogadiya(jd, place,as\_string=False) 1](#_Toc102319766)

[trikalam(jd, place, option='raahu kaalam',as\_string=False) 1](#_Toc102319767)

[durmuhurtam(jd, place,as\_string=False) 1](#_Toc102319768)

[abhijit\_muhurta(jd, place, as\_string=False) 1](#_Toc102319769)

[planetary\_positions(jd, place,as\_string=False) 1](#_Toc102319770)

[ascendant(jd, place, as\_string=False) 1](#_Toc102319771)

[dasavarga\_from\_long(longitude, sign\_division\_factor) 1](#_Toc102319772)

[dhasavarga(jd, place, sign\_division\_factor, as\_string=False) 1](#_Toc102319773)

[solar\_upagraha\_longitudes(jd, upagraha, divisional\_chart\_factor=1, as\_string=False) 1](#_Toc102319774)

[upagraha\_longitude (dob, tob, place, planet\_index, divisional\_chart\_factor=1, upagraha\_part='middle',as\_string=False) 1](#_Toc102319775)

[special\_ascendant(jd, place, time\_of\_birth\_in\_hours, lagna\_rate\_factor=1.0, divisional\_chart\_factor=1, as\_string=False) 1](#_Toc102319776)

[sree\_lagna (jd, place, divisional\_chart\_factor=1, as\_string=True) 1](#_Toc102319777)

# Package Structure:

hora

!- data - contains program configuration data, world cities data, marriage compatibility table

!- ephe - contains swiss ephimeride compressed JPL data

!- images - contains images

!- lang - contains language resource files

!- panchanga - panchanga module to calculate daily panchanga

!- horoscope

!- horoscope.py - horoscope package

!- chart - chart package

!- arudhas.py - arudhas, argala, virodhargal

!- ashtavarga.py - ashtavarga, trikona sodhana, ekadhipatya\_sodhana, sodhaya pinda

!- charts.py - divisional charts, planet combustion, retrograde

!- house.py - aspects, drishti,stronger planets/raasi, kaarakas

!- yoga.py - 100+ yogas

!- raja\_yoga.py - raja\_yoga and its sub-types

!- dhasa - dhasa package

!- ashtottari.py - ashtottari dhasa-bhuthi

!- drig.py - drigdhasa-bhuthi

!- kalachakra.py - kalachakra dhasa-bhuthi

!- moola.py - moola dhasa-bhuthi

!- mudda.py - mudda dhasa-bhuthi

!- narayana.py - narayana dhasa-bhuthi

!- nirayana.py - nirayana dhasa-bhuthi

!- patyayini.py - patyayini dhasa-bhuthi

!- shoola.py - shoola dhasa-bhuthi

!- sudasa.py - sudasa dhasa-bhuthi

!- sudharsana\_chakra.py - sudharsana\_chakra dhasa-bhuthi

!- vimsottari.py - vimsottari dhasa-bhuthi

!- match - marriage compatibility package

!- compatibility.py - marriage compatibility

!- transit - tajaka package

!- tajaka.py - annual, monthly and 60 hour charts, muntha, vargeeya balas, tajaka lord

!- tajaka\_yoga.py - tajaka yogas

!- saham.py - 36 sahams

!- ui - user interface package

!- horo\_chart.py - simple horoscope chart Raasi/Navamsa and calendar information

!- horo\_chart\_tabs.py - horoscope with lot of details

!- match\_ui.py - ui for marriage compatibility

!- utils.py - utility functions

!- const.py - constants related to PyHora package

!- tests - unit/integration tests

!- unit\_tests.py - unit tests for the features based on examples from the book

!- pvr\_tests.py - Exercise problems from book.

# horoscope module - functions

To calculate panchanga/calendar elements such as tithi, nakshatra, etc.

