Atelier de géovisualisation

Cartographie interactive avec Leaflet (partie 1)

0 But de l'atelier

Le but de ce premier atelier est de découvrir comment on peut créer une carte interactive simple à l'aide de HTML, CSS, et Javascript. En passant, nous allons par la même occasion découvrir nos premiers éléments simples de HTML, CSS et Javascript, sans pour autant aller en profondeur.

Un aspect important de la cartographie interactive est que nous utilisons des ressources déjà existantes là où c'est possible. Ainsi, nous utiliserons la libraire Javascript Leaflet (http://leafletjs.com) qui permet de créer très facilement une carte interactive. Une *«librairie»* est une collection de fonctions cohérentes et qui est construite de sorte à être intégrée dans le code de quelqu'un d'autre afin d'atteindre un objectif précis et documenté.

Quant à Leaflet, il s'agit d'une libraire très populaire en géovisualisation. Elle permet d'inclure des données de sources différentes, comme p.ex. OpenStreetMap, Google, Swisstopo ou encore nos propres sources.

Dans cet atelier, nous allons produire une carte simple qui affichera un fond OpenStreetMap d'une région précise. Nous allons également y superposer une couche vectorielle.

Les étapes pour cet atelier sont les suivantes:

1. Écrire le code HTML de base pour une page Web (presque) vide.
2. Inclure la librairie Leaflet dans notre code HTML.
3. Écrire le code HTML pour la carte interactive, et définir la taille de la carte.
4. Écrire un peu de code Javascript pour créer la carte interactive et ajouter le fond de carte OpenStreetMap.
5. Paramétrer la carte pour afficher une région précise.
6. Ajouter une couche vectorielle à notre carte.

1 Code HTML de base

HTML est un langage informatique qui permet de décrire la structure d'une page Web. Le HTML est complété par du code CSS et Javascript, le premier étant pour définir le style de la structure HTML, et le deuxième pour ajouter l'interactivité.

Le code HTML permet de définir des structures comme un titre de niveau 1, 2, 3 etc., un paragraphe, un lien, un tableau, une image etc. Par contre, la présentation exacte est définie avec du CSS, donc par exemple la police d'écriture, la taille du texte, l'interligne, la couleur, la présence ou non d'une bordure, la distance aux autres éléments.

Le Javascript quant à lui est un langage de programmation qui est exécuté dans le navigateur Web. Donc Le HTML et le CSS sont des langages de description du contenu, tandis que le Javascript permet d'intégrer de modifier le contenu de la page Web en fonction d'événements comme des clics de souris etc. Une carte interactive nécessite de recourir aux trois langages informatiques.

Une page Web est toujours composée d'un fichier HTML. Le code CSS et Javascript peut se trouver dans le même fichier (avec extension .html ou .htm), ou dans des fichiers séparés et être inclus depuis le fichier HTML principal.

Un fichier HTML n'est rien d'autre qu'un simple fichier texte où l'extension a été changée en .html ou .htm. Regardons le contenu d'un fichier HTML de près. Voici un exemple extrêmement simple:

```
<!DOCTYPE html>
<html>
<head>
     <title>Ma première carte interactive</title>
</head>
<body>
     <h1>Ma carte interactive ou quelque chose de plus intelligent</h1>
</body>
</html>
```

Ce fichier est composé d'une série de *balises HTML*, et un peu de contenu. Une *balise HTML* est incluse à l'intérieur de signes **<...>**, donc p.ex. **<title>** est une balise HTML pour le titre du document. La plupart des balises HTML ont une balise de début et de fin, et ce qui est à l'intérieur est leur contenu. La balise de fin est incluse dans les signes **</...>**. Donc une balise complète pour le titre est:

```
<title>Le contenu du titre</title>
```

Le contenu de la balise titre sera affiché par le navigateur Web dans l'entête de la fenêtre ou de l'onglet.

Le document HTML lui-même est inclus dans la balise <html>...</html>. Chaque document HTML est organisé en une entête (la balise <head>...</head>) et le corps (la

balise <body>...</body>). L'entête contient des informations descriptives sur le contenu (le titre, l'encodage des caractères etc.) ainsi que du CSS ou Javascript (le code directement ou les lien vers les fichiers correspondants). Le corps contient tout ce qui est affiché directement sur une page Web.

Il est encore à noter que le contenu d'une balise peut contenir d'autres balises.

La première ligne d'un document HTML contient des informations sur la version du code HTML qui est utilisé. `<!DOCTYPE html>` correspond ainsi au HTML 5.

