

<p>Proposition d'activité</p> <p>Sciences numériques et technologie</p> <p>Classe de seconde, enseignement commun</p>

- Thèmes :
 - Localisation, cartographie et mobilité ;
 - Notions de programmation transversale.
- Prérequis des élèves
 - Variables, chaînes de caractères, listes, fonctions.
- Objectifs :
 - appréhender la problématique de la localisation, de la collecte et du traitement des données ;
 - faire preuve d'autonomie, d'initiative et de créativité ;
 - coopérer au sein d'une équipe ;
 - Écrire, exécuter et mettre au point un programme pour décoder une trame NMEA.
- Modalités pédagogiques :
 - activité en groupes ;
 - un ou deux élèves par poste informatique.

Contenus

Affectations, variables

Séquences

Instructions conditionnelles

Boucles bornées et non bornées

Définitions et appels de fonctions

Exemples d'activités

Illustrer ces notions par des activités liées aux différents thèmes du programme.

Protocole NMEA 0183

Capacités attendues

Écrire, exécuter et mettre au point un programme.

Décoder une trame NMEA pour trouver des coordonnées géographiques.

Décodage d'une trame NMEA

Tous les smartphones commercialisés ces dernières années sont dotés d'une puce GPS et c'est grâce à celle-ci que votre téléphone peut être géolocalisé, c'est-à-dire que vous pouvez connaître sa position géographique. L'objectif de cette activité est d'appréhender le fonctionnement de ce dispositif.

Document 1 : La norme NMEA 0183

Il existe plus d'une trentaine de trames GPS différentes. Le type d'équipement est défini par les deux caractères qui suivent le \$. Le type de trame est défini par les caractères suivants jusqu'à la virgule. Par exemple :

\$GPGGA,064036.289,4836.5375,N,00740.9373,E,1,04,3.2,200.2,M,,0000*0E est une trame GPS de type GGA.

Les deux premiers caractères après le signe \$ (talker id) identifient l'origine du signal. Les principaux préfixes sont :

BD ou GB - Beidou ;

GA - Galileo ;

GP - GPS ;

GL - GLONASS.

Source : https://fr.wikipedia.org/wiki/NMEA_0183

Document 2 : La trame GGA

La trame GGA est très courante car elle fait partie de celles qui sont utilisées pour connaître la position courante du récepteur GPS.

\$GPGGA	: Type de trame
064036.289	: Trame envoyée à 06h 40m 36,289s (heure UTC)
4836.5375, N	: Latitude Nord : $48^{\circ}36.5375'$ (DM) = $48,608958^{\circ}$ (DD) = $48^{\circ}36'32.25''$ (DMS)
00740.9373,E	: Longitude Est : $7^{\circ}40.9373'$ (DM) = $7,682288^{\circ}$ (DD) = $7^{\circ}40'56.238''$ (DMS)
1	: Type de positionnement (le 1 est un positionnement GPS)
04	: Nombre de satellites utilisés pour calculer les coordonnées
3.2	: Précision horizontale ou HDOP (Horizontal dilution of precision)
200.2,M	: Altitude 200,2, en mètres
,,,,0000	: D'autres informations peuvent être inscrites dans ces champs
*0E	: Somme de contrôle de parité, un simple XOR sur les caractères entre \$ et *3

Source : d'après https://fr.wikipedia.org/wiki/NMEA_0183

Document 3 : Les notations DMS, DM et DD

Les coordonnées géographiques sont traditionnellement exprimées dans le système sexagésimal, parfois noté « DMS » : degrés (°) minutes (') secondes ("). L'unité de base est le degré d'angle (1 tour complet = 360°), puis la minute d'angle (1° = 60'), puis la seconde d'angle (1° = 3 600"). Ces distances correspondant à un écart de longitude (en degré, minute ou seconde), varient selon la latitude du lieu, puisque les méridiens terrestres se rapprochent progressivement depuis l'équateur vers les pôles. Le tableau ci-dessous en donne quelques exemples illustratifs

Longueurs des écarts de longitude à différentes latitudes

Latitude	Ville	Degré	Minute	Seconde	±0.0001°
59° 56' 02"	Saint-Pétersbourg	55,80 km	0,930 km	15,50 m	5,58 m
51° 28' 38" N	Greenwich	69,47 km	1,158 km	19,30 m	6,95 m
44° 50' 16"	Bordeaux	78,85 km	1,31 km	21,90 m	7,89 m
29° 58'	Nouvelle-Orléans	96,49 km	1,61 km	26,80 m	9,65 m
0° 15' 00"	Quito	111,3 km	1,855 km	30,92 m	11,13 m

De nos jours, les notations équivalentes en minutes décimales ou degrés décimaux sont également utilisées :

DMS : Degré:Minute:Seconde (49° 30' 00" - 123° 30' 00") ;

DM : Degré:Minute (49° 30,0' - 123° 30,0') ;

DD : Degré décimal (49,5000° - 123,5000°), généralement avec quatre décimales.

