

### 3.5.1. Présentation de Leaflet JS

Nous avons évoqué brièvement Leaflet dans le [précédent chapitre](#), avec la commande `gdal2tiles`. Leaflet (<https://leafletjs.com/>) est une bibliothèque Javascript open source qui permet d'intégrer des cartes interactives dans une page web.

Les types de données suivants sont supportés par Leaflet :

- **données raster tuilées** (exemple : tuiles OpenStreetMap, IGN Géoportail, images satellitaires Google Maps, webservices WMTS),
- **données vectorielles** au format **GeoJSON**,
- images issues de web services **WMS**.

Côté fonctionnalités, avec l'API Leaflet vous pourrez notamment :

- associer aux entités de type vectoriel des **popups** (bulles d'information) et des **styles** (symboles et couleurs basés sur des attributs),
- afficher une **liste de couches**,
- afficher une **légende**.

❗ Leaflet dans R

La version originale de l'API Leaflet est basée sur le langage Javascript, le principal langage de script pour les pages web. Il existe aussi une "adaptation" de Leaflet sur R, langage axé sur l'analyse et la visualisation de données. Si vous programmez avec R et que vous souhaitez en savoir plus, consultez le site <https://rstudio.github.io/leaflet/>.

## Une autre API illustre : OpenLayers

Leaflet est relativement simple dans son utilisation et dans ses fonctionnalités. **OpenLayers** (<https://openlayers.org/>) est une autre bibliothèque Javascript open source pour la cartographie. Openlayers permet de développer des interfaces utilisateurs riches, avec par exemple des outils d'interrogation des données et des outils de dessins pour l'édition des données.

### Organisation des TP Leaflet

Dans les TP qui suivent, nous allons créer des pages web dans le répertoire `/home/sist/data/mon_travail/3_leaflet`. Le serveur web Apache de la machine virtuelle est configuré pour que l'URL [http://localhost/3\\_leaflet/](http://localhost/3_leaflet/) pointe sur ce répertoire. Vous pourrez ainsi modifier les fichiers HTML et CSS avec l'éditeur **Visual Studio Code** (ou un autre éditeur de votre choix) et voir le résultat directement dans Firefox.

Le répertoire `/home/sist/data/mon_travail/3_leaflet` contient 2 modèles de page HTML que vous pouvez utiliser pour les TP :

- `modele_simple.html` ([http://localhost/3\\_leaflet/modele\\_simple.html](http://localhost/3_leaflet/modele_simple.html)), un modèle de page vierge
- `modele_2colonnes.html` ([http://localhost/3\\_leaflet/modele\\_2colonnes.html](http://localhost/3_leaflet/modele_2colonnes.html)), un modèle de page à 2 colonnes qui permet, par exemple, de juxtaposer du texte et une carte.

TP : Squelette de la page

Dans l'éditeur de code **Visual Studio Code**, ouvrez le répertoire `/home/sist/data/mon_travail/3_leaflet` avec le menu *File / Open folder...*, puis affichez la page `modele_simple.html`.

Observez l'entête de la page (balise head).

Les feuilles de styles (`normalize.css` et `skeleton.css`) utilisées dans ce modèle sont issues du site <http://getbootstrap.com/> - ce qui permet d'obtenir un rendu à la fois minimaliste, élégant, identique quel que soit le navigateur et "responsive".

Puis, observez le corps de la page (balise `body`).

Il comporte un conteneur principal (balise **div**). La classe **container** appliquée à cette balise permet de centrer la page dans le navigateur.

Les balises **h1**, **h2**, **p** définissent respectivement un titre de niveau 1, un titre de niveau2, un simple paragraphe.

La balise `a` est utilisée pour les liens hypertextes.

Changez le titre (**h1**) et le texte (**p**) dans le corps de la page, et rafraîchissez la page dans Firefox.

## Liens utiles

- <https://developer.mozilla.org/fr/docs/Apprendre>. Le site MDN de Mozilla (documentation de référence pour les développeurs web, tutoriels)
- <https://www.grafikart.fr/formations/debuter-javascript>. Formation Javascript sur Grafikart
- <https://eloquentjavascript.net/>. Un ouvrage d'initiation au Javascript et à la programmation

### 3.5.2. Une carte avec Leaflet

Les étapes pour insérer une carte Leaflet dans une page HTML sont les suivantes.

