



ÉCOLE NATIONALE
DES SCIENCES
GÉOGRAPHIQUES

GpsDateTime

Package python de gestion des temps dans les calculs GNSS

jacques.beilin@ensg.eu

Les échelles de temps GNSS

- Début de l'ère GPS : 6 janvier 1980
- Nombreuses échelles de temps utilisées dans les données et produits :
 - année / jour dans l'année
 - semaine GPS / jour dans la semaine
 - secondes de jour
 - secondes dans la semaine
 - année décimale
 - jour julien / jour julien modifié
 - ...

Les échelles de temps GNSS

Questions auxquelles on doit répondre

- Quelle jour tombe le jour 0 de la semaine 2208 ?
- Quel semaine/jour dans la semaine tombe le 31 décembre 2022 ?
- Quelle durée y a-t-il entre le 31 décembre 2022 23h59min59.997s et le 1er janvier 2023 0h0min0.12s ?
- Quelle instant obtient-on en ajoutant ou soustrayant 63ms à une date ?
- Parmi deux dates/heures, laquelle précède l'autre ?
- Question subsidiaire : l'heure sidérale Greenwich pour un instant t donné ?

Attributs disponibles

Champs	Description	Unité
s1970	secondes depuis 01-01-1970 à 0h00Z	seconde
mjd	Modified Julian Date (=JD-2400000.5)	jours décimaux
jd	Julian Date	jours décimaux
jd50	Julian Date depuis J1950	jours décimaux
wk	Semaine GPS	semaine I4
wsec	seconde dans la semaine GPS	float [0..604800]
yyyy	année sur 4 chiffres	année I4 [1902..2079]
yy	année sur 2 chiffres	année (I2)
mon	mois de 1 à 12	I2 [0..12]
dd	jour dans le mois	I2
hh	heure	I2 [0..24]
min	minute	I2 [0..60]
sec	seconde	I2 [0..60]
doy	jour dans l'année	I3 [1..366]
wd	jour dans la semaine	I1 [0..6]
dsec	seconde dans la journée	float [0..86400]
dy	année décimale	float [0..366]
GMST	Greenwich Mean Sidereal Time	heures décimales
EQEQ	Equation des Equinoxes	heures décimales
GAST	Greenwich Apparent Sidereal Time	heures décimales

Instantiation par défaut

```
1 Python 3.9
>>> import gpsdatetime as gpst
3 >>> t = gpst.gpsdatetime()
```

Objet t instancié avec l'heure système

```
1 Python 3.9
>>> import gpsdatetime as gpst
3 >>> t = gpst.gpsdatetime()
>>> ... some bunch of code ...
5 >>> t.just_now()
```

L'objet t est instancié à l'heure système puis réinitialisé à l'heure de la commande just_now

Instanciation puis initialisation 1/2

- `ymdhms_t` : année, mois, jour, heure, minute, seconde.

```
1 >>> t.ymdhms_t(2012,1,9,12,0,0)
```

- `yyyymmdd_t` : année, jours dans l'année (DOY), seconde de jour.

```
1 >>> t.yyyymmdd_t(2012,9,43200)
```

- `gpswkd_t` : semaine GPS, jour dans la semaine [0..6]. La partie fractionnaire représente un jour. décimal.

```
1 >>> t.gpswkd_t(1400,1.5)
```

- `gpswks_t` : semaine GPS, seconde dans la semaine.

```
1 >>> t.gpswks_t(1400,86400.5)
```

- `mjd_t` : Modified Julian Date (décimal).

```
1 >>> t.mjd_t(54800.5)
```

Instanciación puis initialisation 2/2

- `jd_t` : Julian Date (décimal).

```
1 >>> t.jd_t(2054800.5)
```

- `iso_t` : date au format ISO.

```
1 >>> t.iso_t('2017-10-09T12:00:00Z')
```

- `rinex_t` : date au format Rinex (fichiers d'observations).

```
1 >>> t.rinex_t('17 6 13 7 35 0.0000000')
```

- `snx_t` : date au format Sinex (chaîne de caractères comprenant l'année à 2 chiffres, le jour dans l'année [0..366] et la seconde de jour [0..86400]).

```
1 >>> t.snx_t('17:204:43200')
```

