éduSCOL
Informer et accompagner
les professionnels de l'éducation

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

MINISTÈRE
DE L'ÉDUCATION
NATIONALE ET
DE LA JEUNESSE

Sciences numériques et technologie

ENSEIGNEMENT
COMMUN

LOCALISATION, CARTOGRAPHIE ETMOBILITÉ - ROUTAGE ET CALCULS D'ITINÉRAIRES

VOIE GÉNÉRALE
ET TECHNOLOGIQUE

2DE | 1RE | TLE

Contenus et capacités

Localisation, cartographie et mobilité

Contenus	Capacités attendues
Calculs d'itinéraires	Calculs d'itinéraires. Utiliser un logiciel pour calculer un itinéraire. Représenter un calcul d'itinéraire comme un problème sur un graphe.

Note d'intention

Cette activité permet de comprendre le fonctionnement des applications de location des vélos en libre service à la minute (ou tout autre moyen de transport) et que l'on retrouve souvent aux abords du lycée. Plusieurs sociétés se partage le marché à Paris. Nous allons nous intéresser au fonctionnement de ces vélos afin de savoir :

- Comment elles apparaissent sur une carte lorsqu'elles sont disponibles.
- Comment le problème de calcul d'itinéraires vient du calcul du ^plus court chemin sur un graphe.
- Comment est calculé le trajet entre vous et la trottinette ou visualiser le trajet que vous avez effectué.

Retrouvez éduscol sur :

Exercices

Exercice 1

Lancez la suite anaconda et utilisez le logiciel spyder pour saisir le programme ci-dessous.

```
38 import folium
39 c= folium.Map(location=[48.86998, 2.285564])
40 c.save('macarte1.html')
41
```

Une fois le code ci-dessus exécuté, rendez-vous dans le répertoire que vous avez créé. Vous devriez trouver un fichier "maCarte1.html". Double-cliquez sur ce fichier, cela devrait normalement ouvrir un navigateur web : la carte centrée sur la ville de Paris est à votre disposition. Notez bien que nous avons une véritable carte et pas une simple image (il est possible de zoomer ou de se déplacer).

Explication du programme

- La première ligne : "import folium" permet d'importer la bibliothèque folium afin de pouvoir l'utiliser
- La deuxième ligne est le coeur de notre programme, nous définissons une variable "c" qui va contenir notre objet carte. "folium.Map(location=[48.86998, 2.285564])" génère cet objet carte, la carte sera centrée sur le point de latitude "48.86998" et de longitude "2.285564". Plus généralement nous avons "folium.Map(location=[latitude, longitude])". Il suffit donc de renseigner la bonne longitude et la bonne latitude pour que la carte soit centrée sur le point désiré.
- La dernière ligne permet de générer la page HTML qui va permettre d'afficher la carte.

Exercice 2

Modifiez le programme ci-dessus pour qu'il génère une carte centrée sur la ville de votre choix (la longitude et la latitude d'une ville sont facilement trouvables sur le web).
Il est possible d'obtenir un niveau de zoom différent en ajoutant un paramètre "zoom_start"

```
34 import folium
35 c= folium.Map(location=[48.86998, 2.285564], zoom_start=20)
36 c.save('macarte2.html')
37
```

Plus la valeur de "zoom_start" sera grande et plus le zoom sera important.

Exercice 3

On va maintenant voir comment on peut ajouter des POI (Points Of Interest) sur la carte. Un point d'intérêt sera simplement défini par ses coordonnées (latitude et longitude).

```
28 import folium
29 c= folium.Map(location=[48.86998, 2.285564], zoom_start=20)
30 folium.Marker([48.865519, 2.279513]).add_to(c)
31 c.save('lyceejds.html')
32
```

Nous avons uniquement ajouté la ligne "folium.Marker...", il faut juste renseigner les coordonnées souhaitées (ici 48.865519 pour la latitude et 2.279513 pour la longitude).

Il est possible d'ajouter plusieurs marqueurs sur une même carte, il suffira d'ajouter autant de ligne "folium.Marker([latitude, longitude]).add_to(c)" que de marqueurs désirés.
C'est ainsi que les vélos apparaissent sur la carte de l'application.