## Entête de la page

Dans l'entête de la page (entre les balises **head**), vous devez insérer une référence au fichier CSS de Leaflet. Insérez également une référence au fichier CSS personnalisé `anf_sist.css` que vous utiliserez pour régler les éléments de la page.

```
<link rel="stylesheet" href="css/anf_sist.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/leaflet/1.7.1/leaflet.css">
```

Toujours dans l'entête, vous devez insérer une référence au fichier Javascript de Leaflet après le fichier CSS.

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/leaflet/1.7.1/leaflet.js"></script>
```

#### Utilisation de CDN

Vous pouvez, si vous le souhaitez, télécharger et héberger sur votre site les fichiers de l'API Leaflet. Cependant, il est plus courant de faire appel à un "CDN" (site tiers) comme <https://cdnjs.com/> pour optimiser le temps de chargement de la page.

### Corps de la page

Dans le corps de la page HTML (quelque part entre les balises **body**), un conteneur avec les balises **div** doit être inséré : c'est l'espace où la carte s'affichera.

```
<div id="mapid"></div>
```

Pour définir les dimensions de la carte, ajoutez un nouveau sélecteur CSS dans le fichier `css/anf_sist.css` : dans notre exemple, le sélecteur **#mapid** fait référence à l'identifiant du **div** que nous venons d'ajouter à la page.

```
#mapid {  
  height: 240px;  
}
```

Toujours dans le corps de la page HTML (vers la fin), insérez le code Javascript, entre les bases **script**. Voici un exemple de carte avec un fond de carte OSM.

```
<script>  
  // javascript : une carte Leaflet centrée sur Sète  
  var lat_lng_lazaret = [43.3935, 3.6734];  
  var opt_carte = {  
    center: [43.402, 3.696],  
    zoom: 12  
  };  
  var ma_carte = L.map('mapid', opt_carte);  
  var lyr = L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {  
    attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contri  
  });  
  lyr.addTo(ma_carte);  
</script>
```

Sur la page [Tutorials](#) du site de Leaflet, vous pouvez trouver un exemple de carte assez similaire avec *Quick start guide*.

#### TP : une carte de Sète.

Dans ce TP, vous allez partir du modèle de page proposé avec `modele_simple.html`, et y ajouter une carte de Sète.

Dans **Visual Studio Code**, commencez par enregistrer le modèle de page HTML sous le nom `tp_lazaret.html`, dans le répertoire `3_leaflet`.

Dans l'entête, ajoutez les références à `anf_sist.css`, `leaflet.css` et `leaflet.js`.

Dans le corps de la page, ajoutez un **div** avec l'identifiant `mapid`.  
Vers la fin du fichier, juste avant la balise fermante `body`, ajoutez le code Javascript qui initialise la carte (cf. ci-dessus).

Enfin, ouvrez le fichier `anf_sist.css` et ajoutez un nouveau sélecteur CSS **#mapid**.

Enregistrez les fichiers HTML et CSS. Il est maintenant le temps de tester votre page ! Dans Firefox, rendez-vous sur [http://localhost/3\\_leaflet/tp\\_lazaret.html](http://localhost/3_leaflet/tp_lazaret.html)

### JavaScript : rappels et analyse du code

Analysons le code JavaScript ci-dessous. Dans les lignes qui commencent par **var**, nous initialisons des objets.

```
var lat_lng_lazaret = [43.3935, 3.6734]; // un tableau à 2 éléments  
var opt_carte = {  
  center: [43.402, 3.696],  
  zoom: 12  
}; // un objet avec 2 attributs sous forme de clé / valeur
```

Certains objets sont créés avec une fonction qui commence par "L." d'après des **classes** (modèles d'objet) propres à Leaflet : **L.map**, **L.tileLayer**.

```
var ma_carte = L.map('mapid', opt_carte);  
var lyr = L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {  
  attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributor  
});  
lyr.addTo(ma_carte);
```

**L.map** est la classe la plus importante de Leaflet. Elle sert à créer une carte interactive "vierge" dans la page. Elle renvoie ici un objet `ma_carte` que nous allons réutiliser dans le code. **L.tileLayer** sert à créer une couche raster tuillée, que nous ajoutons ensuite à `ma_carte` avec la méthode **addTo** (une **méthode** est une fonction qui s'applique à un objet).

Les **paramètres** nécessaires à la création des objets ainsi que les **méthodes** figurent dans la [documentation officielle](#) de Leaflet.

Par exemple, la fonction **L.map** a 2 paramètres : 1) une chaîne avec l'ID du div, et 2) un objet pour les **options**. L'objet `opt_carte` comporte ici 2 options sous forme d'attributs : **center** (un tableau) et **zoom** (un entier).

**Exercice 2.** Entraînez-vous à lire le code et la documentation de Leaflet.

Ouvrez la [page de documentation de la classe tileLayer](#).

La création d'une couche tuillée se fait avec **L.tileLayer** qui prend 2 paramètres.

Dans le code vu précédemment, que signifient **[s]**, **[z]**, **[x]** et **[y]** dans l'argument 1 ? Quels sont le **nom** et le **type** de l'option passée dans l'argument 2 ?

[Voir la solution](#)

### Ajout d'un marqueur

Nous verrons dans le chapitre suivant comment ajouter une couche de marqueurs issus d'un fichier

GeoJSON. Contentons-nous pour l'instant d'ajouter un seul marqueur.

De la même façon qu'avec les couches, il faut :

1. créer un objet, avec la fonction `L.marker`
2. l'ajouter à la carte avec la méthode `addTo`

Dans un 3ème temps, il est possible de créer une **popup** liée au marqueur (bulle de texte qui apparaît quand on clique sur le symbole) avec la méthode `bindPopup`.

**Exercice 3.** Un marqueur sur le Lazaret avec une popup.

Vous allez ajouter dans le code JavaScript de la page `tp_lazaret.html` 2 lignes, pour afficher un marqueur à l'emplacement du Lazaret. Les coordonnées du Lazaret figurent dans `lat_lng_lazaret`. Inspirez-vous des 2 lignes suivantes et complétez-les.

```
var marker = L.marker(______);  
marker.addTo(______).bindPopup("<strong>ANF SIST 2020</strong><br>Le Lazaret, Sète")
```

[Voir la solution](#)

### Pour aller plus loin : d'autres fonds de carte

Vous pouvez expérimenter d'autres fonds de carte que les tuiles OSM, avec les liens ci-dessous.

- Tester la [clé IGN Géoportail en libre accès](#) en vous inspirant de l'exemple Leaflet présenté sur la page
- Tester les fonds de carte de la société Stamen : <http://maps.stamen.com/>
- Voir les fonds de carte proposés par Leaflet-Providers : <https://leaflet-extras.github.io/leaflet-providers/preview/>

### 3.5.3. A RETENIR

- Leaflet est une bibliothèque Javascript opensource qui permet d'intégrer une cartes interactive dans une page web.
- Leaflet est relativement simple dans ses fonctionnalités et son utilisation, ce qui la rend idéale pour visualiser des données.
- OpenLayers est une autre bibliothèque répandue pour l'intégration de carte interactive dans une page web, qui demande un temps d'apprentissage plus long pour maîtriser ses nombreuses fonctionnalités.
- Les données raster tuilées ainsi que les données vectorielles GeoJSON sont les formats utilisés de préférence avec Leaflet.
- L'intégration d'une carte dans une page HTML se fait en insérant dans l'entête un lien vers `leaflet.css` et vers `leaflet.js`, et dans le corps une balise `div` et une section `script`.
- Dans la partie Javascript, la classe d'objet `L.map` est utilisée pour créer une carte. La classe `L.tileLayer` permet d'ajouter une couche de type raster tuilé, par exemple un fond de carte OpenStreetMap.
- Toujours en Javascript, `L.marker` est utilisé pour afficher un marqueur (symbole ponctuel cliquable), auquel peut être associé un `L.popup`.

[Previous](#)

[Next](#)