

# Traitement d'informations publiques

Le projet proposé est l'exploitation de bases de données publiques. Les bases utilisées sont celles mises à disposition par la municipalité de Grenoble.



[http://data.metropolegrenoble.fr/ckan/dataset?res\\_format=GeoJSON](http://data.metropolegrenoble.fr/ckan/dataset?res_format=GeoJSON)

## Les arbres de Grenoble

Ce jeu de données contient toute les données relatives à l'identification et la localisation des arbres sur le territoire de la Ville de Grenoble aussi bien gérés par la Ville...

[GeoJSON](#) [KML](#) [CSV](#)

<http://data.metropolegrenoble.fr/ckan/dataset>

## Balades dans les massifs autour de Grenoble

Ces balades sont issues du site <http://www.Grenoble-Montagne.com> Elles sont classées par activité : Alpinisme Randonnée pédestre Raquette ski de randonnée et par massif ...

[JSON](#)

## Ces travaux nous permettrons :

- Découvrir les formats json et geojson dans un contexte professionnel.
- Extraire des informations des bases de données proposées.
- Exploiter les cartes au format Leaflet.
- Afficher des données prélevées dans les bases de données sur les cartes.
- Communiquer en présentant ses travaux: UML, algorithmes, essais et tests, et les résultats obtenus.

**Charme commun**

Plante

Le charme, ou charme commun, parfois appelé charmillle, est un arbre de taille moyenne, à feuilles caduques et appartenant à la famille des bétulacées. Il est assez répandu dans les forêts d'Europe centrale. Wikipédia

Nom scientifique : *Carpinus betulus*  
Famille : Betulaceae  
Classification supérieure : Charme  
Rang : Espèce  
Ordre : Fagales

Recherches associées [Voir d'autres éléments \(plus de 15\)](#)

Troène  
 Charme  
 Hêtre commun  
 Erable champêtre  
 Hêtres

Commentaires

# Les travaux à réaliser

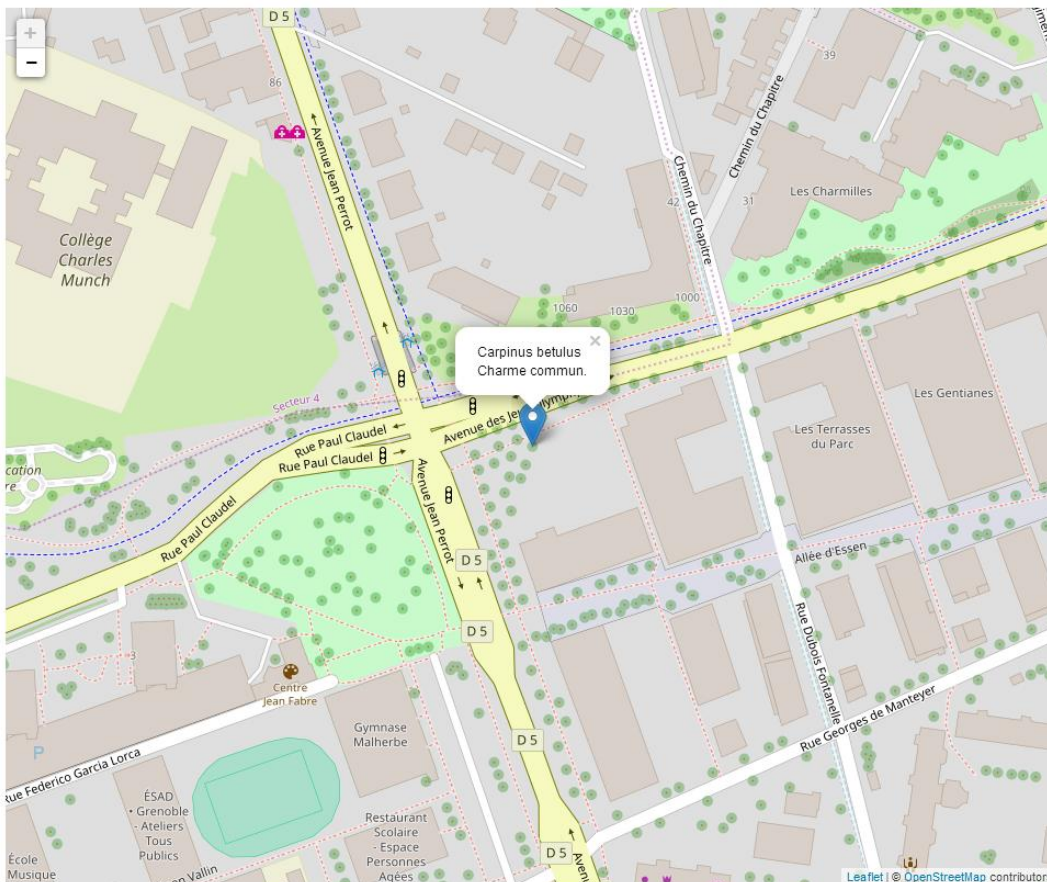
Exploitation de la base  ARBRES\_TERRITOIRE\_GRENOBLE.json

