
	<b>SNT – Thème 7 – Localisation et cartographie</b>	
	<b>Document Élève - Activité 2 Groupe B</b>	
	<b>Principe de la localisation</b>	1/7

## À LA RECHERCHE DU PARCHEMIN PERDU

Une rumeur court depuis longtemps en Touraine : Léonard de Vinci aurait confié ses plus précieux parchemins à un monastère. En effet ils contiennent des plans de ses inventions les plus brillantes, mais il ne souhaite pas qu'elles tombent entre de mauvaises mains. Mais au fil des siècles, le monastère se vida peu à peu de ses moines et il ferma ses portes en 2010. Le dernier moine décida alors de cacher les manuscrits de Léonard et dissémina quelques indices pour que les personnes initiées puissent le retrouver. Votre équipe a pu retrouver ce texte mystérieux :



***Quelque part non loin de [AB], proche  
de [CF], tout près de [DH] et à  
proximité de [EG] se tient le lieu,  
gardien du secret.***

Vous avez également trouvé un fichier texte (voir document fourni) contenant des données qui pourraient provenir d'un récepteur GPS, mais quel charabia !!

Manifestement, le peu d'information à votre disposition ne va pas permettre de retrouver ce trésor historique, il sera peut-être utile de collaborer avec d'autres équipes ...

La page suivante contient des informations qui peuvent permettre de démêler cette énigme.

Noter les étapes de la démarche envisagée :

	<b>SNT – Thème 7 – Localisation et cartographie</b>	
	<b>Document Élève - Activité 2 Groupe B</b>	
	<b>Principe de la localisation</b>	2/7

## DOCUMENTS

### Document 1 : les trames NMEA

Il existe plusieurs protocoles de communication pour les récepteurs GPS. Le protocole NMEA 0183 est celui qui est utilisé par tous les récepteurs grand public (smartphones, boîtiers GPS).

NMEA 0183 utilise les caractères ASCII qui forment des « trames ». Les trames sont définies par un format normalisé. Attention, ces trames n'ont rien à voir avec les informations envoyées par les satellites. Elles sont générées par le récepteur GPS à partir des informations reçues depuis les satellites et partagées ensuite aux applications nécessitant une localisation (carte numérique, réseaux social, ...).

Chaque trame commence par le caractère '\$' (ascii hexa 0x24) et finit par les caractères retour chariot et saut de ligne (ascii hexa 0x0D et 0x0A).

La trame est ensuite constituée d'un identifiant de trame suivi des champs de valeurs séparées par des virgules. Après le dernier champ et avant les caractères de fin de ligne, on trouve le caractère '\*' suivi d'un checksum (« somme de contrôle » : permet de vérifier l'intégrité de la transmission).

L'identifiant est constitué de 2 lettres indiquant la source du signal (BD ou GB pour Beidou, GA pour Galileo, GP pour GPS et GL pour GLONASS) suivi d'un code de 3 lettres indiquant le type de trame. Il existe plus de 70 types différents. Les plus intéressantes pour déterminer sa position au sol sont les **GGA, RMC et GLL**.

### Document 2 : extrait d'une documentation sur les types de trame du protocole NMEA

**RMC** - NMEA has its own version of essential gps pvt (position, velocity, time) data. It is called RMC, The Recommended Minimum, which will look similar to:

```
$GPRMC,123519,A,4807.038,N,01131.000,E,022.4,084.4,230394,003.1,W*6A
```

Where:

RMC	Recommended Minimum sentence C
123519	Fix taken at 12:35:19 UTC
A	Status A=active or V=Void.
4807.038,N	Latitude 48 deg 07.038' N
01131.000,E	Longitude 11 deg 31.000' E
022.4	Speed over the ground in knots
084.4	Track angle in degrees True
230394	Date - 23rd of March 1994
003.1,W	Magnetic Variation
*6A	The checksum data, always begins with *

### Document 3 : le système sexagésimal

Les latitudes et les longitudes peuvent être exprimées de différentes façons. Les cartes numériques acceptent des coordonnées en « degré décimal » (exemple : 53,4567 °). On peut également utiliser le système « sexagésimal », soit exprimer un angle en degré, minute d'arc (1/60 de degré) (exemple : 53° 27,402')

Formulation générale :

degrés décimaux = degrés + (minutes / 60)

