

## Trame NMEA

### Compétences visées

Un des objectifs de l'enseignement de SNT est de comprendre le principe de fonctionnement d'appareils de type GPS, afin de les envisager autrement qu'une simple "boîte noire" qui donne des résultats.

Le programme de SNT propose dans la partie Localisation, "*cartographie et mobilité*" de travailler sur "*les récepteurs GPS [qui] fournissent la localisation sous une forme normalisée facilement décodable, par exemple selon le protocole NMEA 0183 (National Marine Electronics Association), ou directement dans les métadonnées EXIF d'une photo.*"

On propose dans cette activité de construire plusieurs fonctions au sein d'un programme qui aura importé des données.

Ceci est à mettre en lien avec la partie "*Les données structurées et leur traitement*" avec un script qui a comme objectif de fournir des données et un autre qui réalisera le traitement de ces données.

Le script Python mis en jeu fait appel à des fonctions et des techniques en lien avec les chaînes de caractères.

### Situation déclenchante

Les systèmes de localisation par satellites (américains, européens, chinois et russes) envoient des données codées selon divers protocoles.

Comment décoder ces données et pouvoir ainsi les interpréter ?



### Problématique

Une trame NMEA 0183 nous étant donné, créer des scripts en Python permettant de :

- Déterminer le système de positionnement par satellite à l'origine de cette trame ;
- Déterminer l'heure correspondant à cette trame ;
- Déterminer la longitude d'une part, la latitude d'autre part du point correspondant.

## Fiche méthode

### Déroulement possible du projet

#### En amont du projet :

Les propositions suivantes permettent d'intégrer l'activité dans le cadre d'une réflexion à mener autour du thème de la localisation par satellites.

- Un groupe d'élèves peut réaliser un travail montrant le principe de géolocalisation par trois satellites et expliquer pourquoi en pratique il en faut quatre.
- Un groupe d'élèves peut présenter un historique du système de positionnement par satellite, en particulier du développement des systèmes GPS et Galileo.
- Un groupe d'élèves peut présenter les enjeux sous-jacents à la maîtrise d'un système de positionnement par satellites.
- Un groupe d'élèves peut par ailleurs chercher des fonctions Python pour traiter des chaînes de caractères.

#### Différentes évolutions possibles pour le projet :

Les propositions suivantes donnent des pistes pour gérer les différences de vitesse d'exécution des élèves.

On peut aussi considérer ces différentes étapes comme une manière de séquencer le projet pour ceux qui en auraient besoin.

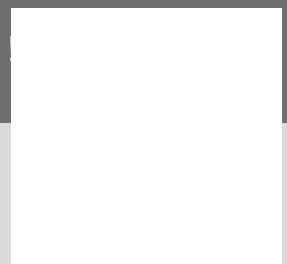
- 1<sup>ère</sup> étape : L'enseignant donne un script à l'ensemble de la classe contenant plusieurs trames NMEA nommées t1, t2 ...etc. La méthode la plus efficace sera un transfert de calculatrice à calculatrice du script contenant ces données. Les élèves créent tous un nouveau script (qui traitera ces données) qui commence par l'importation de ces données.
- 2<sup>ème</sup> étape : Une fonction permettant de retourner le type de système de positionnement par satellite est créée en classe ; l'ensemble des élèves peut chercher une fonction qui retournera l'heure à laquelle ce positionnement a été effectué.
- 3<sup>ème</sup> étape : D'autres fonctions peuvent être créées par les élèves plus à l'aise, donnant la longitude et la latitude. On peut aussi donner l'une des deux et demander aux élèves de trouver l'autre, ou alors de donner ces fonctions en partie et laisser les élèves les compléter.

#### A la suite du projet :

Synthétiser ce travail par rapport au cours de SNT :

- Est-il courant de séparer les données et leur traitement (des exemples) ?
- Existe-t-il d'autres types de trames liées au système de positionnement par satellites ?
- Quel est l'intérêt d'avoir un protocole tel que le protocole NMEA ? Par qui est-il fréquemment utilisé ?

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



## Fiche méthode

### Proposition de résolutions

- L'enseignant (ou des élèves) peut présenter des techniques liées aux chaînes de caractères, en particulier les manières d'extraire des éléments d'une chaîne de caractères par [ : ].
- On donne aux élèves le script **TDONNEES** qui contient plusieurs trames saisies par l'enseignant (la saisie peut se faire sur ordinateur et le script peut être transféré sur calculatrice via le logiciel *TI Connect*).
- On remarquera que, pour le type de système de positionnement comme pour l'heure, les éléments sont toujours positionnés à la même place dans les trames NMEA, ce qui n'est pas le cas pour les valeurs de longitude et latitude (en effet, selon les valeurs numériques codées pour l'heure, les valeurs données pour longitude et latitude ne seront pas exactement à la même place dans la chaîne de caractères de la trame).
- Le traitement du type de système et de l'heure d'une part, et de la longitude et de la latitude d'autre part ne seront pas tout à fait du même ordre, demandant des compétences différentes. Cela peut permettre de différencier le travail au sein du groupe classe.

```
ÉDITEUR : TDONNEES
LIGNE DU SCRIPT 0002
t1 = "$GPGGA,095039.00,4910.7631
94,N,00022.968045,W,1"
t2 = "$GPGGA,064036.289,4836.537
5,N,00740.9373,E,1"
```