Voici une liste non exhaustive de quelques balises HTML fréquentes:

`<h1>...</h1>`	Titre de niveau 1
`<h2>...</h2>`	Titre de niveau 2
`<h7>...</h7>`	Titre de niveau 7
`<p>...</p>`	Un paragraphe de texte
`<i>...</i>`	Un texte en italique
`<b>...</b>`	Un texte gras (bold)

Certaines balises nécessitent l'indication d'un attribut, comme par exemple un lien:

```
<a href="http://unil.ch">Lien vers le site de l'UNIL</a>
```

Le texte affiché dans le navigateur est *«Lien vers le site de l'UNIL»,* et l'URL qui sera ouverte est indiquée dans l'attribut `href="..."`. Plusieurs attributs sont simplement séparés par un espace, comme par exemple:

```
<a href="http://unil.ch" target="_blank">Site de l'UNIL</a>
```

qui est un lien qui s'ouvre dans une nouvelle fenêtre.

Certaines balises n'ont pas besoin de balises de fermeture, comme p.ex. une image:

Dans ce cas, on peut écrire un / à la fin avant > pour indiquer qu'il n'y a pas de balise de fermeture.

Voici quelques balises supplémentaires fréquentes:

`<img src="..." />`	Une image où src indique le lien vers le fichier de l'image (JPEG, GIF ou PNG seulement)
`<a href="...">...</a>`	Un lien vers l'URL indiquée dans href
`<div>...</div>`	Un containeur vide, permet de regrouper un certain nombre d'autres éléments HTML. Est aussi utilisé pour donner un style commun aux éléments à l'intérieur.

Il y a encore beaucoup d'autres balises et d'attributs, mais que nous avons pas besoin de connaître à ce stade.

2 Code HTML pour la carte interactive

Écrivons maintenant notre fichier HTML pour une carte interactive. Pour cela, nous allons utiliser un éditeur de texte professionnel, dont voici quelques exemples:

- Sublime Text (http://www.sublimetext.com) pour OSX, Windows et Ubuntu
- Atom (https://atom.io) pour OSX et Windows
- TextMate (http://macromates.com) pour OSX

Par convention, nous allons nommer ce fichier **index.html**. En effet, le fichier de départ d'un site ou à l'intérieur d'un dossier est généralement appelé comme ça. Les parties qui changent par rapport à l'exemple précédent sont écrits en gras:

```
<!DOCTYPE html>
<html>
<head>
     <title>Ma première carte interactive</title>
     <link rel="stylesheet" href="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.css" />
     <script src="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.js"></script>
</head>
<body>
     <h1>Ma carte interactive ou quelque chose de plus intelligent</h1>
     <div id="map" style="width: 600px; height: 450px;"></div>
</body>
</html>
```

La balise `<link>` dans l'entête permet de charger un fichier de style CSS externe. En l'occurrence, il s'agit du style CSS faisant partie de la librairie Leaflet (http://leafletjs.com), qui nous permet de créer la carte interactive.

La balise `<script>...</script>` permet d'inclure du code Javascript. En l'occurrence, il s'agit d'un fichier externe (indiqué par l'attribut `src="..."`) et qui appartient également à Leaflet.

Le dernier changement est l'ajout d'une balise `<div>` dans le body du document. Cette balise définit l'espace qui sera prise par la carte interactive. Lorsque nous allons créer la carte interactive avec un petit script Javascript et en utilisant Leaflet, nous allons indiquer que la carte doit aller dans ce containeur. Pour pouvoir indiquer de quelle balise il s'agit, nous lui avons donné un identifiant avec `id="map"`. Finalement, dans l'attribut `style="..."` nous avons indiqué une largeur et hauteur du containeur. En fait, le contenu de cet attribut est du code CSS que nous regarderons plus tard plus en détail.

Ouvrez maintenant votre fichier HTML avec un navigateur Web et regardez le résultat. Il ne doit pas y avoir de carte, mais au moins un titre. C'est déjà pas mal...

Un problème qui peut apparaître si nous avons des accents dans le texte est que leur affichage n'est pas correct. C'est dû à ce qu'on appelle l'encodage du texte qui n'est

pas encore défini dans notre fichier HTML. Du coup nous le rajoutons quelque part dans l'entête de notre document HTML (rajouter la partie en gras):

```
<html>
<head>
     ...
     <meta charset="utf-8" />
</head>
<body>
     ...
</body>
</html>
```

Un mot encore par rapport à l'indentation (c'est le fait que certaines lignes sont décalées à droite). L'indentation est faite avec un tabulateur ou des espaces et sert uniquement à l'augmentation de la lisibilité du code par un être humain.

Afin d'avoir un code lisible et bien maintenu, il est essentiel, important, crucial et primordial de faire une indentation correcte! Ceci vous évitera des maux de tête inutiles!

3 Carte interactive et la première couche

Il est maintenant temps de rendre notre carte interactive opérationnelle. Il nous manque juste une petite étape qui est l'écriture d'un petit code Javascript.

