



## Photos géolocalisées et protection des données personnelles Cartes numériques, openstreetmap, uMap et Folium Calculs d'itinéraires

### 1) Localiser des photos à partir des données EXIF

D'où viennent ces photos ?

Un ami vous envoie des photos et vous demande de l'aider à identifier l'endroit où elles peuvent avoir été prises. Comme il s'agit visiblement d'un centre de recherche lié au numérique, il aimerait postuler pour faire un stage.

À réaliser :

Retrouver les renseignements suivants en explorant les métadonnées EXIF (Exchangeable Image File Format) d'une des huit photos : coordonnées géographiques du lieu de prise de vue, titre, description, copyright.

Les photos sont à l'adresse suivante :

<https://rafut.ddns.net/nextcloud/s/6dc7SzNAiY9qMeW>

Pour trouver les métadonnées, télécharger les photos dans un dossier de travail, puis décompresser le fichier. Clic droit sur le fichier > Propriétés > Détails permet d'obtenir les coordonnées géographiques du lieu de prise de vue. Les coordonnées géographiques sont exprimées en degrés sexagésimaux (DMS).

⇒ *Point sécurité : Prudence à propos des photos postées sur les réseaux sociaux. Est-il nécessaire de toujours envoyer des photos géolocalisées ?*

- *Pour désactiver la géolocalisation sur Android,*  
*1° dans le menu des réglages, désactiver « Position » ;*  
*2° dans l'appli de l'appareil photo, taper sur la roue crantée, et désactiver « Tags de localisation ».*

- a) Localiser le lieu de la prise de vue sur une carte numérique. On pourra s'aider d'un site web comme [www.openstreetmap.org](http://www.openstreetmap.org)

NB. Les métadonnées de localisation de l'image sont exprimées en degrés sexagésimaux (DMS). Pour les exploiter avec openstreetmap, il faudra les convertir en degrés décimaux (DD).

Utiliser un convertisseur comme :

<https://presqu-ile-de-crozon.com/conversion-des-coordonnees-gps.php>




- b) Avec le tableur Calc, réaliser un tableau avec les colonnes suivantes : Latitude, Longitude (en DD), et d'autres métadonnées EXIF.

Exemple.

NB. Dans le tableau, les décimales doivent être marquées par un POINT et non une VIRGULE ! Enregistrer ce tableau au format CSV dans un dossier de travail (accepter les paramètres par défaut).

## 2) Localiser les photos sur une carte avec le site uMap

<https://umap.openstreetmap.fr/fr/>

- Donner un nom à cette carte et l'enregistrer dans le site ;  
Bien noter l'URL de la carte pour pouvoir la retrouver et la réutiliser ;  
Copier-coller l'URL dans un traitement de texte et enregistrer dans le dossier de travail ;
- Si le menu Edition n'est pas ouvert (partie droite de l'écran, cliquer sur  en haut à droite de l'écran ;
- Cliquer sur  / importer des données / parcourir / sélectionner le fichier .csv qui contient les coordonnées des photos ;  
Cliquer sur Importer les données dans la carte à partir du fichier CSV. Enregistrer. Les points s'affichent ;
- Éditer les paramètres de la carte pour la personnaliser 
  - Propriétés de forme par défaut : choisir les couleurs et types des marqueurs (faire des essais)
  - Options d'interaction par défaut : choisir "Tableau" dans le menu *Gabarit du contenu de la popup* et choisir "toujours" dans le menu *Afficher une étiquette* (faire des essais).
  - Enregistrer à chaque étape !
  - Clic droit sur un point => Quitter le mode d'édition

Tuto détaillé de uMap

[https://wiki.cartocite.fr/doku.php?id=umap:tutoriel\\_umap](https://wiki.cartocite.fr/doku.php?id=umap:tutoriel_umap)

### 3) Utiliser Python pour créer des cartes avec Folium

Ouvrir un interpréteur Python. Y recopier ou copier/coller le source du programme. L'enregistrer dans le dossier de travail. Adapter et modifier les variables écrites en *italiques*. Enregistrer. Exécuter.

#### a) Placer un marqueur sur une carte avec la bibliothèque Python *folium*

À réaliser :

Afficher sur la carte un marqueur de la latitude et la longitude du lycée.

- Créer un fichier python dont le contenu est le suivant, en remplaçant *latitude* et *longitude* par les coordonnées de votre lycée, puis celles de chez vous.