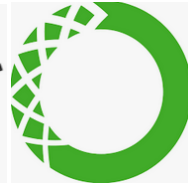
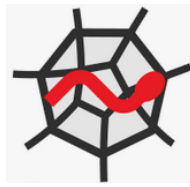
## Pour toutes les équipes séance 1:

Les tâches ci-dessous seront réparties entre les différents membres de l'équipe, la répartition sera donnée au professeur. Durée une séance de 2H. Les résultats seront préparés sous la forme d'un diaporama au format Libre Office ou Power Point.

- Tâche n° 1. Découvrir le format json et geojson et le présenter brièvement.
- Tâche n° 2. Présenter le contenu et l'organisation de la base.  
Indiquez sous quelle licence son exploitation est possible.
- Tâche n° 3. Découvrir le fonctionnement de Leaflet et le présenter brièvement.  
Présenter la méthode pour afficher des marqueurs sur les cartes Leaflet.
- Tâche n° 4. Proposer un algorithme permettant de déterminer la liste de toutes les espèces d'arbres présente dans la base avec le nom commun et le nom latin.

## Exemple de pointage d'un arbre particulier :

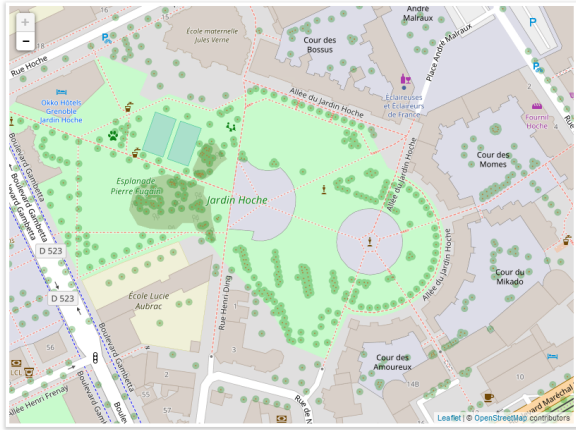




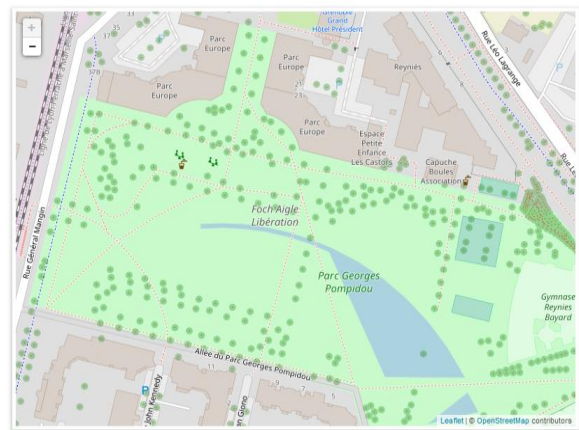
**Pour toutes les équipes séance 2 et 3 :**

