# GeodePy

To import a package from GeodePy, you must enter it as follows:

import geodepy.package

where “package” is replaced with the package name.

## geodepy.constants

['Ellipsoid', 'Projection', 'Transformation', '\_\_builtins\_\_', '\_\_cached\_\_', '\_\_doc\_\_', '\_\_file\_\_', '\_\_loader\_\_', '\_\_name\_\_', '\_\_package\_\_', '\_\_spec\_\_', 'agd66togda94', 'agd66togda94\_act', 'agd66togda94\_nt', 'agd66togda94\_tas', 'agd66togda94\_vicnsw', 'agd84togda94', 'ans', 'c\_vac', 'gda94to20', 'grs80', 'iers2trans', 'itrf00to88', 'itrf00to89', 'itrf00to90', 'itrf00to91', 'itrf00to92', 'itrf00to93', 'itrf00to94', 'itrf00to96', 'itrf00to97', 'itrf00togda94', 'itrf05to00', 'itrf05togda94', 'itrf08to00', 'itrf08to05', 'itrf08to88', 'itrf08to89', 'itrf08to90', 'itrf08to91', 'itrf08to92', 'itrf08to93', 'itrf08to94', 'itrf08to96', 'itrf08to97', 'itrf08togda94', 'itrf14to00', 'itrf14to05', 'itrf14to08', 'itrf14to88', 'itrf14to89', 'itrf14to90', 'itrf14to91', 'itrf14to92', 'itrf14to93', 'itrf14to94', 'itrf14to96', 'itrf14to97', 'itrf14togda20', 'itrf96togda94', 'itrf97togda94', 'k\_0', 'sqrt', 'utm']

### Classes

#### x = geodepy.constants.Ellipsoid(semimaj, inversef)

Input:

* semimaj = Ellipsoid semi-major axis
* inversef = Inverse flattening (inverse relationship between semi-major and semi-minor axes)

Output:

* x.f = Flattening (reciprocal of inverse flattening)
* x.semimin = Ellipsoid semi-minor axis
* x.ecc1sq = Eccentricity squared
* x.ecc2sq = Second eccentricity squared
* x.ecc1 = Eccentricity
* x.n = Third flattening
* x.n2 = Third flattening squared

#### x = geodepy.constants.Projection(falseeast, falsenorth, cmscale, zonewidth, initialcm)

Input:

* falseeast = False easting value (500000)
* falsenorth = False northing value (1000000)
* cmscale = scale factor at the central meridian (0.9996)
* zonewidth = width of a zone (6)

## geodepy.convert

['Angle', '\_\_builtins\_\_', '\_\_cached\_\_', '\_\_doc\_\_', '\_\_file\_\_', '\_\_loader\_\_', '\_\_name\_\_', '\_\_package\_\_', '\_\_spec\_\_', 'dd2dms', 'dd2dms\_v', 'dec2hp', 'dec2sex', 'dms2dd', 'dms2dd\_v', 'hp2dec', 'hp2sex', 'modf', 'sex2dec', 'sex2hp']

### geodepy.convert.dec2hp(lon, lat)

Converts coordinates in decimal degrees to degrees, minutes and seconds in hp notation “ddd.mmssss”.

Input: float

Output: float

### geodepy.convert.hp2dec(lon, lat)

Converts coordinates in hp notation “ddd.mmssss” to decimal degrees.

Input: float

Output: float

### geodepy.convert.dec2sex(lon, lat)

Converts coordinates in decimal degrees to a string of degrees, minutes and seconds in sexagesimal format “dd mm ss.ss”.

Input: float

Output: string

### geodepy.convert.sex2dec(lon, lat)

Converts coordinates in sexagesimal format “ddd mm ss.ss” to decimal degrees. Input must be as strings.

Input: string

Output: float

### geodepy.convert.hp2sex(lon, lat)

Converts coordinates in hp notation “ddd.mmssss” to sexagesimal format “ddd mm ss.ss”.

Input: float

Output: string

NOTE: Currently does not work.

### geodepy.convert.sex2hp(lon, lat)

Converts coordinates in sexagesimals format “ddd mm ss.ss” to hp notation “ddd.mmssss”.

Input: string

Output: float

### geodepy.convert.dd2dms\_v(dd)

TypeError: 'float' object is not subscriptable

Looks as though it should do the same as “geodepy.convert.dd2dms(dd)”.

### geodepy.convert.dd2dms(dd)

Converts a single decimal degree measurement into degrees, minutes and seconds in hp notation “ddd.mmssss”.

Input: float

Output: float

### geodepy.convert.dms2dd\_v(dms)

TypeError: 'float' object is not subscriptable

Looks as though it should do the same as “geodepy.convert.dms2dd(dms)”.

### geodepy.convert.dms2dd(dms)

Converts a single degrees, minutes, seconds measurement in hp notation “ddd.mmssss” to decimal degrees.

Input: float

Output: float

### geodepy.convert.read\_dnacoord(fn)

Not callable??