Uses swiss ephemeris

## Required imports

from geopy.geocoders import Nominatim

from pytz import timezone, utc

from math import ceil

from collections import namedtuple as struct

import swisseph as swe

from \_datetime import datetime, timedelta

from datetime import date

import math, os, warnings

from collections import OrderedDict as Dict

from hora import utils, const

Since datetime does not accept BC year values Use the following stucture to represent dates

Date = struct('Date', ['year', 'month', 'day'])

To represent Place use the following structure as an argument

Place = struct('Place', ['Place','latitude', 'longitude', 'timezone'])

## Lambda functions

revati\_359\_50 = lambda: swe.set\_sid\_mode(swe.SIDM\_USER, 1926892.343164331, 0)

galc\_cent\_mid\_mula = lambda: swe.set\_sid\_mode(swe.SIDM\_USER, 1922011.128853056, 0)

reset\_ayanamsa\_mode = lambda: swe.set\_sid\_mode(swe.SIDM\_FAGAN\_BRADLEY)

from\_dms = lambda degs, mins, secs: degs + mins/60 + secs/3600

Ketu is always 180° after Rahu, so same coordinates but different constellations

i.e if Rahu is in Pisces, Ketu is in Virgo etc

ketu = lambda rahu: (rahu + 180) % 360

rahu = lambda ketu: (ketu + 180) % 360

Julian Day number as on (year, month, day) at 00:00 UTC

gregorian\_to\_jd = lambda date: swe.julday(date.year, date.month, date.day, 0.0)

jd\_to\_gregorian = lambda jd: swe.revjul(jd, swe.GREG\_CAL) returns (y, m, d, h, min, s)

solar\_longitude = lambda jd: sidereal\_longitude(jd, swe.SUN)

lunar\_longitude = lambda jd: sidereal\_longitude(jd, swe.MOON)

raahu\_kaalam = lambda jd, place,as\_string=False: trikalam(jd, place, 'raahu kaalam', as\_string=as\_string)

yamaganda\_kaalam = lambda jd, place,as\_string=False: trikalam(jd, place, 'yamagandam', as\_string=as\_string)

gulikai\_kaalam = lambda jd, place,as\_string=False: trikalam(jd, place, 'gulikai',as\_string=as\_string)

navamsa\_from\_long = lambda longitude: dasavarga\_from\_long(longitude,9)

kaala\_longitude = lambda dob,tob,place,divisional\_chart\_factor=1,as\_string=False: upagraha\_longitude(dob,tob,place,planet\_index=0,divisional\_chart\_factor=divisional\_chart\_factor,upagraha\_part='middle',as\_string=as\_string)

mrityu\_longitude = lambda dob,tob,place,divisional\_chart\_factor=1,as\_string=False: upagraha\_longitude(dob,tob,place,planet\_index=2,divisional\_chart\_factor=divisional\_chart\_factor,upagraha\_part='middle',as\_string=as\_string)

artha\_praharaka\_longitude = lambda dob,tob,place,divisional\_chart\_factor=1,as\_string=False: upagraha\_longitude(dob,tob,place,planet\_index=3,divisional\_chart\_factor=divisional\_chart\_factor,upagraha\_part='middle',as\_string=as\_string)

yama\_ghantaka\_longitude = lambda dob,tob,place,divisional\_chart\_factor=1,as\_string=False: upagraha\_longitude(dob,tob,place,planet\_index=4,divisional\_chart\_factor=divisional\_chart\_factor,upagraha\_part='middle',as\_string=as\_string)

gulika\_longitude = lambda dob,tob,place,divisional\_chart\_factor=1,as\_string=False: upagraha\_longitude(dob,tob,place,planet\_index=6,divisional\_chart\_factor=divisional\_chart\_factor,upagraha\_part='middle',as\_string=as\_string)

maandi\_longitude = lambda dob,tob,place,divisional\_chart\_factor=1,as\_string=False: upagraha\_longitude(dob,tob,place,planet\_index=6,divisional\_chart\_factor=divisional\_chart\_factor,upagraha\_part='begin',as\_string=as\_string)

bhava\_lagna = lambda jd,place,time\_of\_birth\_in\_hours,divisional\_chart\_factor,as\_string=False: special\_ascendant(jd,place,time\_of\_birth\_in\_hours,lagna\_rate\_factor=1.0,divisional\_chart\_factor=divisional\_chart\_factor,as\_string=as\_string)

hora\_lagna = lambda jd,place,time\_of\_birth\_in\_hours,divisional\_chart\_factor=1,as\_string=False: special\_ascendant(jd,place,time\_of\_birth\_in\_hours,lagna\_rate\_factor=0.5,divisional\_chart\_factor=divisional\_chart\_factor,as\_string=as\_string)

ghati\_lagna = lambda jd,place,time\_of\_birth\_in\_hours,divisional\_chart\_factor=1,as\_string=False: special\_ascendant(jd,place,time\_of\_birth\_in\_hours,lagna\_rate\_factor=1.25,divisional\_chart\_factor=divisional\_chart\_factor,as\_string=as\_string)

## get\_ayanamsa\_value (jd)

Get ayanamsa value for the julian day number

@param jd: Julian Day Number

@return: ayanamsa value - ayanamsa for the day based on the model used.