Chaque équipe affichera sur la carte qui lui est attribuée une vingtaine d'arbres avec des marqueurs identiques à l'exemple précédent. Durée 2 séances. Les cartes sont les suivantes :

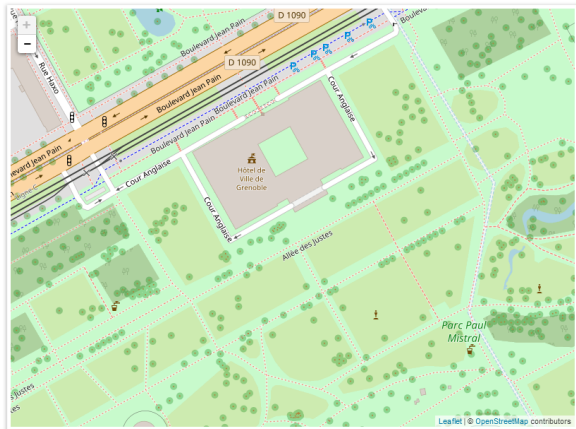
**Équipe 1 : Jardin Hoche (45.18427, 5.72786)**



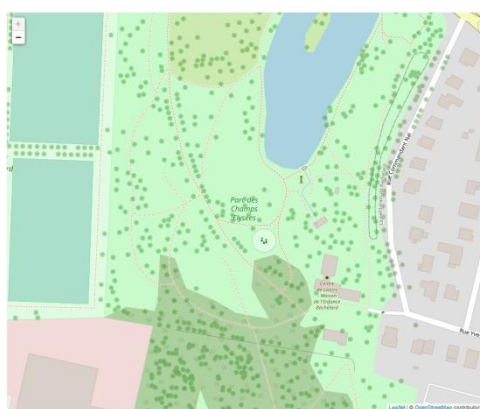
**Équipe 3 : Parc G. Pompidou (45.17722, 5.7199)**



**Équipe 2 : Hôtel de ville (45.18605, 5.73658)**



**Équipe 4 : Ensemble sportif (45.174845, 5.69007)**

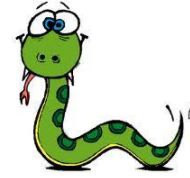
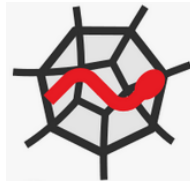


**Équipe 5 : Tour Perret (45.184933, 5.735448)**



**Pour toutes les équipes séance 4 :**








Chaque équipe préparera une présentation présentant ses travaux et résultats : algorithmes, essais et tests, résultats obtenus.







# Les outils et ressources disponibles

## 1 Listes des ressources

PROJET DATA\RESSOURCES

-  COURS HTML
-  COURS JAVASCRIPT
-  COURS JSON
-  COURS UML2
-  LOGICIEL JSONView
-  Classement des langages informatiques 2017.pdf
-  ICN-ISN LES LANGAGES DE PROGRAMMATION 2017.pdf

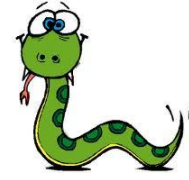
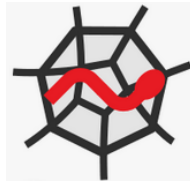
PROJET DATA\DEMO ET EXEMPLES

-  ACCES DATA JSON
-  AFFICHAGE JPG PYTHON
-  CARTES LEAFLET
-  ESSAIS JAVASCRIPT

### Sommaire des ressources :

1	Listes des ressources.....	4
2	Un viewer de fichier Json et GeoJson.....	5
3	Affichage d'une image jpg .....	6
4	Utilisation de JavaScript .....	8
5	Accéder aux données des tables json.....	9
6	Utilisation de Leaflet open source .....	10
7	Description de la conception en UML .....	Erreur ! Signet non défini.

