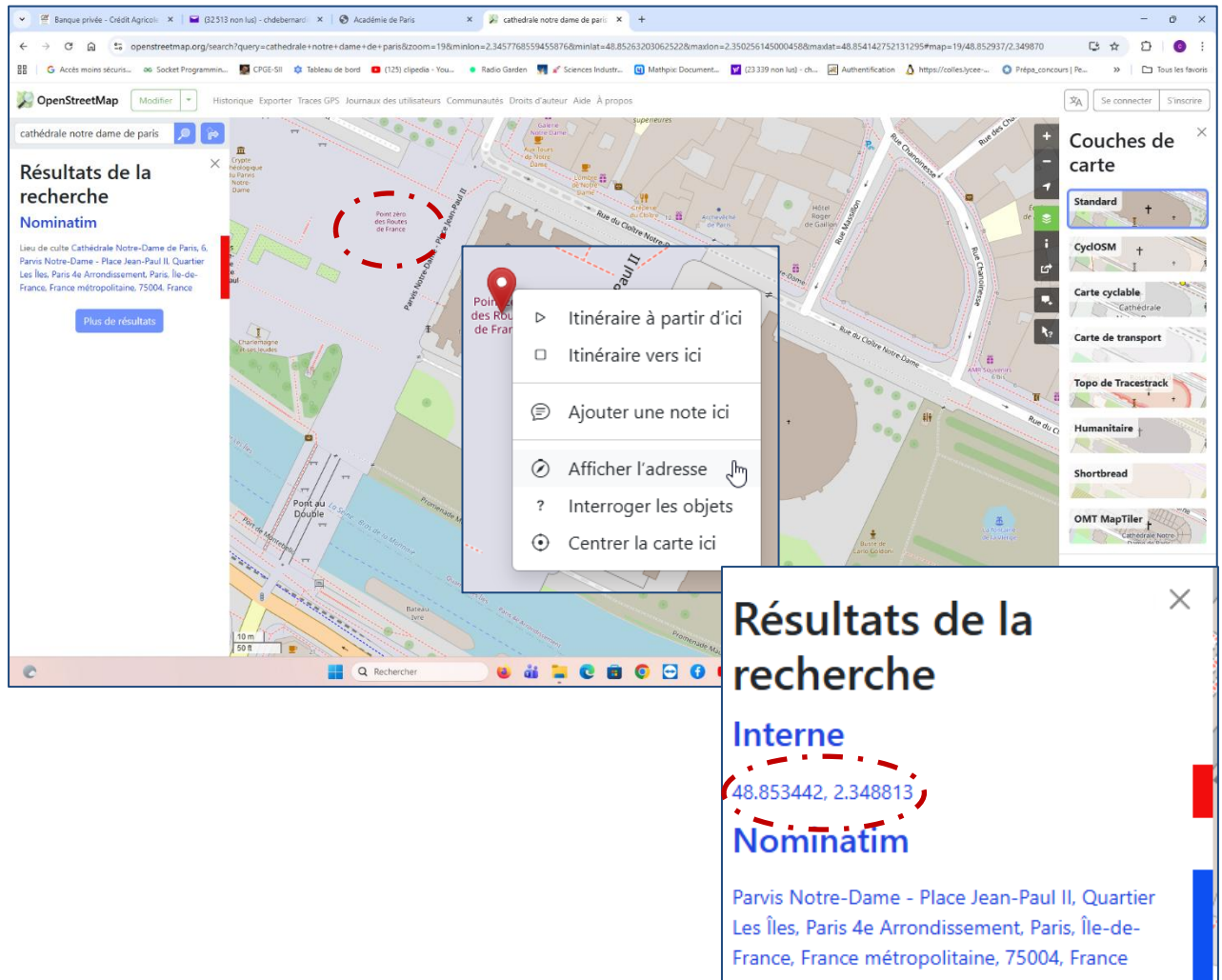


Nom :

Document professeur :

Préliminaire : Récupération des points de géolocalisation sur le site OSM

- 1) Sur le site OSM, rechercher la position GPS exacte du point 0 des routes de France (sur le Parvis de la Cathédrale Notre Dame de Paris.)
Aller sur le site proposé et taper « cathédrale Notre Dame de Paris », rechercher sur le parvis le point 0 des routes de France.



On trouve : Parvis de Notre Dame : 48.85342, 2.348785

- 2) Sur le site OSM, rechercher la position GPS exacte de l'entrée de la Grande Mosquée de Paris dans le V arrondissement de Paris.