## set\_ayanamsa\_mode (ayanamsa\_mode = "KP", ayanamsa\_value=None, jd=None)

Set Ayanamsa mode

@param ayanamsa\_mode - Default - Lahiri

@param ayanamsa\_value - Need to be supplied only in case of 'SIDM\_USER'

See 'available\_ayanamsa\_modes' for the list of available models

@return None

## nakshatra\_pada (longitude)

Gives nakshatra (1..27) and paada (1..4) in which given longitude lies

@param longitude: longitude of the planet

@return [nakshathra index, paadham, longitude remainder in the nakshathra]

Note: Nakshatra index [1..27], Paadha [1..4] reminder in float degrees

## sidereal\_longitude (jd, planet)

Computes nirayana (sidereal) longitude of given planet on jd

Note: This is where the selected/default ayanamsa is adjusted to tropical longitude obtained from swiss ephimeride

@param jd: Julian Day Number of the date/time

@param planet: index of the planet 0..8. 0 is Sun, 1 = Moon, 7=Rahu, 8-Kethu

@return: the sidereal londitude of the planet

## sunrise(jd, place, as\_string=False)

Sunrise when centre of disc is at horizon for given date and place

@param jd: Julian Day Number of the date/time

@param place: Place as struct ('Place',;atitude,longitude,timezone)

@return sunrise time as julian day number, sunrise time in local time

Note: Time is returned as

degrees/minutes/seconds as string or tuple depending on as\_string=True and is\_lat\_long values

## sunset(jd, place, as\_string=False)

Sunset when centre of disc is at horizon for given date and place

@param jd: Julian Day Number of the date/time

@param place: Place as struct ('Place',;atitude,longitude,timezone)

@return sunset time as julian day number, sunset time in local time

Note: Time is returned as

degrees/minutes/seconds as string or tuple depending on as\_string=True and is\_lat\_long values

## moonrise(jd, place, as\_string=False)

Moonrise when centre of disc is at horizon for given date and place

@param jd: Julian Day Number of the date/time

@param place: Place as struct ('Place',;atitude,longitude,timezone)

@return moonrise time as julian day number, moonise time in local time

Note: Time is returned as

degrees/minutes/seconds as string or tuple depending on as\_string=True and is\_lat\_long values

## moonset(jd, place, as\_string=False)

Moonset when centre of disc is at horizon for given date and place

@param jd: Julian Day Number of the date/time

@param place: Place as struct ('Place',;atitude,longitude,timezone)

@return moonset time as julian day number, moonset time in local time

Note: Time is returned as

degrees/minutes/seconds as string or tuple depending on as\_string=True and is\_lat\_long values

## tithi(jd, place, as\_string=False)

Tithi at sunrise for given date and place. Also returns tithi's end time.

@param jd: Julian Day Number of the date/time

@param place: Place as struct ('Place',;atitude,longitude,timezone)

@return [tithi index, tithi ending time, next tithi index, next tithi ending time]

next tithi index and next tithi time is additionally returned if two tithis on same day

tithi index = [1..30]

Note: Time is returned as

degrees/minutes/seconds as string or tuple depending on as\_string=True and is\_lat\_long values

## raasi(jd, place, as\_string=False)

returns the raasi of the day

@param jd: Julian Day Number of the date/time

@param place: Place as struct ('Place',;atitude,longitude,timezone)

@return [raasi index, raasi ending time, next raasi index, next raasi ending time]

next raasi index and next raasi time is additionally returned if two raasis on same day

raasi index = [1..12]

Note: Time is returned as

degrees/minutes/seconds as string or tuple depending on as\_string=True and is\_lat\_long values

## nakshatra(jd, place, as\_string=False)

returns the nakshatra of the day

@param jd: Julian Day Number of the date/time

@param place: Place as struct ('Place',;atitude,longitude,timezone)

@return [nakshatra index, paadham index, nakshatra ending time,

next nakshatra index, next paadham index, next nakshatra ending time]

next nakshatra index and next nakshatra time is additionally returned if two nakshatra on same day

nakshatra index = [1..27] paadham index [1..4]