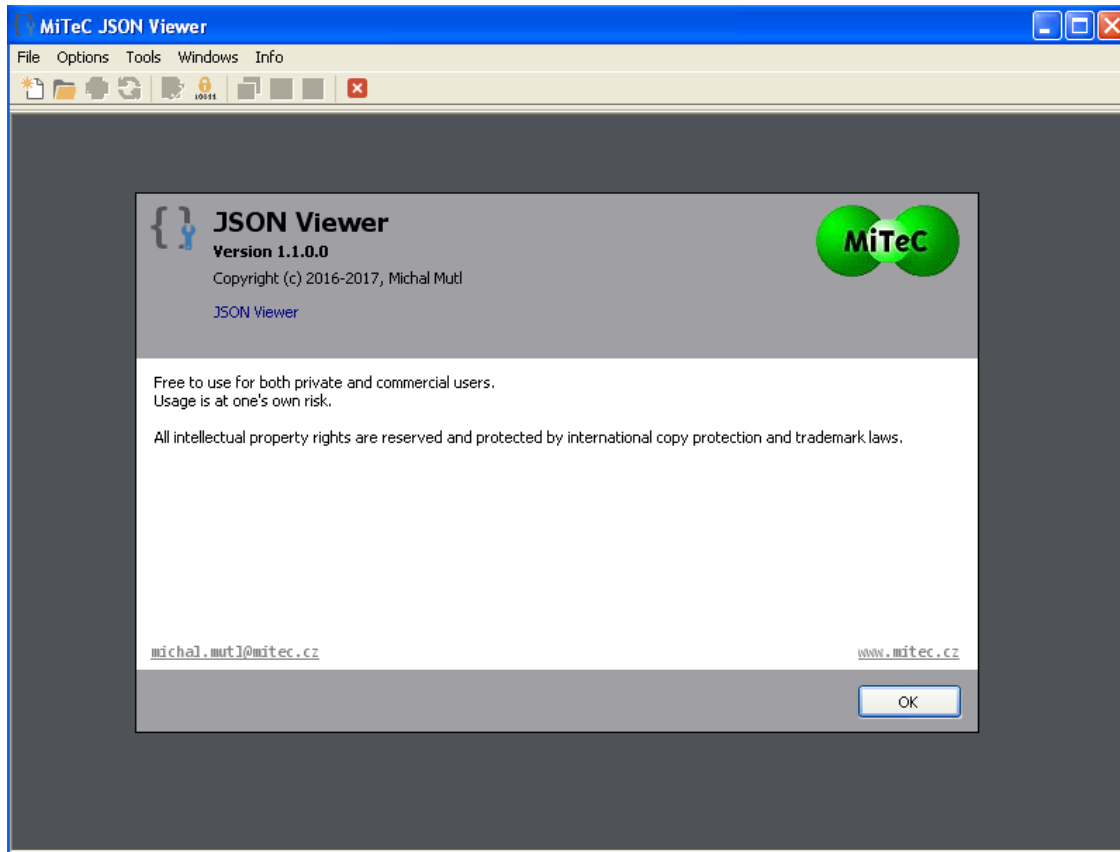


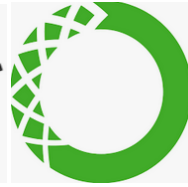
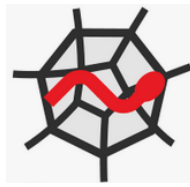


## 2 Un viewer de fichier Json et GeoJson

<https://www.mitec.cz/jsonv.html>

{ } JSONView.exe





### 3 Affichage d'une image jpg

<https://stackoverflow.com/questions/8213522/when-to-use-cls-clf-or-close-for-clearing-a-plot-in-matplotlib>

#### ▼ pyplot interface



`pyplot` is a module that collects a couple of functions that allow `matplotlib` to be used in a functional manner. I here assume that `pyplot` has been imported as `import matplotlib.pyplot as plt`. In this case, there are three different commands that remove stuff:

`plt.cla()` clears an axis, i.e. the currently active axis in the current figure. It leaves the other axes untouched.

`plt.clf()` clears the entire current figure with all its axes, but leaves the window opened, such that it may be reused for other plots.

`plt.close()` closes a window, which will be the current window, if not specified otherwise.