Exercice 4

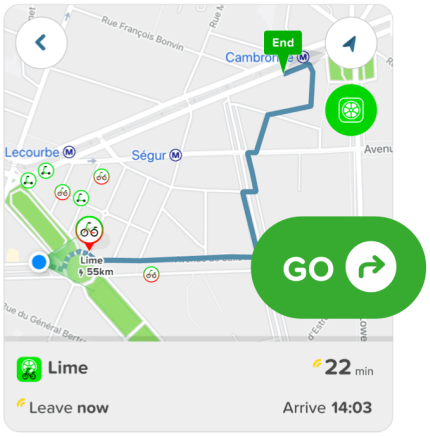
Lorsque vous cliquez sur le vélo de votre choix, un « popup » apparait comme sur la photo ci-dessous. On a donc associé les informations concernant l'autonomie restante, le tarif, une partie du numéro de série…

Saisissez et testez le programme ci-dessous pour centrer

```
import folium
c= folium.Map(location=[48.8652778, 2.28],zoom_start=20)
folium.Marker([48.8652778, 2.28],popup="Lycée J de Sailly").add_to(c)
c.save('maCarte1.html')
```

Il suffit de cliquer sur le marqueur pour que l'information définie par le paramètre "popup" apparaisse à l'écran (ici en cliquant sur le marqueur nous verrons donc apparaitre "Lycée Janson de Sailly").

Lorsque l'on veut récupérer une trottinette ou que l'on a fini son trajet, l'application nous propose d'afficher l'itinéraire.



Nous allons donc nous intéresser au calcul d'itinéraire mais avant cela comprenons ensemble comment notre position est détectée et transmise.

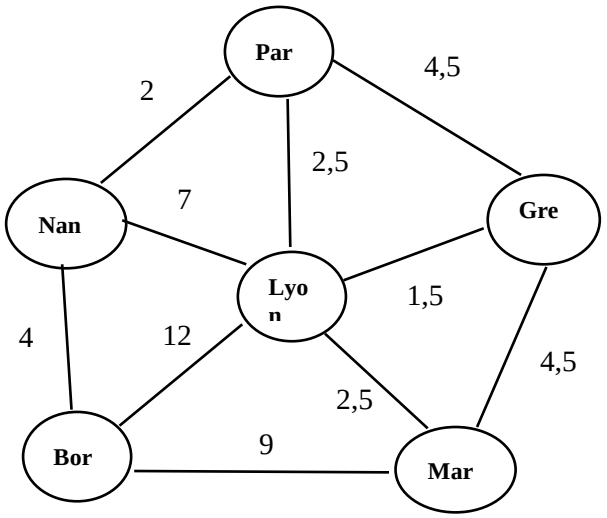
Pour pouvoir trouver le plus court chemin les puces de géo localisation utilisent un algorithme (il diffère en fonction des marques) qui se rapproche de l'algorithme de Dijkstra.

https://www.youtube.com/watch?v=MybdP4kice4&feature=youtu.be

Exercice 2 :

Après avoir regardé la vidéo ci-dessus, vous allez trouver le plus court chemin entre Grenoble et Bordeaux.

Bordeaux → Nantes : 4h
Bordeaux → Marseille : 9h
Bordeaux → Lyon : 12h
Nantes → Paris : 2h
Nantes → Lyon : 7h
Paris → Grenoble : 4h30
Marseille → Lyon :2h30
Marseille → Grenoble : 4h30
Lyon → Grenoble : 1h30
Lyon → Paris : 2h30



Bordeaux	Nantes	Lyon	Marseille	Paris	Grenoble	

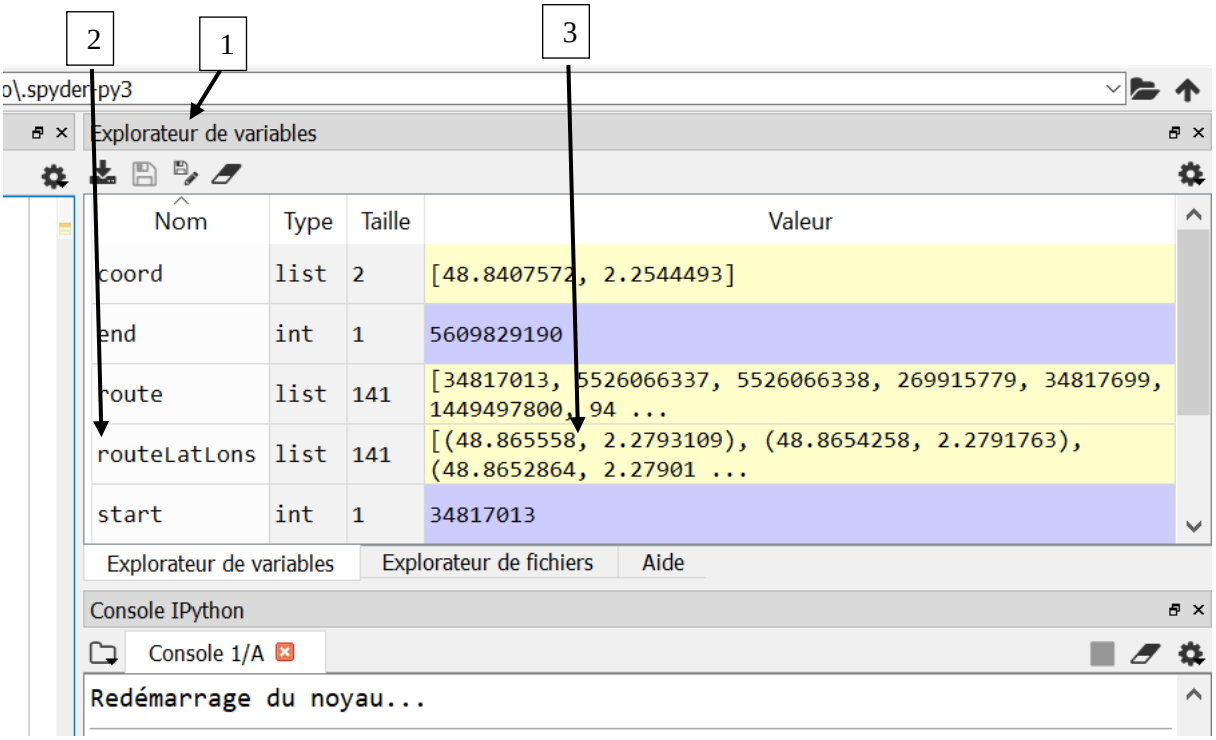
Nous avons utilisé la bibliothèque Folium qui nous a permis d'afficher des cartes centrées sur certains points. Nous allons utiliser la bibliothèque pyroutelib3 qui nous permet de calculer des itinéraires à l'aide des cartes OSM.