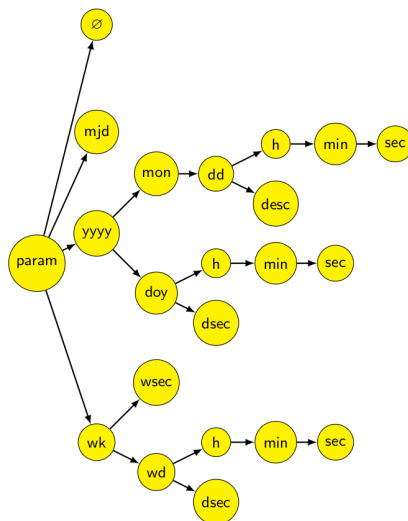
Instantiation directe 1/3

- `t=gpsdatetime()` : si on ne passe aucun paramètre (où si les paramètres ci-dessous sont passés de manière incorrecte), alors l'objet `gpsdatetime` est instancié avec la date système de la machine à l'heure UTC.
- `t=gpsdatetime(mjd=54605.678)` : instantiation à partir d'une `mjd`.
- `t=gpsdatetime(yyyy=2016, doy=004, dsec=45677)` : l'objet est instancié à `yyyy/doy`. les secondes dans la journée sont ajoutées ensuite. Ce paramètre est optionnel.
- `t=gpsdatetime(wk=1400, wsec=600700)` : L'objet est dans un premier temps initialisé au début de la semaine 1400. Les secondes dans la semaine sont ajoutés ensuite. Ce paramètre est optionnel.

Instantiation directe 2/3

- `t=gpsdatetime(wk=1400, wd=1, dsec=46888)` : L'objet est dans un premier temps initialisé au début de la semaine 1400. Le jour puis les secondes dans le jour sont ajoutés successivement. Ces paramètres sont optionnels mais les secondes dans la journées ne sont prises en compte que si le jour dans la semaine est fourni.
- `t=gpsdatetime(yyyy=2016, mon=1, dd=7, hh=3, min=5, sec=5)` : l'objet est instancié à `yyyy/mon/day`. Les heures, minutes et secondes sont optionnels.
- `t2=gpsdatetime(t1)`, `t1` étant lui-même un objet `gpsdatetime` : on instancie une copie `t2` de l'objet `t1`.
- `t=gpsdatetime('16:004:46888')` : instantiation à partir d'une date SINEX.
- `t=gpsdatetime('16:01:04T03:05:05Z')` : instantiation à partir d'une date iso.

Instanciación directe 3/3



Initialisation en début de période

- wk00 : calcul du début de la semaine

```
1 >>> t = gpst.gpsdatetime()  
>>> t.wk00()
```

- day00 : calcul du début du jour

```
>>> t.day00()
```

- h00 : calcul du début de l'heure

```
1 >>> t.h00()
```

- m00 : calcul du début de la minute

```
1 >>> t.m00()
```

Récupération d'attributs / chaînes formatées

```
1 Python 3.9
>>> import gpsdatetime as gpst
3 >>> t = gpst.gpsdatetime()
>>> m = t.mjd
```

Nom de la fonction	Type de format	Format
st_iso_epoch()	Format iso	yyyy-mm-ddThh:mm:ssZ
st_pyephem_epoch()	Format d'entrée du package pyephem	yyyy/mm/ddhh:mm:ss.s
st_snx_epoch()	Format Sinex	yy:ddd:sssss

Impressions complètes des attributs

```
>>> print(t)
Gpsdatetime (version 2016-06-27)
s1970 : 1480272707.601859
YYYY_MM_DD : 2016/11/27
HH:MM:SS : 18:51:47.601858854
GPS week : 1925
Day of week : 0 (SUN)
Second of week : 67907.601858854
Second of day : 67907.601858854
session : s
Modified Julian Date : 57719.785968
Julian Date : 2457720.285968
YYYY : 2016 DOY : 332
GMST (dec. hour) : 23.337554
GAST (dec. hour) : 23.337429
Eq. of Equinoxes (dec. hour) : -0.000125
```

Ajout/soustraction de durées

- Surcharge d'opérateur

```
>>> t += 5.0 # ajout de 5s  
2 >>> t -= 2.0 # soustraction de 2s
```

- add_day : ajout/soustraction de jours décimaux

```
>>> t.add_day(1.5)
```

- add_h : ajout/soustraction d'heures décimales

```
1 >>> t.add_h(0.5)
```

- add_s : ajout/soustraction de secondes décimales

```
1 >>> t.add_s(-0.5)
```

Durées et comparaisons de dates

- Calcul de durée entre 2 objets Gpsdatetime

```
1 >>> t1 = gpst.gpsdatetime()  
>>> t2 = gpst.gpsdatetime()  
3 >>> Delta_t = t2 - t1 # resultat en secondes
```

- Comparaison de 2 objets Gpsdatetime

```
1 >>> t1 = gpst.gpsdatetime()  
>>> t2 = gpst.gpsdatetime()  
3 >>> if t1 < t2:  
>>> ...print("t1 est avant t2")
```