Which functions suits you best depends thus on your use-case.

The `close()` function furthermore allows one to specify which window should be closed. The argument can either be a number or name given to a window when it was created using `figure(number_or_name)` or it can be a figure instance `fig` obtained, i.e., using `fig = figure()`. If no argument is given to `close()`, the currently active window will be closed. Furthermore, there is the syntax `close('all')`, which closes all figures.

#### methods of the Figure class

Additionally, the `Figure` class provides methods for clearing figures. I'll assume in the following that `fig` is an instance of a `Figure`:

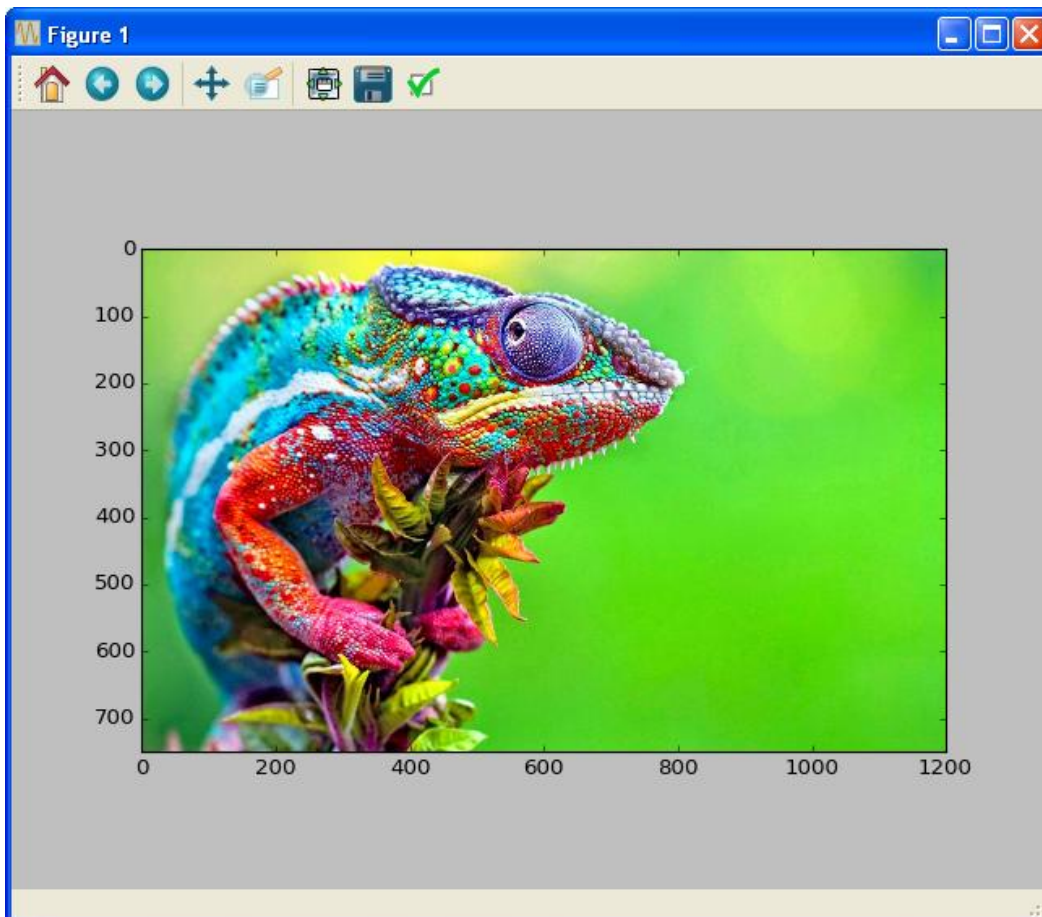
`fig.clf()` clears the entire figure. This call is equivalent to `plt.clf()` only if `fig` is the current figure.

