

LATITUDE
FORMATION
RÉFÉRENTIEL GÉOGRAPHIQUE
CARTOGRAPHIE
ANALYSE SPATIALE
LONGITUDE
COORDONNÉES GÉOGRAPHIQUES
POLOGIQUE
FORMATION GÉOGRAPHIQUE
RASTER
VECTORISATION
LOGICIELS SIG
ATLAS
GPS
BDD
PROJET

Leaflet 



2D3D.GIS Organisme de formation
1 bis Perspective de l'Océan 17000 La Rochelle France



5

ATTRIBUTS

INFOBULLES ET ETIQUETTES



- Manipuler les attributs
- Afficher des infobulles / Interroger des couches
- Afficher des étiquettes

COURS

LES ATTRIBUTS

Le format GeoJSON permet de gérer les géométries mais également les attributs, le format intègre les deux composantes. Il est donc possible d'interroger les données sur la carte pour interroger les attributs.

Ces attributs peuvent être affichés dans des infobulles directement sur la carte ou affichés à tout autre endroit de la page HTML (sidebar, popup, etc.). Le javascript permettant d'interagir avec tous les éléments de la page.

EXERCICE

INFOBULLES

AU CLIC SUR LA COUCHE ROUTE

Nous allons afficher le nom de la voie dans une infobulle au clic sur une route dans la carte.

Commencez par faire afficher le nom de la rue dans la **console** :

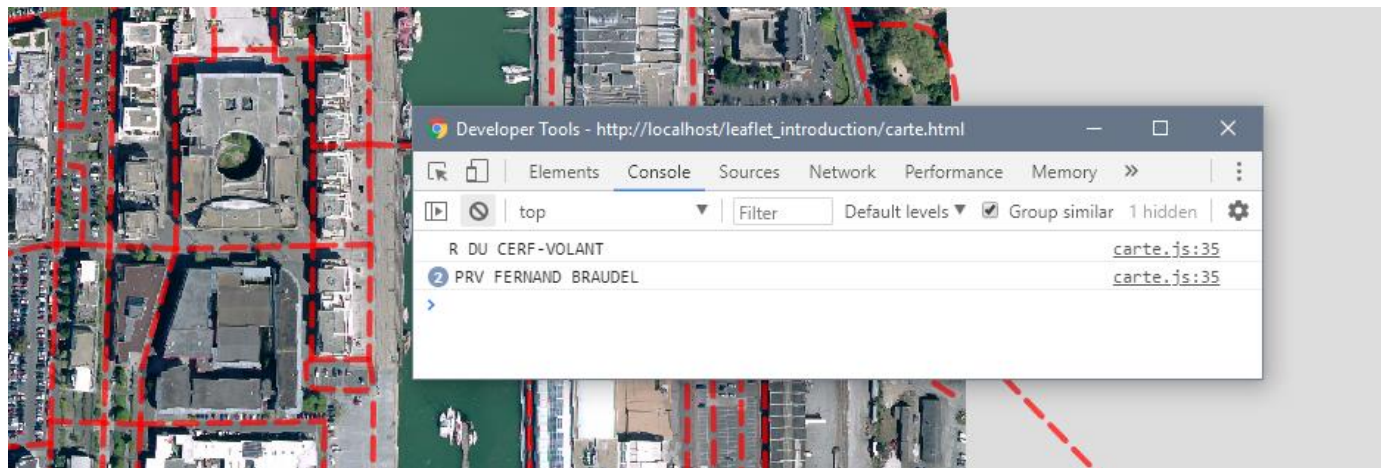
Pour cela, ajoutez un **event listener** `onEachFeature` click sur la couche des routes, à la suite du style de cette dernière :

`onEachFeature` (f, l) signifie que pour chaque entité de la couche, nous allons gérer un événement click qui déclenche une fonction permettant l'affichage de l'attribut `NOM_VOIE_G` dans la console.

```
/* Couche de lignes : route */
$.getJSON(
  "geojson/route_light.geojson",
  {},
  function(data) {
    vRoute = L.geoJSON(data, {
      style: function(f) {
        return {
          color: "#FF0000",
          weight: 4,
          opacity: 0.7,
          dashArray: '15,8'
        };
      }
    });
    onEachFeature: function (f, l) {
      l.on({
        click: function(e) {
          console.log(f.properties.NOM_VOIE_G);
        }
      });
    }
  });
  vRoute.addTo(map);
  lControl.addOverlay(vRoute, "Route");
}
);
```

Enregistrez le code et testez l'exécution.