### Document 4 : script python

```
import folium # module permettant de manipuler une carte numérique
import webbrowser # module permettant d'ouvrir une page HTML

I = (48.873734 , 2.295046) # Définition d'un point I (latitude,longitude)
J = (48.865435 , 2.321181) # Définition d'un point J (latitude,longitude)



m = folium.Map(location=I, zoom_start=14) # création de la carte centrée sur le point I

folium.Marker(I, popup = 'I').add_to(m) # positionnement d'un marqueur en I
folium.Marker(J, popup = 'J').add_to(m) # positionnement d'un marqueur en J



folium.vector_layers.PolyLine([I,J]).add_to(m) # Tracé du segment [I,J]

m.save('themap.html') # sauvegarde de la carte au format HTML

webbrowser.open('themap.html') # Ouverture de la carte dans le navigateur
```

	<b>SNT – Thème 7 – Localisation et cartographie</b>	
	<b>Document Élève - Activité 2 Groupe B</b>	
	<b>Principe de la localisation</b>	3/7

Notez ici les calculs, données, codes python envisagés :



	<b>SNT – Thème 7 – Localisation et cartographie</b>	
	<b>Document Élève - Activité 2 Groupe B</b>	
	<b>Principe de la localisation</b>	4/7

### Contenu de B.txt

```

$GLGSA,A,1,,,,,,,,,,,,,100.000,100.000,100.000*2D
$BDGSA,A,1,,,,,,,,,,,,,100.000,100.000,100.000*20
$QZGSA,A,1,,,,,,,,,,,,,100.000,100.000,100.000*2D
$GPGSV,6,1,22,1,19,141,,2,7,314,,3,50,77,,4,19,144,*7B
$GPGSV,6,2,22,6,45,305,,9,56,209,,17,31,231,,19,38,261,*7F
$GPGSV,6,3,22,22,30,88,,23,79,128,,31,12,35,*4C
$GLGSV,6,4,22,65,30,259,,71,39,48,,72,70,313,,73,18,81,*6E
$GLGSV,6,5,22,80,13,23,,86,25,182,,87,63,257,,88,33,323,*57
$BDGSV,6,6,22,205,8,118,,207,6,56,,209,7,52,*54
$GPGGA,101316.49,4723.5995,N,00041.47692,E,0,,100.000,120.064,M,,M,0,*4A
$GNGSA,A,1,,,,,,,,,,,,,100.000,100.000,100.000*2F
$GPGSA,A,1,,,,,,,,,,,,,100.000,100.000,100.000*31
$GLGSA,A,1,,,,,,,,,,,,,100.000,100.000,100.000*2D

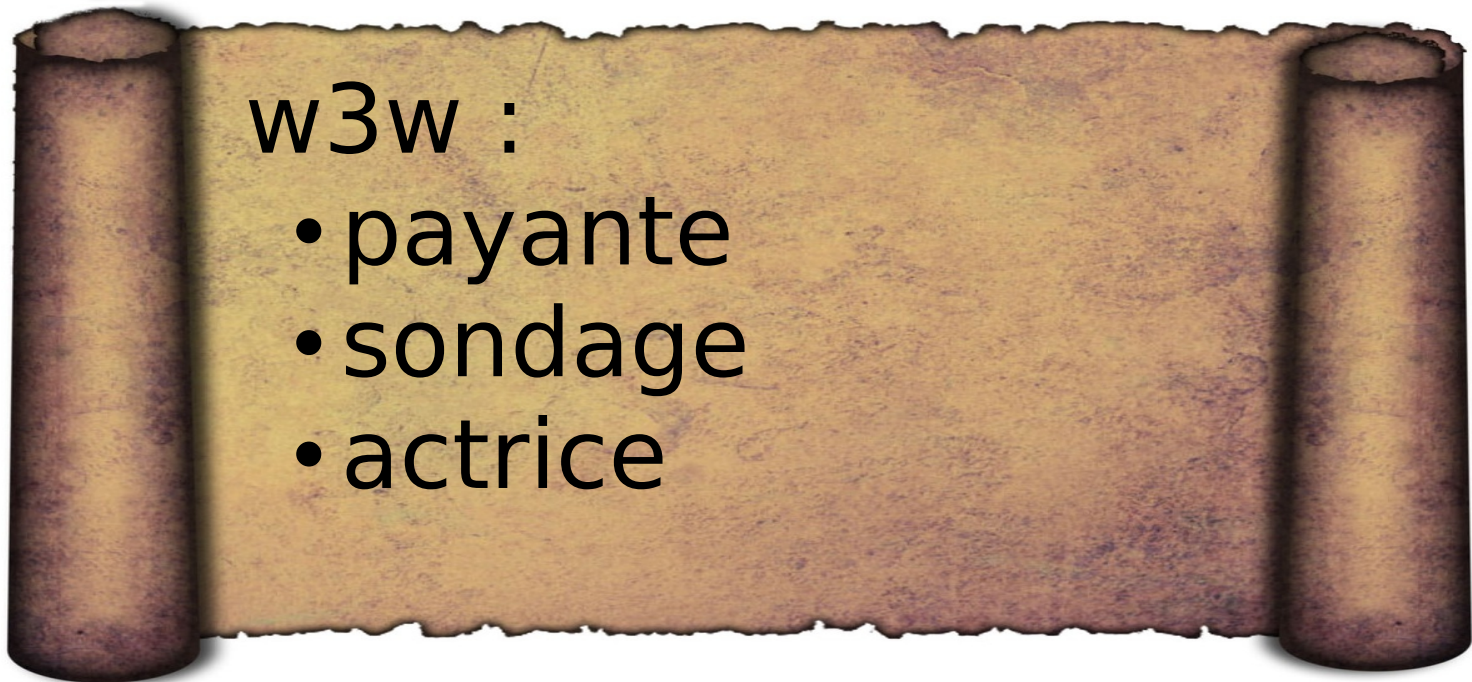
```