### Etapes de résolution

L'importation du script **TDONNEES** peut être facilité en allant chercher **from SCRIPT import \*** dans le catalogue. Pour cela :

- Touche **2nde** suivie de la touche **0**
- Touche **alpha** suivie de **resol** pour obtenir la lettre F
- Descendre avec les flèches directionnelles.

Ne reste plus qu'à saisir lettre à lettre **TDONNEES**.

```
ÉDITEUR : TRAME2
LIGNE DU SCRIPT 0001
from TDONNEES import *
```

La fonction **sat** prend comme argument une chaîne de caractères qui représente une trame ; on en extrait les caractères situés aux positions n°1 et n°2 de la chaîne (le premier élément de la chaîne est à la position 0) :

- Si ces caractères sont **GP**, on retourne comme système GPS ;
- Si ces caractères sont **GL**, on retourne comme système GLONASS ;
- Si ces caractères sont **GA**, on retourne comme système Galileo ;
- Si ces caractères sont **GD** ou **GB**, on retourne comme système Beidou.

```
ÉDITEUR : TRAME2
LIGNE DU SCRIPT 0022
def sat(tr):
    s = tr[1:3]
    if s == "GP":
        sys = "GPS"
    elif s == "GL":
        sys = "GLONASS"
    elif s == "GA":
        sys = "Galileo"
    elif s == "GD" or s=="GB":
        sys = "Beidou"
    return s, sys
```

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



## Fiche méthode

La fonction **h** prend comme paramètre une chaîne de caractères correspondant à une trame NMEA :

- les caractères situés en position 7 et 8 sont ceux de l'heure ;
- les caractères situés en position 9 et 10 sont ceux des minutes ;
- les caractères situés en position 11 et 12 sont ceux des secondes ;
- les autres caractères qui seraient éventuellement présents ne sont pas pris en compte ici.

On retourne une chaîne de caractères comprenant ces données ainsi que les caractères " h ", " mn " et " s " afin de rendre le résultat lisible.

Les caractères sont concaténés à l'aide du symbole +

```
EDITEUR : TRAME2
LIGNE DU SCRIPT 0018
def h(tr):
    *heure = tr[7:9]+" h "+tr[9:11]
    *   +" min "+tr[11:13]+" s"
    *return heure
```

```
EDITEUR : TRAME2
LIGNE DU SCRIPT 0021
def separe(ch):
    *return ch.split(",")
```

```
ÉDITEUR : TRAME2
LIGNE DU SCRIPT 0032
def lat(tr):
    *liste = separe(tr)
    *val = float(liste[2])
    *deg = int(val/100)
    *min = val%100
    *if liste[3] == "N":
    *   *return deg + min/60
    *else :
    *   *return -deg - min/60
```

```
ÉDITEUR : TRAME2
LIGNE DU SCRIPT 0042
def long(tr):
    *liste = separe(tr)
    *val = float(liste[4])
    *deg = int(val/100)
    *min = val%100
    *if liste[5] == "E":
    *   *return deg + min/60
    *else :
    *   *return -deg - min/60
```

## Pour aller plus loin

Pour travailler sur la longitude et la latitude, il va falloir distinguer les éléments de la trame NMEA séparés par des virgules.

Pour cela, on utilise la méthode **.split** (méthode native à python) qui s'applique à une chaîne de caractères, en indiquant comme paramètre l'élément séparateur.

Cette méthode renvoie une liste dont les éléments sont différentes chaînes de caractères.

Ainsi, `separe("$GPGGA,095039.00,4910.763194,N,00022.968045,W,1")` sera la liste `["$GPGGA", "095039.00", "4910.763194", "N", "00022.968045", "W", "1"]` ; les éléments situés en position 2 et 3 permettront de déterminer la latitude et les éléments situés en position 4 et 5 permettront de déterminer la longitude.

- La partie utile de la trame NMEA pour déterminer la latitude est la chaîne située en position n°2 : on la stocke dans la variable **val** ; **liste[2]** étant une chaîne de caractères, on force le type en indiquant **float** pour que cette chaîne devienne un nombre.
- Il s'agit ensuite de convertir des valeurs données en *degrés / minutes* en grandeur décimales. C'est l'objet des variables **deg** et **min**.
- Il faut ensuite tenir compte de la position Nord ou Sud indiquée dans la trame par **N** ou **S** ; selon le résultat, le résultat donné sera positif ou négatif, selon les conventions habituelles.

Le travail à réaliser pour déterminer la longitude est évidemment très similaire à celui sur la latitude.

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus