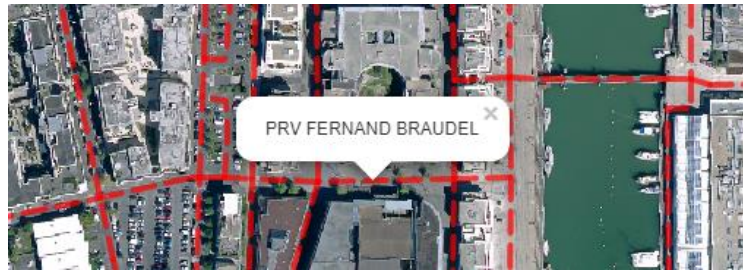
S'il est difficile de cliquer sur les routes, augmentez l'épaisseur du trait.



Remplacez ensuite la ligne de la console par la ligne suivante, permettant d'ouvrir la popup :

```
popup = L.popup({closeOnClick:false}).setLatLng(e.latlng).setContent(f.properties.NOM_VOIE_G).openOn(map);
```

A l'exécution, la popup apparait avec le nom de la rue :



Certaines routes n'ont pas de nom, (valeur null dans le fichier geosjon), il faut donc ajouter le test suivant pour n'afficher la popup que si l'attribut NOM_VOIE_G contient quelque-chose :

```
if (f.properties.NOM_VOIE_G) {
  popup = L.popup({closeOnClick:false}).setLatLng(e.latlng).setContent(f.properties.NOM_VOIE_G).openOn(map);
}
```

INFOBULLES AU SURVOL SUR LA COUCHE LIEU-DIT

En ajoutant un `onEachFeature`, nous allons déclencher la fonction `bindTooltip` qui permet l'ouverture d'une popup au survol d'une géométrie sur la carte.

```
/* Couche de points : lieu-dit */
$.getJSON(
  "geojson/lieu_dit.geojson",
  {},
  function(data) {
    vlieudit = L.geoJSON(data, {
      pointToLayer: function (feature, latlng) {
        return L.marker(latlng, {
          icon: new L.Icon({
            iconUrl: './img/icon.png',
            iconSize: [28, 28],
            iconAnchor: [14, 14]
          })
        });
      }
    });
    onEachFeature: function (feature, layer) {
      layer.bindTooltip(feature.properties.NOM, {permanent: false, sticky: false, opacity: 1, direction: "top"});
    }
  });
  vlieudit.addTo(map);
  lControl.addOverlay(vlieudit, "Lieu-dit");
}
);
```

L'option **sticky** permet de choisir si la popup suit le curseur de la souris de l'utilisateur à l'écran. La direction (top, bottom, left, right, center) permet quant-à-elle de choisir la position d'ouverture de la popup par rapport à l'endroit survolé.

Le résultat est le suivant :



Il est possible d'enrichir les popups avec du contenu HTML comme par exemple, utiliser un nom d'attribut et également mettre plusieurs attributs.

```
contenu = "<b>Nom : </b>" + feature.properties.NOM + "<br>";  
contenu += "<b>Type : </b>" + feature.properties.NATURE;  
layer.bindTooltip(contenu, {permanent: false, sticky: false, opacity: 1, direction: "top"});
```

A pour résultat :



EXERCICE EN AUTONOMIE

AFFICHER LES ATTRIBUTS DES SURFACES EN EAU

Pour cet exercice, vous allez ajouter une popup au **clic** sur les surfaces en eau, montrant l'ensemble des attributs (ID, NATURE, REGIME) présentés de cette manière :



ETIQUETTES

Le plus simple pour gérer l’affichage d’étiquettes est d’utiliser les **Tooltips** comme précédemment et de forcer leur affichage permanent. Pour ce qui est de l’apparence et du style des étiquettes, il suffit de les gérer en CSS grâce à une classe personnalisée **className**.

- Modifier l’ajout de la popup sur la couche de point pour la transformer en étiquette

```
onEachFeature: function (feature, layer) {
  contenu = feature.properties.NOM;
  layer.bindTooltip(contenu, {permanent: true, sticky: false, opacity: 1, direction: "top", className:"etiquetteLieuDit"});
}
```

La propriété permanent permet l’affichage des étiquettes.