	<b>SNT – Thème 7 – Localisation et cartographie</b>	
	<b>Document Élève - Activité 2 Groupe B</b>	
	<b>Principe de la localisation</b>	5/7



## PROLONGEMENT

Félicitations, vous avez trouvé le bâtiment dans lequel sont cachés les documents du célèbre inventeur italien ! Cependant, le lieu est vaste, une plus grande précision ne serait pas du luxe !!

Au dos du papier du moine, on trouve une inscription encore plus étrange :



Comment peut-on interpréter cette nouvelle information ?

	<b>SNT – Thème 7 – Localisation et cartographie</b>	
	<b>Document Élève - Activité 2 Groupe B</b>	
	<b>Principe de la localisation</b>	6/7

## DÉCODAGE D'UNE TRAME NMEA PAR PROGRAMMATION

Quelques rappels de Python :

```
>>> ligne="DURANT;David;homme;17" # ligne est une chaîne de caractères
>>> infos=ligne.split(";") # la fonction split découpe une chaîne de
                             caractères

>>> infos
['DURANT', 'David', 'homme', '17'] # infos est une liste
>>> infos[1] # infos[1] est le deuxième élément de la liste
                             # on compte à partir de 0

'David'
>>> prenom=infos[1]
>>> prenom[2:4] # slicing : découpage de chaîne de caractères
'vi' # on compte toujours à partir de 0 !!
                             # mais le dernier index est exclu (le 4)
>>> prenom[3:] # on prend les derniers caractères à partir
'id' # de la position 3
>>> element[3]
'17' # element[3] est une chaîne de caractères
>>> age = int(element[3]) # transtypage en entier
>>> age
17 # age est un entier
>>> age = float(element[3]) # transtypage en nombre flottant
>>> age
17.0 # age est un nombre à virgule flottante (décimal)
```



On considère la trame suivante :

**\$GPGGA,123519,4852.42404,N,000217.70276,E,1,08,0.9,545.4,M,46.9,M,,\*47**

Vérifier par un calcul que la latitude indiquée par cette trame vaut 48.873734° :

Quelle instruction en python permet d'obtenir une liste à partir d'une chaîne de caractères nommée `trame` ?

```
trame = "$GPGGA,123519,4852.42404,N,000217.70276,E,1,08,0.9,545.4,M,46.9,M,,*47"
liste =
```

	<b>SNT – Thème 7 – Localisation et cartographie</b>	
	<b>Document Élève - Activité 2 Groupe B</b>	
	<b>Principe de la localisation</b>	7/7

Écrire l'instruction python qui va permettre de stocker dans la variable `lat` les caractères correspondant à la latitude.

`lat =`

Écrire l'instruction python qui va permettre de stocker dans la variable `lat_deg` la valeur du nombre de degrés de la latitude au format « float » :

`lat_deg =`

Écrire l'instruction python qui va permettre de stocker dans la variable `lat_min` les minutes de la latitude au format float en les convertissant en fraction décimale de degré.

`lat_min =`

Écrire l'instruction python qui va permettre de stocker dans la variable `lat_dec` la somme des deux précédentes variables pour obtenir la latitude au format décimal.

`lat_dec =`

Reproduire la même démarche pour la longitude pour obtenir `long`, `long_deg`, `long_min` et enfin `long_dec` (attention il y a 4 chiffres pour les degrés dans la trame) :

Taper l'instruction suivante à la suite de toutes les précédentes lignes :

`webbrowser.open('https://www.openstreetmap.org/#map=' + '16' + '/' + str (lat_dec)+ '/' + str (long_dec))`. Sur quel monument est centrée la carte qui s'affiche ?