

NSI – Le Projet AIS

Une classe – Un enseignant – Un chercheur

Voici les objectifs essentiels du projet :

Objectif 1 : présenter la technologie AIS

- généralités, qui est concerné...
- intérêt, finalité de la technologie
- matériel et fonctionnement, généralités

Rendu : sous forme d'un site WEB

Les données AIS étudiées portent sur le 22 janvier 2021.
Un fichier zip avec toutes les données est également à disposition.
Pour les objectifs suivants, vous pourrez utiliser en priorité :
- LE FICHER data-mini.csv, QUI CONTIENT LES DATA DES 5 PREMIÈRES SECONDES DE LA JOURNÉE
- LE FICHER data-init.csv QUI CONTIENT LES DATA DE LA PREMIÈRE HEURE

Un fichier allégé de la journée entière sera également mis à disposition...

Attention : quand le fichier est volumineux, le traitement peut prendre un certain temps...

Objectif 2 : comprendre l'organisation des données et les importer

- description générale du fichier CSV , taille, date...
 - nom des champs, et leur significations
 - le champ MMSI en particulier
 - les types de messages...
- importer un fichier AIS en Python pour qu'il soit traité, soit avec une liste de **liste**, soit avec une liste de **dictionnaires** avec la bibliothèque CSV. Voir le document d'accompagnement et le cours sur CSV.

A programmer :

1) écrire une fonction `importer_AIS(nom_du_fichier)` qui prend comme paramètre le str « nom_du_fichier » et qui renvoie le contenu exploitable du CSV :

```
def importer_AIS(nom_du_fichier):  
    ...  
    return data
```

2) On a remarqué que les champs 'Latitude' et 'Longitude' sont de la forme d'un str comme '49,23'.

Or pour que cette donnée soit exploitable sous forme d'un flottant, il faut qu'elle s'écrive : 49.23

Écrire une fonction `to_float(valeur)` qui prend comme paramètre une chaîne de caractère comme '49,23' et qui renvoie cette valeur sous la forme d'un flottant donc ici 49.23 .

Cette fonction devra utiliser à un moment la fonction `float()` , puis être testée.

```
def to_float(valeur):  
    ...  
    return v_float
```

Objectif 3 : le champ MMSI

→ expliquer ce qu'est le MMSI, et son intérêt pour les données.

→ écrire une fonction `nombre_MMSI(nom_du_fichier)` qui prend comme argument le nom d'un fichier AIS et qui renvoie le nombre de numéro MMSI **distincts**.

```
def nombre_MMSI(nom_fichier) :  
    ...  
    return nombre_MMSI_distincts
```

Remarque : penser à retenir les MMSI distincts dans une liste.

Pour les objectifs suivants, utiliser le fichier de documentation de la bibliothèque « folium ».

Objectif 4 : représenter sur une carte les bateaux d'un fichier

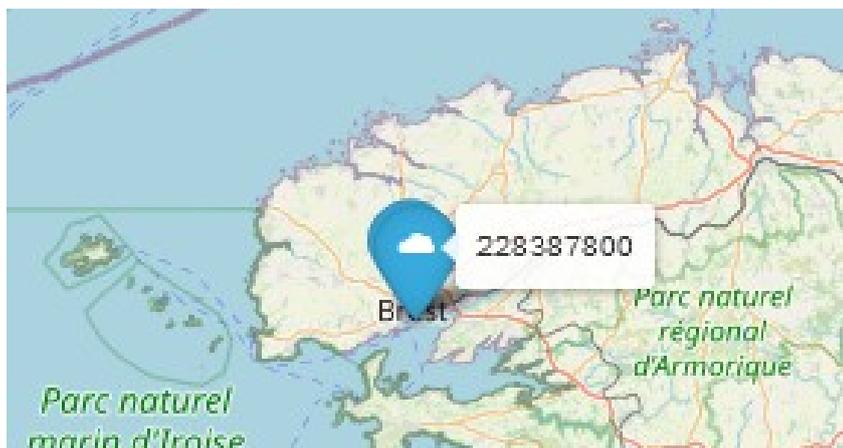
→ écrire une fonction `afficher_bateau_folium(nom_du_fichier)` qui prend comme argument un nom de fichier AIS et qui affiche dans une carte générée par folium avec les bateaux présents.

```
def afficher_bateau_folium(nom_du_fichier) :  
    m = folium.Map(...)  
    ...  
  
    m.save('carte1.html')
```

Remarques :

→ il est conseillé de ne faire apparaître sur la carte chaque numéro MMSI qu'une seule fois.

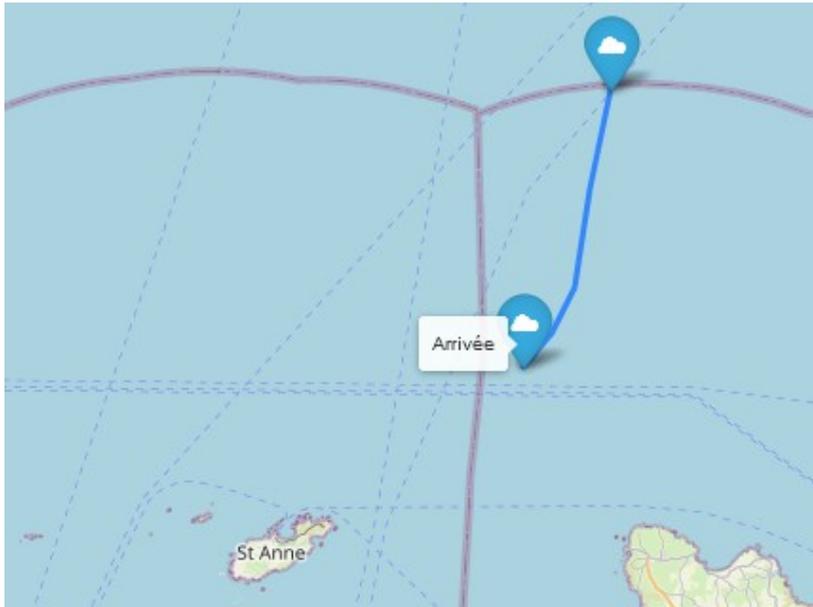
→ chaque bateau sera repéré par un Marker. Quand on passe la souris dessus, son numéro MMSI doit apparaître, et quand on clique dessus, on doit voir son nom.



Objectif 5 : représenter sur une carte les bateaux d'un fichier

→ écrire une fonction `affiche_trajet_folium(nom_du_fichier, num_MMSI)` qui prend comme argument le nom d'un fichier AIS et un numéro MMSI, et qui affiche sa trajectoire avec un Polyline.

```
def afficher_bateau_folium(nom_du_fichier) :  
    m = folium.Map(...)  
    ...  
  
    m.save('carte2.html')
```



Remarques :

- vous pourrez par exemple afficher la trajectoire du bateau dont le MMSI est « 235118115 ».
- en approchant le pointeur de la souris, afficher le numéro MMSI
- il est possible d'ajouter un Marker au début et à la fin pour les distinguer

Objectif 6 : représenter les bateaux qui sont dans ou proche du Havre

→ écrire une fonction `afficher_bateaux_LH(nom_du_fichier)` qui prend comme argument le nom d'un fichier AIS et qui affiche **uniquement** les bateaux qui sont dans le port du Havre, avec leur nom et leur MMSI.