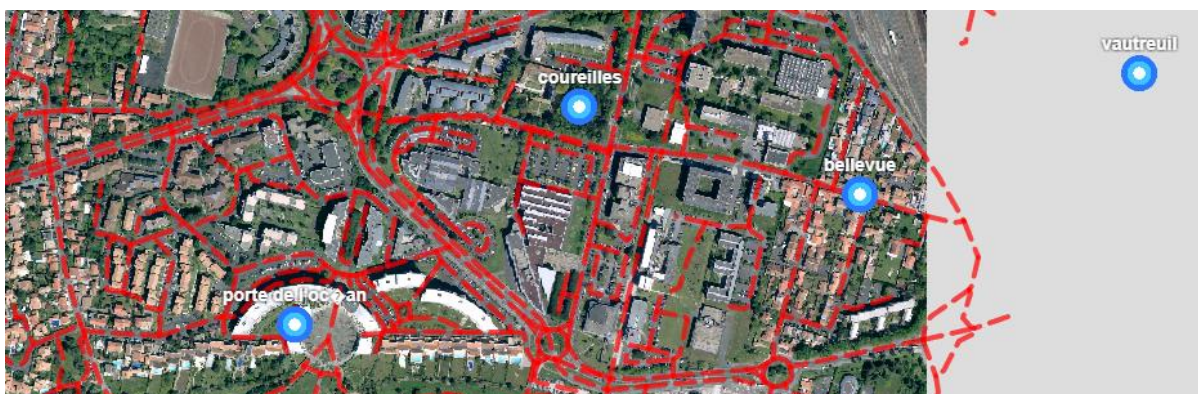


Pour changer le style des étiquette, la classe `etiquetteLieuDit` doit être définie dans le fichier CSS.

```
.etiquetteLieuDit {
  background:none;
  color:white;
  text-shadow:0px 0px 4px black;
  font-size:14px;
  font-weight:600;
  border:0px;
  box-shadow: none;
}
```

```
.etiquetteLieuDit::before {
  border-color:transparent;
}
```

Cette propriété `::before` (pseudo élément) permet de retirer la flèche blanche de la popup d’origine ;



6

PLUGINS



- Trouver les plugins utiles
- Télécharger et utiliser les plugins
- Apprendre à suivre une documentation de plugin

COURS

TROUVER DES PLUGINS

Les plugins permettent d'enrichir les fonctionnalités natives de Leaflet. Elles se présentent sous la forme de dossiers zippés contenant des fichiers JS, CSS et parfois images.

Le site officiel de Leaflet recense des plugins utiles et les plus couramment utilisés : <http://leafletjs.com/plugins.html>

Leaflet Plugins

While Leaflet is meant to be as lightweight as possible, and focuses on a core set of features, an easy way to extend its functionality is to use third-party plugins. Thanks to the awesome community behind Leaflet, there are literally hundreds of nice plugins to choose from.

Tile & image layers	Overlay Display	Map interaction	Miscellaneous
Basemap providers	Markers & renderers	Layer switching controls	Geoprocessing
Basemap formats	Overlay animations	Interactive pan/zoom	Routing
Non-map base layers	Clustering/decluttering	Bookmarked pan/zoom	Geocoding
Tile/image display	Heatmaps	Fullscreen	Plugin collections
Tile load	DataViz	Minimaps & synced maps	
Vector tiles		Measurement	Integration
	Overlay interaction	Mouse coordinates	Frameworks & build systems
Overlay data	Edit geometries	Events	3rd party
Overlay data formats	Time & elevation	User interface	
Dynamic data loading	Search & popups	Print/export	
Synthetic overlays	Area/overlay selection	Geolocation	
Data providers			Develop your own

La liste conséquente en propose plus d'une centaine mais d'autres peuvent se trouver en utilisant Google.

Pour trouver une extension utile, il faut rechercher par catégorie en fonction de ce que l'on veut faire.

7

PHP ET POSTGIS



- Utiliser une base de données spatiale
- Produire des fichiers GeoJSON en PHP

COURS

POURQUOI UTILISER PHP ET POSTGIS ?

Nous allons introduire deux nouvelles composantes à l'outil webmapping actuel, PHP et PostGIS. Ces outils sont indispensables pour plusieurs raisons :

- Les données attributaires comme spatiales ne sont pas figées dans des fichiers, il est donc nécessaire d'avoir une base de données pour en modifier le contenu.
- Une base de données spatiale permet le stockage de gros volumes de données et peuvent être requêtées (ex : j'ai l'ensemble des routes sur un département, je peux n'afficher que celles d'une commune).

Pour attaquer une base de données spatiale, il est obligatoire d'utiliser un langage serveur, ici PHP. Nous utiliserons également l'extension PDO qui permet de simplifier les accès bases de données pour PostgreSQL/PostGIS.

EXERCICE

CREER UNE BASE DE DONNES POSTGIS

En autonomie, créez une base de données PostGIS et y charger 3 couches de votre choix :

- Point
- Ligne
- Polygone

NB : La projection n'a pas d'importance mais n'oubliez pas de la renseigner lors de l'import des fichiers.

CREER UN SCRIPT PHP

A partir de vos connaissances, créez un script PHP permettant d'attaquer une table de la base de données pour en afficher le contenu sur une page HTML.

Enregistrez ce fichier dans un dossier **data**.