On trouve : Grande Mosquée de Paris : 48.84209, 2.35514

- 3) Sur le site OSM, rechercher la position GPS exacte du parvis du Sacré Cœur dans le XVIII arrondissement de Paris.

On trouve : Parvis du Sacré Cœur : 48.88622, 2.34276

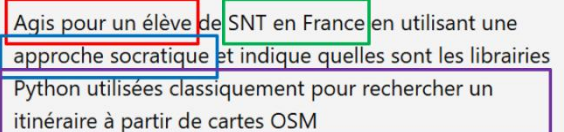
Nom :

Découverte des librairies existantes par l'art du prompt

Un premier prompt imposé

4) Dans le prompt précédent, encadrer en couleur les différentes parties du prompt.

- ❖ En rouge, le rôle,
- ❖ En violet la tâche,
- ❖ En vert le contexte,
- ❖ En bleu la ou les contraintes.



Agis pour un élève de SNT en France en utilisant une approche socratique et indique quelles sont les librairies Python utilisées classiquement pour rechercher un itinéraire à partir de cartes OSM

Exploitation d'un des résultats possibles

5) L'approche Socratique est-elle connue de l'IA ? Est-ce une approche par interactions de type Questions - Réponses ou une approche linéaire donnant simplement la réponse attendue ?

L'approche est connue par l'IA et elle sait qu'il faut mettre en place un questionnement suivi (auquel par ailleurs elle répond elle-même). Il n'est pas du tout impossible de lui demander d'analyser les réponses élèves mais dans ce cas l'approche Socratique est utilisée en phase de **révision** et non en phase **d'apprentissage** (ou de découverte).

6) A l'issue de cette première partie de dialogue, y-a-t-il des questions plus précises qui apparaissent et finalement l'IA propose-t-elle une démarche dans la résolution du problème ?

L'IA met en place un ensemble de questions plus précises et elle indique ainsi une trame dans les phases nécessaires pour résoudre le problème proposé. **C'est le grand apport de la démarche Socratique et il faut insister avec les élèves sur cette manière d'utiliser l'IA.**

7) De manière plus concrète, quelles sont les étapes à réaliser pour afficher l'itinéraire demandé ?

Charger les données, tracer des graphes, et rechercher le plus court chemin par l'algorithme de Dijkstra (ou A*). Il n'est pas ici question de maîtriser ces deux algorithmes ou même un des deux ; en revanche, cette approche correspond nettement au niveau 1 de la taxonomie de Bloom (j'en ai entendu parler). Elle permettra dans le futur de s'appuyer sur cet exemple pour commencer à étudier des algorithmes de plus court chemin.

8) Quelles sont donc les deux librairies à retenir ?

On va prendre le combo classique, osmnx + networkx

Un second prompt imposé également et codage de la solution proposée

Un second prompt imposé également

9) Par l'analyse du code proposé, répondre alors aux questions qui sont posées dans le cadre conversationnel avec l'IA et qui sont reproduites ci-dessous :

- ❖ Comment récupérer la carte ?

```
G = ox.graph_from_point(notre_dame, dist=2000, network_type="walk")
```

- ❖ Comment transformer ces routes en graphe ?

idem

- ❖ Comment trouver un chemin le plus court ?

Nom :

```
route = nx.shortest_path(G, start_node, end_node, weight='length')
```

❖ Comment afficher le résultat ?

```
ox.plot_graph_route(G, route, route_linewidth=3, node_size=0)
```

10) Quelle est l'information présente dans le code indiquant que la recherche de l'itinéraire est faite pour un trajet pédestre ?

On constate que la carte téléchargée correspond à un réseau de type walk.

```
G = ox.graph_from_point(notre_dame, dist=2000, network_type="walk")
```

Codage de la solution proposée

11) Indiquer quelle est la raison principale de cette durée d'exécution ?

Ce script est long à s'exécuter car le parcours du graphe par l'algorithme de plus court chemin est tout simplement très long.

12) Dans l'encadré ci-dessous (ou sur votre compte rendu informatique), recopier la carte proposée et vérifier qu'elle correspond bien à votre demande.

On obtient une des solutions proposées ci-dessous (ou autre).



Nom :

Des prompts libres mais des contraintes imposées

13) Proposer deux prompts distincts permettant d'obtenir un exemple de script et faire afficher les cartes.