`fig.clear()` is a synonym for `fig.clf()`

Note that even `del fig` will not close the associated figure window. As far as I know the only way to close a figure window is using `plt.close(fig)` as described above.



```
1 #!/usr/bin/env python
2 # -*- coding: utf-8 -*-
3
4 # Lecture_image_jpg_ok.py
5
6 from os import chdir
7 path = r'P:\PRO\USB\ICN2nD\Preparation ICN ISN 2017 2018\Python json'
8 chdir(path)
9
10 import imageio as mg
11 import matplotlib.pyplot as plt
12 import numpy as np
13
14 '''Attention, imageio transforme un tableau png en des nombres entiers de type jpg'''
15
16
17 cam=mg.imread(".\cameleon.jpg")
18
19 # visualisation de l'image
20 plt.clf()
21 plt.imshow(cam)
```



## 4 Utilisation de JavaScript

..\PROJET DATA\DEMO ET EXEMPLES\ESSAIS JAVASCRIPT

- 📁 JSON SUPER HEROS
- 📁 Manipuler des données JSON \_ MDN\_fichiers
- 📄 billiejoe\_javascript\_fiches.pdf
- 📄 Manipuler des données JSON \_ MDN.htm

Exemple de page entièrement créée par JavaScript

DOM : Document Object Model

# **SUPER HERO SQUAD**

Hometown: Metro City // Formed: 2016

### **MOLECULE MAN**

Secret identity: Dan Jukes  
Age: 29  
Superpowers:

- Radiation resistance
- Turning tiny
- Radiation blast

### **MADAME UPPERCUT**

Secret identity: Jane Wilson  
Age: 39  
Superpowers:

- Million tonne punch
- Damage resistance
- Superhuman reflexes

### **ETERNAL FLAME**

Secret identity: Unknown  
Age: 1000000  
Superpowers:

- Immortality
- Heat Immunity
- Inferno
- Teleportation
- Interdimensional travel

<https://developer.mozilla.org/fr/docs/Learn/JavaScript/Objects/JSON>






## 5 Accéder aux données des tables json

Travaux sur les données json

..\PROJET DATA\DEMO ET EXEMPLES\ACCES DATA JSON

 Lecture\_json.py

Exemple de donnée récupérée :

```
Arbre numéro 12000 référence > 30187
Genre > Platanus
Espèce > acerifolia
Longitude > 5.71445343204024
Latitude > 45.1752436660953
```

Code python :

```
json_data=open('..\ARBRES_TERRITOIRE_GRENOBLE.json')
tableDesArbres = json.load(json_data)
json_data.close()

print(type(tableDesArbres))

print('tableDesArbres.get("name") > ',tableDesArbres.get("name"))

print(len(tableDesArbres))

print(tableDesArbres.keys())

print(tableDesArbres['name'],'\n')

print(tableDesArbres['type'],'\n')

num = 1
print('Arbre numéro ',num,' référence > ',tableDesArbres['features'][num]['properties']['ELEM_POINT_ID'])
print('Genre > ',tableDesArbres['features'][num]['properties']['GENRE_BOTA'])
print('Espèce > ',tableDesArbres['features'][num]['properties']['ESPECE'])
print('Longitude > ',tableDesArbres['features'][num]['geometry']['coordinates'][0])
print('Latitude > ',tableDesArbres['features'][num]['geometry']['coordinates'][1])
print('\n')
num = 12000
print('Arbre numéro ',num,' référence > ',tableDesArbres['features'][num]['properties']['ELEM_POINT_ID'])
print('Genre > ',tableDesArbres['features'][num]['properties']['GENRE_BOTA'])
print('Espèce > ',tableDesArbres['features'][num]['properties']['ESPECE'])
print('Longitude > ',tableDesArbres['features'][num]['geometry']['coordinates'][0])
print('Latitude > ',tableDesArbres['features'][num]['geometry']['coordinates'][1])
```



