



## Objectifs

1. Utiliser un logiciel pour calculer un itinéraire.
2. Représenter un calcul d'itinéraire comme un problème sur un graphe.

# 1 Créer une carte personnalisée avec Python

(d'après [https://pixees.fr/informatiquelycee/n\\_site/snt\\_carto\\_osmPerso.html](https://pixees.fr/informatiquelycee/n_site/snt_carto_osmPerso.html))

## A faire vous même 1

Sur votre PC, avec l' Explorateur de fichier Windows, créez un dossier (repérez bien l'endroit) et nommez-le par exemple "carte\_OSM".

## A faire vous même 2

1. Lancez préalablement Geoportail (voir activité précédente)
2. Validez la géolocalisation de votre lieu de connexion afin d'obtenir les coordonnées à saisir (par défaut, ce sera -15.972552 , -5.724564 et vous serez perdus quelque part au milieu de l'océan Atlantique comme un certain...Napoléon.
3. Indiquez vos coordonnées ici :

## A faire vous même 3

On vous donne le programme Python avec des commentaires (en vert).

```
import folium # C' est une bibliothèque permettant de créer vos propres cartes
c= folium.Map(location=[-15.972552 , -5.724564 ]) # une variable "c" contient l'objet carte.
# elle sera centrée sur le point de latitude/longitude choisi
c.save('maCarte1.html') # génère la page HTML qui va permettre d'afficher la carte.
```

Dans ce script, l'instruction :

```
folium.Map(location=[latitude, longitude])
```

génère une carte centrée sur les coordonnées longitude et latitude qu'il suffit de renseigner.

1. Utilisez le logiciel EduPython pour saisir le programme ci-dessus (copier/coller).  
Attention ! pensez à modifier les coordonnées dans le programme avec celles que vous avez relevées au *À faire vous-même 2*.
2. Le fichier devra être enregistré dans le répertoire créé au *À faire vous-même 1* (le fichier pourra être nommé "carte\_1.py").
3. Exécutez-le.
4. Une fois le code ci-dessus exécuté, rendez-vous dans le répertoire que vous

avez créé au *À faire vous-même 1*. Vous devriez trouver un fichier « `maCarte1.html` ».

5. Double-cliquez sur ce fichier, cela devrait normalement ouvrir votre navigateur web : la carte centrée sur votre lieu de villégiature 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).
6. Faites une copie d'écran de la carte.

#### A faire vous même 4

Il est possible d'obtenir un niveau de zoom différent en ajoutant un paramètre "zoom\_start = xx", plus la valeur de « `zoom_start` » sera grande et plus le zoom sera important.

1. Modifiez comme indiqué en grisé ci-dessous et testez votre programme  
`c= folium.Map(location=[latitude, longitude ],zoom_start=15)`
2. Faites une copie d'écran de la carte.

#### A faire vous même 5

Afin de vraiment personnaliser la carte, il est possible d'ajouter des marqueurs sur la carte. Un marqueur sera simplement défini par ses coordonnées (latitude et longitude). L'instruction est de la forme :

```
folium.Marker([vos_coordonnées]).add_to(votre_carte).
```

Rq : Il est possible d'ajouter plusieurs marqueurs sur une même carte, il suffira d'ajouter autant de ligne « `folium.Marker` » que de marqueurs désirés.

1. Rajoutez après la deuxième ligne de votre programme précédent la ligne ci-dessous avec vos coordonnées du *À faire vous-même 1* et testez votre programme.

```
folium.Marker([latitude, longitude]).add_to(c)
```

#### A faire vous même 6

Il est possible d'associer une information à un marqueur (qui s'affiche en cliquant dessus) en ajoutant le paramètre "popup = votre\_commentaire" .

1. Modifiez comme indiqué en grisé ci-dessous et testez votre programme  
`folium.Marker([latitude, longitude],popup="Ile Sainte Hélène").add_to(c)`
2. Faites une copie d'écran de la carte avec le marqueur et l'information "popup".