Source : d'après https://fr.wikipedia.org/wiki/Coordonnées_géographiques

Document 4 : rappels

```
>>> ligne="nom;prénom;age;17" # ligne est une chaine de caracteres
>>> element=ligne.split(";")   # element est une liste
>>> element
['nom', 'prénom', 'age', '17']
>>> element[1]                 # element[1] est le deuxieme element de la liste
'prénom'
>>> prenom=element[1]
>>> prenom[2:4]                # slicing
'én'
>>> prenom[4:]
'om'
>>> element[3]                 # element[3] est une chaine de caracteres
'17'
>>> int(element[3])            # transtypage en entier
17
>>> float(element[3])          # transtypage en nombre flottant
17.0
```

Questions :

1. Vérifier par un calcul que la latitude $48^{\circ}36.5375'$ (DM) du document correspond à $48^{\circ}36'32.25''$ (DMS).
2. D'après le document 4, quelle instruction en python permet d'obtenir une liste à partir d'une chaîne de caractères nommée trame ?
trame="\$GPGGA,064036.289,4836.5375,N,00740.9373,E,1,04,3.2,200.2,M,,,0000*0E"
liste=
3. Écrire une procédure origine(type_trame) qui à partir du premier élément de la trame affiche l'origine du signal.
Application :
>>> origine('\$GPGGA')
C'est une trame GPS
4. Écrire un programme qui à partir d'une trame **NMEA** affiche l'heure, la latitude, et la longitude sous la forme :
heure : hh : mm : ss
latitude : 48 ° 36 ' 32 " N
longitude : 07 ° 40 ' 56 " E
Appliquer ce programme à la trame suivante :
trame="\$GPGGA,094527.289,47.6316,N,6.8503,E,1,04,3.2,200.2,M,,,0000*0E" .
5. Écrire une procédure affichage(zoom,latitude, longitude) à l'aide des instructions ci-dessous pour obtenir l'affichage du lieu correspondant à l'aide site Openstreetmap.
import webbrowser
zoom='16'
webbrowser.open('https://www.openstreetmap.org/#map='+zoom+'/' +str(latitude)
+'/' +str(longitude))

Éléments de correction

Fonction origine(type_trame) :

```
1 # prerequisites
2 # typage str, les chaines de caracteres slicing fonctions
3 # les structures conditionnelles
4
5 #trame="$GPGGA,064036.289,4836.5375,N,00740.9373,E,1,04,3.2,200.2,M,,,0000*0
6 trame="$GPGGA"
7 def origine(data):
8     """ retourne le type de trame"""
9     type_trame=data[1:3]
10    if type_trame=="GD" or type_trame=="GB":
11        type_signal="Beidou"
12    elif type_trame=="GA":
13        type_signal="Galileo"
14    elif type_trame=="GP":
15        type_signal="GPS"
16    elif type_trame=="GL":
17        type_signal="GLONASS"
18    return type_signal
19
20 type_signal=origine(trame)
21 print ('Le signal est du type',type_signal)
résultat :
Le signal est du type GPS
```

programme décodage d'une trame

```
1 # prerequisites
2 # typage str float les chaines de caracteres slicing fonctions
3 # utilisation des listes
4
5 trame="$GPGGA,094527.289,47.6316,N,06.8500,E,1,04,3.2,200.2,M,,,0000*0E"
6
7 # obtention d'une liste
8 élément=trame.split(",")
9
10 #identification des éléments de la liste
11 heure=élément[1]
12
13 latitude=élément[2]
14 latitude_hémisphère= élément[3]
15 latitude_degré=latitude[0:2]
16 latitude_minutes=latitude[3:]
17
18 longitude=élément[4]
19 longitude_des=élément[5]
20
21 # affichage de l'heure
```

```

22 def affichage_heure(heure):
23     """ affichage de l'heure à partir d'une chaîne de caractères 'HHMMSS' """
24     h=heure[0:2]
25     mn=heure[2:4]
26     s=heure[4:6]
27     print("heure\t\t:",h,":",mn,":",s)
28     # return h,mn,s    possibilité de faire une fonction
29
30 affichage_heure(heure)
31
32 def conversion(data):
33     """ conversion en degré minutes secondes de données en degré et minutes """
34     data_degré=data[0:2]
35     data_minutes=data[3:5]
36     data_secondes=float("0."+data[5:])*60
37     return data_degré,data_minutes,data_secondes
38
39 latitude_degré,latitude_minutes,latitude_secondes=conversion(latitude)
40 print("latitude\t:\t",latitude_degré,"°",latitude_minutes,"'",latitude_secondes,"'",latitude_hémisphère)
41
42 longitude_degré,longitude_minutes,longitude_secondes=conversion(longitude)
43
44 print("longitude\t:\t",longitude_degré,"°",longitude_minutes,"'",longitude_secondes,"'",longitude_des)
45
46 # affichage carte
47 def affichage(zoom,latitude, longitude):
48     """ affiche la carte sur openstreetmap.org """
49     import webbrowser
50     webbrowser.open('https://www.openstreetmap.org/#map='+str(zoom)+'/'+(latitude)+'/'+(longitude))
51
52 #https://www.openstreetmap.org/?mlat=#map=16/47.6316/6.8503
53 affichage(16,latitude,longitude)

```

résultat :

```

heure      :   09 : 45 : 27
latitude   :   47 ° 63 ' 9.6 " N
longitude  :   06 ° 85 ' 0.0 " E

```