Exercice 3

Tapez le programme ci-dessous avec spyder

```
from pyroutelib3 import Router # Importe les outils pour calculer l'itinéraire
import folium # Importe les cartes d'OSM
router = Router("car") # Initialisation du calcul d'itinéraire pour une voiture

depart = router.findNode(48.8656, 2.2794400000000223) # Point de départ
arrivee = router.findNode(48.8407, 2.2545000000000073) # Point d'arrivée

status, route = router.doRoute(start, end) # Trouve la route - une liste d'intersections sur OSM

if status == 'success':
    routeLatLons = list(map(router.nodeLatLon, route)) # Liste des différents points de passage

```

Une fois l'exécution du programme terminée (cela peut prendre quelques minutes en fonction du nombre de points à calculer), à l'aide de l'explorateur de variables(1) de spyder, regardez le contenu de la variable "routeLatLons"(2). Comme vous pouvez le constater, cette variable contient une liste de couples de valeurs (latitude, longitude)(3). Cette liste contient donc les coordonnées des différents points par lesquels il faut passer pour se rendre du point de départ jusqu'au point d'arrivée (en passant bien évidemment par les routes définies dans Open Street Map).



Exercice 4

Modifiez le programme du "À faire vous-même 2", pour calculer le trajet entre deux villes de votre choix avec le moyen de transport de votre choix (car, foot, horse…). Vous pouvez utiliser le site ci-dessous pour récupérer les coordonnées GPS de différents endroits.

https://www.coordonnees-gps.fr/

Exercice 5

Nous allons maintenant afficher la liste des différents points sur une carte pour en faire un itinéraire.
Testez le programme ci-dessous (une fois le programme Python exécuté, ouvrez avec un navigateur web le fichier "Itin.html").

```
from pyroutelib3 import Router # Importe les outils pour calculer l'itinéraire
import folium # Importe les cartes d'OSM
router = Router("car") # Initialisation du calcul d'itinéraire pour une voiture

depart = router.findNode(48.8656, 2.2794400000000223) # Point de départ
arrivee = router.findNode(48.8407, 2.2545000000000073) # Point d'arrivée

status, route = router.doRoute(start, end) # Trouve la route - une liste d'intersections sur OSM

if status == 'success':
    routeLatLons = list(map(router.nodeLatLon, route)) # Liste des différents points de passage

    c= folium.Map(location=[48.852968, 2.349901],zoom_start=14) #On centre la carte sur un endroit avec un zoom
for coord in routeLatLons: # On parcourt la liste des coordonnées de l'itinéraire
    coord=list(coord)
    folium.Marker(coord).add_to(c) # On ajoute un marker pour chaque coordonnée
c.save('Itin.html') # On enregistre l'itinéraire sur une carte

```

Le calcul d'itinéraire est entre quels endroits :
Départ : ……
Arrivée : ……..

Exercice 6

Modifiez le programme ci-dessus pour faire apparaitre sur une carte un itinéraire de votre choix (en définissant le point de départ, le point d'arrivée dans notre cas le type de véhicule).

Retrouvez éduscol sur :

Retrouvez éduscol sur :