## 2 Calculs d'itinéraires

(d'après [https://pixees.fr/informatiquelycee/n\\_site/snt\\_carto\\_route.html](https://pixees.fr/informatiquelycee/n_site/snt_carto_route.html))

Comme vous avez pu le constater quand vous avez travaillé sur Open Street Map, il est possible de définir les voies de communication (principalement les routes). La

base de données OSM contient donc les routes (enfin, la plupart des routes...). En utilisant ces données, il est possible de développer des outils capables de calculer des itinéraires routiers (comme le propose tous les logiciels "GPS" : Waze, ViaMichelin, Mappy...) : vous renseignez votre lieu de départ, votre lieu d'arrivée puis le logiciel calcule votre itinéraire. Ce calcul d'itinéraire repose sur des algorithmes relativement complexes, par exemple l'algorithme de Dijkstra qui permet d'obtenir le plus court chemin entre deux points.

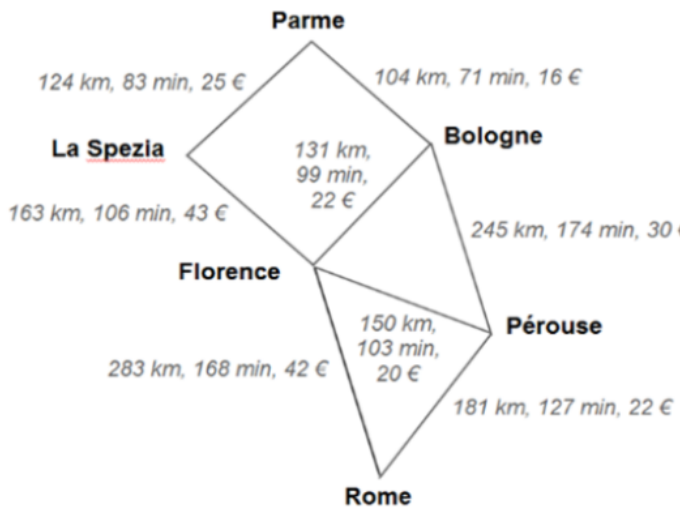
Sans entrer dans les détails, l'algorithme de Dijkstra travaille sur des graphes (chaque ville est un sommet du graphe et chaque route est une arête du graphe). Visionnez cette vidéo simplificatrice <https://youtu.be/JPeCmKFrKio> pour comprendre.

Rq : si vous souhaitez approfondir un peu, tout en restant compréhensible voici deux liens <https://www.youtube.com/watch?v=MybdP4kice4> et

<https://www.maths-cours.fr/methode/algorithme-de-dijkstra-etape-par-etape/>

En général, on recherche le chemin le moins long, ou le plus rapide, ou le moins cher ...

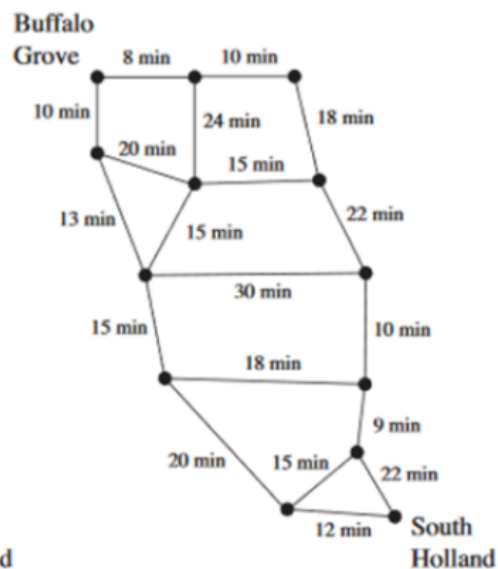
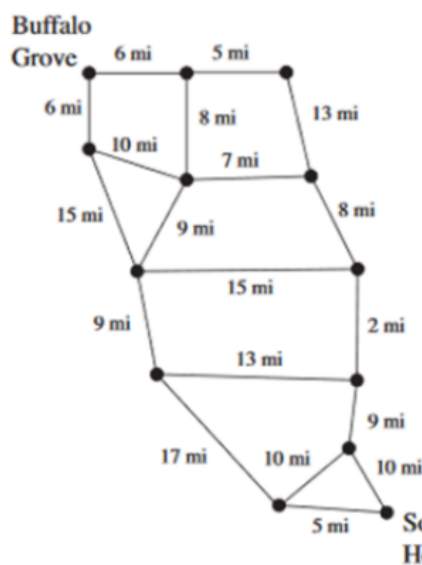
#### A faire vous même 7