Note: Time is returned as

degrees/minutes/seconds as string or tuple depending on as\_string=True and is\_lat\_long values

## yogam(jd, place, as\_string=False)

returns the yogam of the day

@param jd: Julian Day Number of the date/time

@param place: Place as struct ('Place',;atitude,longitude,timezone)

@return [yogam index, yogam ending time,

next yogam index, next yogam ending time]

next yogam index and next yogam time is additionally returned if two yogam on same day

yogam index = [1..27] 1 = Vishkambha, 2 = Priti, ..., 27 = Vaidhrti

Note: Time is returned as

degrees/minutes/seconds as string or tuple depending on as\_string=True and is\_lat\_long values

## karana(jd, place, as\_string=False)

returns the karanam of the day

@param jd: Julian Day Number of the date/time

@param place: Place as struct ('Place',;atitude,longitude,timezone)

@return karanam index, karanam ending time,

karanam index = [1..60] 1 = Kimstugna, 2 = Bava, ..., 60 = Naga

## vaara(jd, as\_string=False)

Weekday for given Julian day.

@param jd: Julian Day Number of the date/time

@return: day of the date

0 = Sunday, 1 = Monday,..., 6 = Saturday

## maasa(jd, place, as\_string=False)

Returns lunar month and if it is adhika or not.

@param jd: Julian Day Number of the date/time

@param place: Place as struct ('Place',;atitude,longitude,timezone)

@return: indian month index, whether leap month (adhika maasa) or not - boolean

0 = Chaitra, 2 = Vaisakha, ..., 11 = Phalguna

True if adhika maasa

## elapsed\_year(jd, maasa\_index)

returns Indian era/epoch year indices (kali year number, saka year and vikrama year numbers)

@param jd: Julian Day Number of the date/time

@param maasa\_index: [1..12] (use panchanga.maasa function to get this)

@return kali year number, vikrama year number, saka year number

## samvatsara(jd, maasa\_index, as\_string=False, north\_indian\_tradition=False)

return the year name index for the given julian day number of the date

@param jd: Julian Day Number of the date/time

@param maasa\_index: [1..12] (use panchanga.maasa function to get this)

@param north\_indian\_tradition: Set to to True

Note: South Indian year names are off by 14 years compared to North Indian Tradition after Kali Year 4009.

@return year/samvastara index

1=Prabhava, 2=Vibhava... 59=Krodhana, 60=Akshaya

## ritu(maasa\_index)

returns ritu / season index.

@param maasa\_index: [1..12] (use panchanga.maasa function to get this)

@return: ritu index 0 = Vasanta,...,5 = Shishira

## gauri\_chogadiya(jd, place,as\_string=False)

Get end times of gauri chogadiya for the given julian day

Chogadiya is 1/8th of daytime or nighttime practiced as time measure in North India

@param jd: Julian Day Number of the date/time

@param place: Place as struct ('Place',;atitude,longitude,timezone)

@return: end times of chogadiya as a list

Note: Time is returned as

degrees/minutes/seconds as string or tuple depending on as\_string=True and is\_lat\_long values

## trikalam(jd, place, option='raahu kaalam',as\_string=False)

Get tri kaalam (Raahu kaalam, yama kandam and Gulikai Kaalam) for the given Julian day

@param jd: Julian Day Number of the date/time

@param place: Place as struct ('Place',;atitude,longitude,timezone)

@param option: one of 'raahu kaalam', 'gulikai', 'yamagandam'. Default:'raahu kaalam'

Note: One can use separate lambda function for each of these options

raahu\_kaalam = lambda jd, place,as\_string=False: trikalam(jd, place, 'raahu kaalam', as\_string=as\_string)

yamaganda\_kaalam = lambda jd, place,as\_string=False: trikalam(jd, place, 'yamagandam', as\_string=as\_string)

gulikai\_kaalam = lambda jd, place,as\_string=False: trikalam(jd, place, 'gulikai',as\_string=as\_string)

@return: start and end time of requested tri column - as list e.g. [start\_time, end\_time]

Note: Time is returned as

degrees/minutes/seconds as string or tuple depending on as\_string=True and is\_lat\_long values