## 6 Utilisation de Leaflet open source

Affichage d'une carte centrée sur les coordonnées

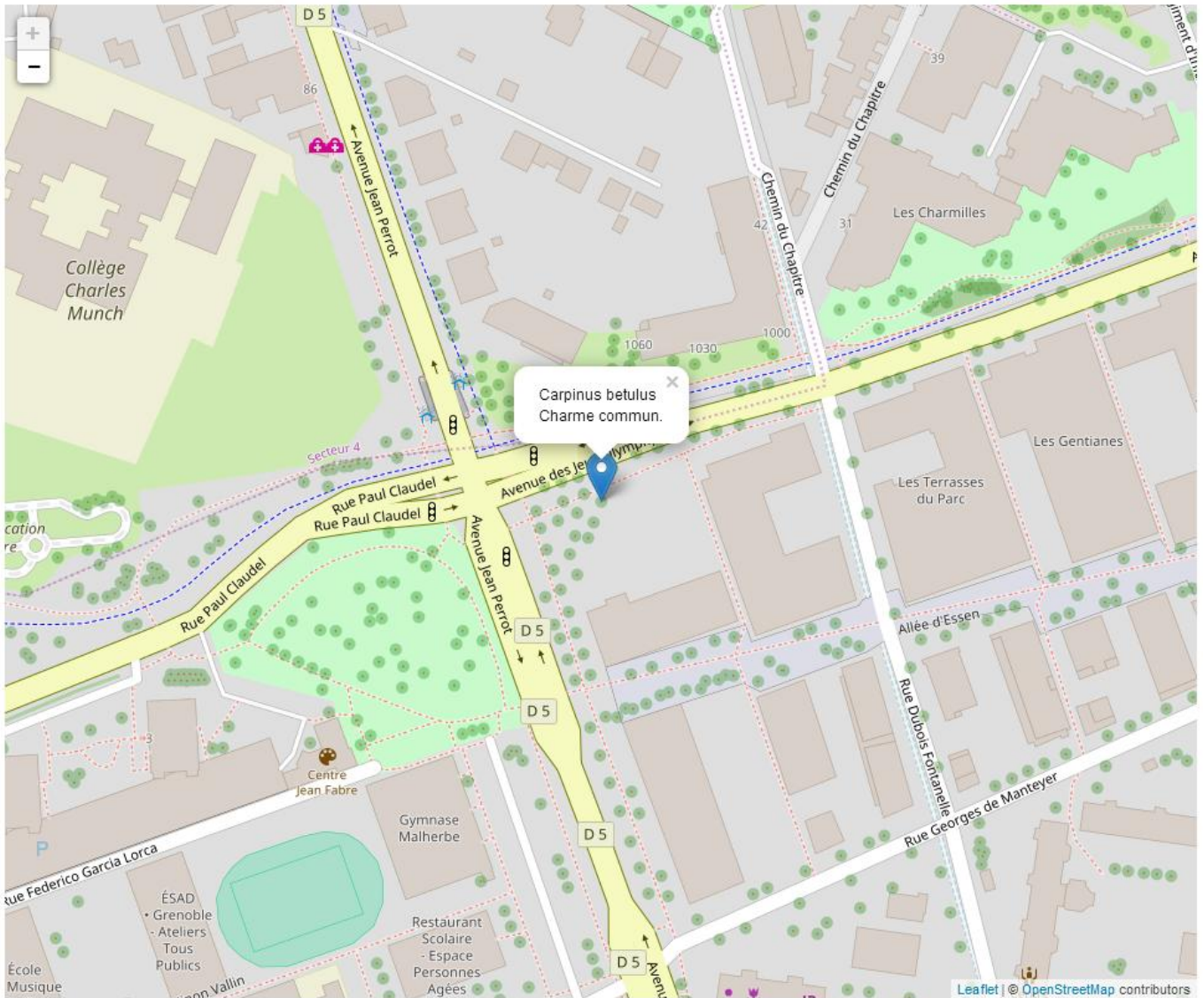
Latitude : 45.1730718638659

Longitude : 5.73922667253226

Avec un marqueur sur ces coordonnées et une bulle d'information sur l'espèce d'arbre.