 <p><b>Source (distances, durées, et coûts) : ViaMichelin.</b></p>	<p>Tous les chemins mènent à Rome, mais pour un parmesan (un habitant de Parme !):</p> <p>Quel est le moins long ? .....</p> <p>Le plus rapide ? .....</p> <p>Le moins cher ? .....</p> <p>Est-ce le même chemin dans les trois cas ? .....</p>
--	---

#### A faire vous même 8

Les deux cartes suivantes représentent

- à gauche : le trajet en miles (mi) du train Expressways in the Chicago area
- à droite la durée (min) des différentes étapes.



1. Déterminer le plus court trajet de Buffalo Grove à South Holland.

2. Déterminer le trajet le plus rapide pour relier les deux mêmes villes.

#### A faire vous même 9

Avec Geoportail relevez les coordonnées de la piscine de Dinard (par défaut, ce sera -15.972552 , -5.724564 et vous serez toujours perdus...)\*\*.

1. Indiquez les coordonnées de la piscine de Dinard ici :

#### A faire vous même 10

On vous donne le script Python ci-dessous permettant de générer un itinéraire.

```
1. from pyrouelib3 import Router
2. router = Router("car")
3. depart = router.findNode(-15.947465 , -5.684566)
4. arrivee = router.findNode(-15.96331 , -5.648689)
5. status, route = router.doRoute(depart, arrivee)
6. if status == 'success':
7.     routeLatLons = list(map(router.nodeLatLon, route))
```

Ce programme, fait appel en 1ère ligne à la bibliothèque Python pyrouelib3 qui contient des "outils" qui mettent en œuvre l'algorithme de Dijkstra pour calculer des itinéraires à partir des données d'OpenStreetMap.

1. Dans le dossier, créé au À faire vous-même 1, avec l'Explorateur de fichier Windows, copier/coller le fichier "pyrouelib3.py"

([http://ninoo.fr/cours/0\\_Les\\_Cordeliers\\_2019\\_2020/2nd\\_SNT/chap5\\_la\\_geolocalisation/pyrouelib3.py](http://ninoo.fr/cours/0_Les_Cordeliers_2019_2020/2nd_SNT/chap5_la_geolocalisation/pyrouelib3.py) )

2. Utilisez le logiciel EduPython pour saisir le programme\* ci-dessus

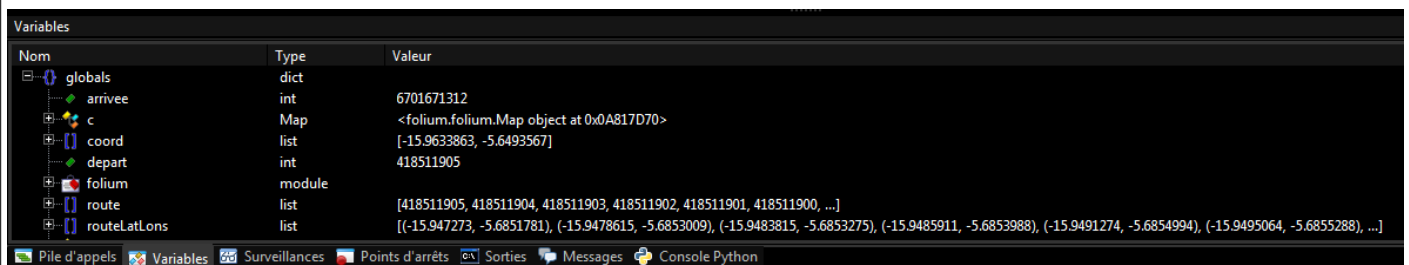
(copier/coller), en respectant l'indentation de la dernière ligne.

Attention ! pensez à modifier dans le programme les coordonnées (latitude, longitude) avec celles que vous avez relevées au *À faire vous-même 2* et *À faire vous-même 9*.