✚ Pour la première demande : Exemple de prompt :

Je souhaite désormais que tu modifies le code pour avoir un fond de carte blanc ; un tracé en rouge d'un itinéraire à pied entre le parvis de Notre Dame de Paris et celui du parvis du Sacré cœur avec écrit en haut à droite de la carte Itinéraire vers le Sacré Cœur.

```
import osmnx as ox
import networkx as nx
import matplotlib.pyplot as plt

print ('Debut 3')

# 1. Coordonnées GPS
notre_dame = (48.8530, 2.3499)
sacre_coeur = (48.8867, 2.3431)

# 2. Télécharger le graphe piéton (rayon élargi pour Montmartre)
G = ox.graph_from_point(notre_dame, dist=4000, network_type="walk")

# 3. Trouver les nœuds les plus proches
start_node = ox.distance.nearest_nodes(G, notre_dame[1], notre_dame[0])
end_node = ox.distance.nearest_nodes(G, sacre_coeur[1], sacre_coeur[0])

# 4. Calcul de l'itinéraire le plus court
route = nx.shortest_path(G, start_node, end_node, weight="length")

# 5. Affichage avec fond blanc et itinéraire rouge
fig, ax = ox.plot_graph_route(
    G,
    route,
    route_color="red",
    route_linewidth=4,
    node_size=0,
    bgcolor="white",
    edge_color="lightgray",
    show=False,
    close=False
)

# 6. Ajouter le texte en haut à droite
ax.text(
    0.98, 0.98,
    "Itinéraire vers le Sacré Coeur",
    transform=ax.transAxes,
    ha="right",
    va="top",
    fontsize=12,
    color="black",
    bbox=dict(facecolor="white", edgecolor="none", alpha=0.8)
)

plt.show()

print ('Fin 3')
```

Nom :

On obtient alors :



Nom :

✚ Pour la seconde demande : Exemple de prompt :

Je souhaite désormais que tu modifies le code pour avoir un fond de carte blanc ; un tracé en bleu d'un itinéraire à pied entre le parvis de Notre Dame de Paris et l'entrée du Jardin du Luxembourg côté rue de Vaugirard avec écrit en bas à gauche de la carte Itinéraire vers le Luxembourg.

```
import osmnx as ox
import networkx as nx
import matplotlib.pyplot as plt

print ('Debut 4')

# 1. Coordonnées GPS
notre_dame = (48.8530, 2.3499)
luxembourg = (48.8483, 2.3372)

# 2. Télécharger le graphe piéton
G = ox.graph_from_point(notre_dame, dist=2500, network_type="walk")

# 3. Trouver les nœuds les plus proches
start_node = ox.distance.nearest_nodes(G, notre_dame[1], notre_dame[0])
end_node = ox.distance.nearest_nodes(G, luxembourg[1], luxembourg[0])

# 4. Calcul du plus court chemin à pied
route = nx.shortest_path(G, start_node, end_node, weight="length")

# 5. Affichage : fond blanc et itinéraire bleu
fig, ax = ox.plot_graph_route(
    G,
    route,
    route_color="blue",
    route_linewidth=4,
    node_size=0,
    bgcolor="white",
    edge_color="lightgray",
    show=False,
    close=False
)

# 6. Texte en bas à gauche
ax.text(
    0.02, 0.02,
    "Itinéraire vers le Luxembourg",
    transform=ax.transAxes,
    ha="left",
    va="bottom",
    fontsize=12,
    color="black",
    bbox=dict(facecolor="white", edgecolor="none", alpha=0.8)
)

plt.show()

print ('Fin 4')
```

Nom :

On obtient alors :



Note aux enseignants :

On peut poser des questions sur les deux False qui apparaissent dans `show=False` et `close=False` mais cela nous éloigne du sujet.

Les éléments de réponse sont liés au souhait d'écrire du texte sur une carte préalablement créée.

Pour pouvoir écrire sur une carte (c'est le bloc suivant dans les scripts), il faut que cette dernière ne soit pas fermée et qu'elle ne soit pas encore affichée. C'est donc la ligne `plt.show()` qui permettra son affichage (et dans ce cas, la carte apparaît avec le texte supplémentaire).

Le lien entre les deux lignes de code se fait à l'aide de l'objet `ax`.

```
fig, ax = ox.plot_graph route(G,  
    route,  
    route_color="blue",  
    route_linewidth=4,  
    node_size=0,  
    bgcolor="white",  
    edge_color="lightgray",  
    show=False,  
    close=False  
)
```

Nom :

```
ax.text(
    0.02, 0.02,
    "Itinéraire vers le Luxembourg",
    transform=ax.transAxes,
    ha="left",
    va="bottom",
    fontsize=12,
    color="black",
    bbox=dict(facecolor="white", edgecolor="none", alpha=0.8)
)
```

Et pour finir :

Si vous avez des questions sur cette proposition de TP, vous pouvez m'écrire à chdebernardi@yahoo.fr