Nous insérons le code Javascript directement dans notre fichier HTML (dans un projet un peu plus important, on séparera le code Javascript dans un fichier séparé; c'est pour plus tard...). Le code Javascript sera inséré à l'intérieur d'une balise `<script>...</script>` (en gras), après la balise `<div>...</div>` qui accueillera la carte:

```
<!DOCTYPE html>
<html>
<head>
     <meta charset="utf-8" />
     <title>Ma première carte interactive</title>
     <link rel="stylesheet" href="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.css" />
     <script src="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.js"></script>
</head>
<body>
     <h1>Ma carte interactive ou quelque chose de plus intelligent</h1>
     <div id="map" style="width: 600px; height: 450px;"></div>
     <script type="text/javascript">
     ...
     </script>
</body>
</html>
```

Notre code fera 2 choses:

1. Créer la carte interactive en spécifiant les options souhaitées
2. Ajouter la couche OpenStreetMap à la carte interactive

La première chose peut être faite avec la commande suivante:

```
var map = new L.Map('map', {
      center: [47, 8],
      maxBounds: [[46, 6], [48,10]],
      minZoom: 4,
      maxZoom: 8,
      zoom: 6
});
```

Il s'agit ici d'une seule commande qui est sur plusieurs lignes. Une commande peut en effet être sur une ligne, ou sur plusieurs à condition qu'il y a une parenthèse non fermée. En plus, une commande se termine généralement sur un point-virgule (en réalité c'est optionnel mais c'est un bonne pratique car c'est plus clair). Ces instructions pourraient très bien être écrites sur une seule ligne, mais la séparation sur plusieurs lignes augmente la lisibilité du script. **La lisibilité est notre assurance-vie** lors de l'écriture du code; faites tout ce qui est possible pour avoir un code lisible! De toute manière, vous allez faire des fautes dans votre code, et si votre code est mal lisible, vous allez passer beaucoup plus de temps à trouver vos erreurs…

Pour l'instant, les choses qu'il faut comprendre sont:

- La nouvelle carte est stockée dans une **variable** `map` (`var map = ...`)
- `L.Map()` est une **fonction**, les choses entre les parenthèses sont les arguments de la fonction.
- Le premier arguments à `L.Map` est `'map'` qui correspond à l'id de la balise où ira la carte.
- La partie entre accolades `{...}` définit les options de la carte, comme la position du point central (`center`; à 47° latitude et 8° longitude), l'étendue maximale (`maxBounds`, à nouveau en latitude/longitude), le niveau minimal et maximal possible de zoom (`minZoom`, `maxZoom`), ainsi que le niveau de zoom initial lors du chargement de la carte (`zoom`). (Il y aura plus de détails sur comment ça fonctionne plus tard).

Pour l'instant, ce code créé une carte, mais n'affiche encore rien puisque nous n'avons pas ajouté de couche. Nous allons donc créer la couche OpenStreetMap:

```
var osmLayer = L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png', {
      attribution: '&copy; OpenStreetMap contributors'
});
```

Ceci crée une couche OpenStreetMap dont nous avons spécifié l'URL (d'une manière un peu spéciale, plus sur cela dans un autre cours), ainsi que l'attribution du copyright

qui sera visible en bas à droite de la carte interactive (le `&copy;` étant le code HTML pour le signe ©). Par contre, ce code ne permet toujours pas d'afficher la carte, puisque nous n'avons pas encore donné l'instruction d'ajouter cette couche à la carte (il n'y a pas de lien entre la variable `map` et la variable `osmLayer`). Pour faire cela, nous devons ajouter encore une troisième commande:

```
osmLayer.addTo(map);
```

Voici le code complet de notre fichier **index.html**:

```
<!DOCTYPE html>
<html>
<head>
     <meta charset="utf-8" />
     <title>Ma première carte interactive</title>
     <link     rel="stylesheet"     href="http://cdn.leafletjs.com/leaflet-0.7.3/
leaflet.css" />
     <script src="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.js"></script>
</head>
<body>
     <h1>Ma carte interactive ou quelque chose de plus intelligent</h1>
     <div id="map" style="width: 600px; height: 450px;"></div>
     <script type="text/javascript">
  var map = new L.Map('map', {
     center: [47, 8],
     maxBounds: [[46, 6], [48,10]],
     minZoom: 4,
     maxZoom: 8,
     zoom: 6
     });

     var osmLayer = L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png', {
     attribution: '&copy; OpenStreetMap contributors'
     });

     osmLayer.addTo(map);
     </script>
</body>
</html>
```

Rechargez maintenant votre fichier index.html complet dans le navigateur Web. Vous devriez voir la carte OSM:

Voici notre premier carte interactive!

Dans les ateliers à venir, nous allons expliquer plus en détail le HTML, CSS et Javascript pour ensuite créer une carte interactive Leaflet un peu plus sophistiquée.

Et n'oubliez pas:

Si ça ne marche pas, c'est normal,
si ça marche, c'est exceptionnel,
et gardez votre bonne humeur dans les deux cas !