---

```
import folium

# définit les coordonnées du lieu à pointer
carte = folium.Map(location=[latitude, longitude], zoom_start=15)

# affiche un marqueur avec les coordonnées
folium.Marker([latitude, longitude], popup="Mon lycée").add_to(carte)

# pour voir la carte dans la fenêtre graphique (Basthon : 3e bouton sous la
# console)
carte.display()

# sauvegarde de la carte (basthon : 3e bouton sous l'éditeur ; choisir dossier
# et nom de fichier.
carte.save('maCarte.html')

# pour enregistrer
import basthon
# ouvre une boîte de dialogue pour enregistrer le fichier
basthon.download("maCarte.html")
```

---

- Lors de l'exécution du programme Python, un fichier nommé maCarte.html a été créé dans le répertoire de travail utilisé par votre environnement de développement Python. Le trouver et l'ouvrir dans un navigateur. Que contient-il ?
- Modifier le texte du "popup" en citant le nom du lycée.
- A quoi correspond le paramètre zoom\_start ? Tester différentes valeurs entre 0 et 18.
- Quelles valeurs semblent pertinentes pour présenter un lycée ?
- Quelles valeurs semblent pertinentes pour présenter votre domicile ?

b- Tracer un itinéraire sur une carte avec *folium*

**À réaliser :**

- Créer un fichier python dont le contenu est le suivant :

---

```
import folium

# définir les coordonnées du point de départ
carte2 = folium.Map(location=[45.5, 2.0], zoom_start=6)

# Définir les points de l'itinéraire
route = [
    (48.116178, -1.639469),
    (47.471627, -0.548873),
    (47.393767, 0.682236),
    (45.763663, 4.823724),
    (45.218181, 5.807142),
]

# donner le nom des points de départ et d'arrivée en survolant la ligne
folium.PolyLine(route, tooltip = "itinéraire Rennes - Grenoble").add_to(carte2)

# pour voir la carte dans la fenêtre graphique (Basthon : 3e bouton sous la
# console)
carte2.display()

# sauvegarde de la carte (basthon : 3e bouton sous l'éditeur ; choisir dossier
# et nom de fichier.
carte2.save('ma2eCarte.html')

# pour enregistrer
import basthon
# ouvre une boîte de dialogue pour enregistrer le fichier
basthon.download("ma2eCarte.html")
```

---

- La carte que produit ce programme contient une "ligne brisée" représentant un itinéraire. Identifier les villes de départ et d'arrivée sur la carte et modifier le message du "tooltip" pour qu'il affiche au survol les villes de départ et d'arrivée.
- Ajouter au départ et à l'arrivée des marqueurs affichant les noms des villes lorsqu'on les clique. (cf. popup §2 a)

b) un outil qui affiche les coordonnées de chaque point que l'on clique.  
**À réaliser :**

---

```
import folium

latitude = 46.6
longitude = 2.6
zoom = 6

carte3 = folium.Map(location=[latitude, longitude], zoom_start=zoom)

#outil d'affichage des coordonnées des lieux cliqués
carte3.add_child(folium.LatLngPopup())

# pour voir la carte dans la fenêtre graphique (Basthon : 3e bouton sous la
# console)
carte3.display()

# sauvegarde de la carte (basthon : 3e bouton sous l'éditeur ; choisir
dossier
# et nom de fichier.
ccrte3.save('ma3eCarte.html')

# pour enregistrer
import basthon
# ouvre une boîte de dialogue pour enregistrer le fichier
basthon.download("ma3eCarte.html")
```

---

- c) Pour aller plus loin : tracer un polygone sur la carte
- Le mot "**hexagone**" est fréquemment utilisé pour désigner la partie continentale de la France métropolitaine. Utiliser l'instruction `folium.Polygon` ci-dessous (à insérer dans le programme précédent) pour déterminer un hexagone qui illustre cette affirmation.

---

```
# liste des coordonnées à compléter
zone = folium.Polygon( [
    [latitude1,longitude1],
    ...
    [latitude6,longitude6],
    ],

# colorier la surface
color='green', fill = True
).add_to(carte3)
```

---

- Quelle modification permettrait que la zone soit rouge et non verte ?
- Remplacer `fill = True` par `fill = False` : à quoi sert le paramètre `fill` ?
- Ajouter un outil d'affichage des coordonnées des lieux cliqués : outil `add_child(folium.LatLngPopup())`
- Ajouter le méridien de Greenwich en rouge : outil `folium.PolyLine`

#### 4) Calcul de l'itinéraire le plus court – algorithme de Dijkstra

Utiliser l'appli en ligne

<https://www.graphonline.ru/fr>

- récupérer le fichier graphe\_itinéraire.graphml
- cliquer à droite sur les sommets pour leur donner des noms de villes
- cliquer à droite sur les arêtes pour leur donner un poids :
  - distance kilométrique, ou bien
  - temps de parcours, ou bien
  - montant des péages
- menu Algorithmes / Plus court chemin avec l'algorithme de dijkstra
  - cliquer sur le sommet de départ et sur le sommet d'arrivée pour visualiser le trajet le plus court.
  - Calculer des itinéraires alternatifs en supprimant un lien (route barrée)