Ce nouveau fichier devra être enregistré dans le répertoire créé au *À faire vous-même 1* (le fichier pourra être nommé "carte\_2.py").

### 3. Exécutez-le.

Une fois l'exécution du programme terminée (cela peut prendre quelques minutes), à l'aide de l' "explorateur de variables" d'EduPython, visionnez le contenu de la variable "routeLatLons".



The screenshot shows the 'Variables' window in EduPython. It contains a table with three columns: 'Nom', 'Type', and 'Valeur'. The variable 'routeLatLons' is highlighted, showing it is a 'list' type with a long list of coordinate pairs. Other variables like 'globals', 'arrivee', 'c', 'coord', 'depart', 'folium', and 'route' are also visible.

Nom	Type	Valeur
globals	dict	
arrivee	int	6701671312
c	Map	<folium.folium.Map object at 0x0A817D70>
coord	list	[[-15.9633863, -5.6493567]]
depart	int	418511905
folium	module	
route	list	[418511905, 418511904, 418511903, 418511902, 418511901, 418511900, ...]
routeLatLons	list	[(-15.947273, -5.6851781), (-15.9478615, -5.6853009), (-15.9483815, -5.6853275), (-15.9485911, -5.6853988), (-15.9491274, -5.6854994), (-15.9495064, -5.6855288), ...]

Comme vous pouvez le constater, cette variable contient une liste de couples de valeurs (latitude, longitude). 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 OpenStreetMap).

## A faire vous même 11

Comme l'avez lu le script Python a été fourni sans commentaires. Les voici, dans le désordre bien sûr.

- |    |   |
|----|---|
| A. | # Définit le moyen de transport pour le trajet, parmi: car, cycle, foot, horse, tram, train |
| B. | # import de la bibliothèque "pyroutelib3" pour générer des itinéraires                      |
| C. | # définit les coordonnées du point de départ, ('node' signifie 'noeux')                     |
| D. | # définit les coordonnées du point d'arrivée  |
| E. | # la ligne suivante est exécutée si le calcul est mené à son terme ("if" ligne précédente)  |
| F. | # "routeLatLons" contient liste des coord. des points du chemin (départ => arrivée)         |
| G. | # permet d'effectuer le calcul de l'itinéraire entre les points de départ et d'arrivée      |

1. Collez le bon commentaire ( avec le #) en face de chaque ligne de votre programme nommé "carte\_2.py".
2. Faites une copie d'écran du programme avec les commentaires

## A faire vous même 12

Avoir une liste de coordonnées, c'est déjà pas mal, mais cette liste n'est pas très exploitable telle quelle. Nous allons donc utiliser ce que nous avons déjà vu précédemment sur la création de cartes.

Vous allez maintenant compléter votre programme "carte\_2.py" comme suit :

1. Rajoutez en début de programme la ligne :

```
import folium #import de la bibliothèque de génération de cartes sur fond OpenStreetMap
```

2. Rajoutez en fin de programme les lignes suivantes en respectant bien les indentations des deux lignes qui suivent le "for" et en modifiant les coordonnées avec celle du *À faire vous-même 2*:

```
c= folium.Map(location=[-15.947465 , -5.684566],zoom_start=12)
for coord in routeLatLons: # pour chaque coordonnée de la liste routeLatLons
    coord=list(coord) # pour chaque couple de coordonnées ...
    folium.Marker(coord).add_to(c) # ... ajouter un marqueur.
c.save('maCarte2.html')
```

3. Exécutez ce programme. Puis ouvrez avec votre navigateur web le fichier "maCarte2.html" qui vient d'être créée dans le dossier du *À faire vous-même 1*.
4. Faites une copie d'écran de la carte obtenue avec les marqueurs de cheminement

### A faire vous même 13

1. Modifiez le programme "carte\_2.py" pour qu' apparaisse sur l' itinéraire de la piscine des Pommiers à Léhon pour rejoindre le lycée Les Cordeliers à vélo.
2. Faites une copie d'écran de la carte obtenue avec les marqueurs de cheminement.