## durmuhurtam(jd, place,as\_string=False)

Get dhur muhurtham timing for the given julian day

@param jd: Julian Day Number of the date/time

@param place: Place as struct ('Place',;atitude,longitude,timezone)

@return: start and end time of dhur muhurtham - as list e.g. [start\_time, end\_time]

Note: Time is returned as

degrees/minutes/seconds as string or tuple depending on as\_string=True and is\_lat\_long values

## abhijit\_muhurta(jd, place, as\_string=False)

Get Abhijit muhurta timing for the given julian day

Abhijit muhurta is the 8th muhurta (middle one) of the 15 muhurtas during the day\_duration (~12 hours)

@param jd: Julian Day Number of the date/time

@param place: Place as struct ('Place',;atitude,longitude,timezone)

@return: start and end time of Abhijit muhurta - as list e.g. [start\_time, end\_time]

Note: Time is returned as

degrees/minutes/seconds as string or tuple depending on as\_string=True and is\_lat\_long values

## planetary\_positions(jd, place,as\_string=False)

Computes instantaneous planetary positions (i.e., which celestial object lies in which constellation)

Also gives the nakshatra-pada division

@param jd: Julian Day Number of the date/time

@param place: Place as struct ('Place',;atitude,longitude,timezone)

@return: 2D List of [ [planet index, planet longitude, planet constellation],...]]

Example: [ [0,87.32148,2],...] - Sun longitude 87.32148, Gemini,...

## ascendant(jd, place, as\_string=False)

Compute Lagna (=ascendant) position/longitude at any given time & place

@param jd: Julian Day Number of the date/time

@param place: Place as struct ('Place',;atitude,longitude,timezone)

@return: [constellation of Lagna, longitude of lagna, Lagna nakshatra number, Lagna paadham number]

## dasavarga\_from\_long(longitude, sign\_division\_factor)

Calculates the dasavarga-sign in which given longitude falls

@param longitude: longitude of the planet

@param sign\_division\_factor: divisional chart index as below.

sign\_division\_factor = 2 => Hora, 3=>Drekana 4=>Chaturthamsa 5=>Panchamsa, 6=>Shashthamsa

7=>Saptamsa, 8=>Ashtamsa, 9=>Navamsa, 10=>Dasamsa, 11=>Rudramsa, 12=>Dwadamsa, 16=>Shodamsa,

20=>Vimsamsa, 24=>Chaturvimsamsa, 27=>Nakshatramsa, 30=>Trisamsa, 40=>Khavedamsa,

45=>Akshavedamsa, 60=>Shastyamsa

@return: constellation,longitude within\_raasi

0 = Aries, 1 = Taurus, ..., 11 = Pisces

## dhasavarga(jd, place, sign\_division\_factor, as\_string=False)

Calculate planet positions for a given divisional chart index

@param jd: Julian Day Number of the date/time

@param place: Place as struct ('Place',;atitude,longitude,timezone)

@param sign\_division\_factor: divisional chart index as below.

sign\_division\_factor = 2 => Hora, 3=>Drekana 4=>Chaturthamsa 5=>Panchamsa, 6=>Shashthamsa

7=>Saptamsa, 8=>Ashtamsa, 9=>Navamsa, 10=>Dasamsa, 11=>Rudramsa, 12=>Dwadamsa, 16=>Shodamsa,

20=>Vimsamsa, 24=>Chaturvimsamsa, 27=>Nakshatramsa, 30=>Trisamsa, 40=>Khavedamsa,

45=>Akshavedamsa, 60=>Shastyamsa

@return: 2D List of planet positions in the following format:

[ [planet\_index,[planet\_raasi, planet\_longitude],...]

The planet index is in range ['L',0..8] 'L' is Lagname (first element of the 2D List

## solar\_upagraha\_longitudes(jd, upagraha, divisional\_chart\_factor=1, as\_string=False)

Get logitudes of solar based upagrahas

['dhuma', 'vyatipaata', 'parivesha', 'indrachaapa', 'upaketu']

@param jd: Jilian day number for the date/time

@param upagraha: one of the values from ['dhuma', 'vyatipaata', 'parivesha', 'indrachaapa', 'upaketu']

@param divisional\_chart\_factor: divisional chart factor

divisional\_chart\_factor = 2 => Hora, 3=>Drekana 4=>Chaturthamsa 5=>Panchamsa, 6=>Shashthamsa