Leaflet\_Ajout\_Un\_Arbre.html



Autre exemple plus sophistiqué :

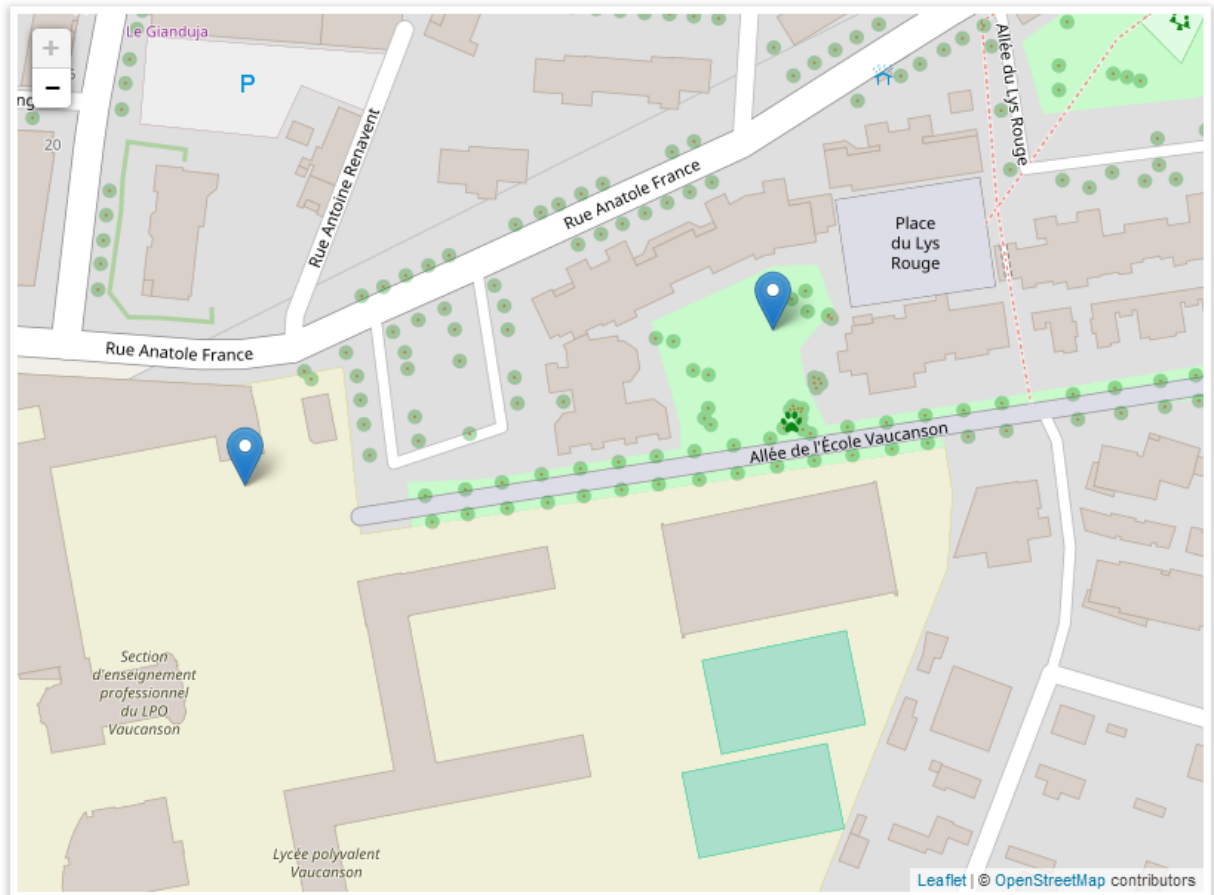
Leaflet\_Carte\_Marqueurs\_Interactive.html



## Markers avec context-Menu simple

Un clic sur la carte ajoute un marqueur à la carte.

Un clic droit sur le marqueur fait apparaître un context-menu.



Quelques liens :

<https://leafletjs.com/>

<https://zestedesavoir.com/tutoriels/1365/des-cartes-sur-votre-site/>

<https://www.supinfo.com/articles/single/205-leafletjs-s-amuser-avec-cartographie>

<https://leafletjs.com/download.html>