7=>Saptamsa, 8=>Ashtamsa, 9=>Navamsa, 10=>Dasamsa, 11=>Rudramsa, 12=>Dwadamsa, 16=>Shodamsa,

20=>Vimsamsa, 24=>Chaturvimsamsa, 27=>Nakshatramsa, 30=>Trisamsa, 40=>Khavedamsa,

45=>Akshavedamsa, 60=>Shastyamsa

@return: [constellation,longitude]

## upagraha\_longitude (dob, tob, place, planet\_index, divisional\_chart\_factor=1, upagraha\_part='middle',as\_string=False)

get upagraha longitude from dob,tob, place-lat/long and day/night ruling planet's part

@param dob Date of birth as Date(year,month,day)

@param tob Time of birth as (hours,minutes,seconds)

@param planet\_index 0=Sun, 1=Moon, 2=Mars, 3=Mercury,4=Jupiter, 5=Venus, 6=Saturn

@param upagraha\_part = 'Middle' or 'Begin'

Notes:

Since Kaala is upagraha of Sun so planet\_index should be 0

Kaala rises at the middle of Sun’s part. In other words, we find the time at the

middle of Sun’s part and find lagna rising then. That gives Kaala’s longitude.

Since Mrityu is upagraha of Mars so planet\_index should be 2

Mrityu rises at the middle of Mars’s part.

Since Artha Prabhakara is upagraha of Mercury so planet\_index should be 3

Artha Praharaka rises at the middle of Mercury’s part.

Since Kaala is upagraha of Jupiter so planet\_index should be 4

Yama Ghantaka rises at the middle of Jupiter’s part.

Since Gulika is upagraha of Saturn so planet\_index should be 6

Gulika rises at the middle of Saturn’s part.

Since Maandi is upagraha of Saturn so planet\_index should be 6

Maandi rises at the beginning of Saturn’s part.

You can use specific lambda functions for these.

@param as\_string = return results as string or list. True or False

@return: [constellation of upagraha,upagraha\_longitude within constellation]

## special\_ascendant(jd, place, time\_of\_birth\_in\_hours, lagna\_rate\_factor=1.0, divisional\_chart\_factor=1, as\_string=False)

Get constellation and longitude of special lagnas (Bhava Lagna, Hora Lagna and Ghato Lagna)

For Sree Lagna use sree\_lagna function

@param jd: Julian day number

@param place: Struct ('place name',latitude,longitude,time zone)

@param time\_of\_birth\_in\_hours: Time of birth in hours. example 5:37 PM = 17.37

@param lagna\_rate\_factor:

lagna\_rate\_factor = 1 for Bhava Lagna

= 0.5 for Hora Lagna

= 5/4 for Ghati Lagna

@param divisional\_chart\_factor: divisional chart factor

divisional\_chart\_factor = 2 => Hora, 3=>Drekana 4=>Chaturthamsa 5=>Panchamsa, 6=>Shashthamsa

7=>Saptamsa, 8=>Ashtamsa, 9=>Navamsa, 10=>Dasamsa, 11=>Rudramsa, 12=>Dwadamsa, 16=>Shodamsa,

20=>Vimsamsa, 24=>Chaturvimsamsa, 27=>Nakshatramsa, 30=>Trisamsa, 40=>Khavedamsa,

45=>Akshavedamsa, 60=>Shastyamsa

@return: [special lagnas constellation, special lagna's longitude within constellation]

## sree\_lagna (jd, place, divisional\_chart\_factor=1, as\_string=True)

Get constellation and longitude of Sree Lagna

@param jd: Julian day number

@param place: Struct ('place name',latitude,longitude,time zone)

@param divisional\_chart\_factor: divisional chart factor

divisional\_chart\_factor = 2 => Hora, 3=>Drekana 4=>Chaturthamsa 5=>Panchamsa, 6=>Shashthamsa

7=>Saptamsa, 8=>Ashtamsa, 9=>Navamsa, 10=>Dasamsa, 11=>Rudramsa, 12=>Dwadamsa, 16=>Shodamsa,

20=>Vimsamsa, 24=>Chaturvimsamsa, 27=>Nakshatramsa, 30=>Trisamsa, 40=>Khavedamsa,

45=>Akshavedamsa, 60=>Shastyamsa

@return: [Sree lagna constellation, Sree lagna's longitude within constellation]