

Ce document présente et détaille le travail réalisé, fichier par fichier, pour le devoir maison de développement web. On y retrouvera le détail de certaines parties du code (qui a également été commenté dans chaque fichier).

Fichier index.html

Le fichier index.html décrit la structure de la page web. Il suit la structure de base d'un fichier HTML, et comporte **2 div dans le body** : une que l'on appelle « **position** », et l'autre que l'on appelle « **distance** ». Ces 2 div serviront à afficher les résultats de la géolocalisation récupérés par le script contenu dans le fichier geolocalisation.js (importé dans le « head ») :

```
1 <!DOCTYPE html>
2 <html lang="fr">
3
4   <head>
5     <title>API HTML5 Géopositionnement</title>
6     <meta name="Viewport" content="width=device-width, user-scalable=no, initial-scale=1.0">
7     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
8     <link rel="stylesheet" href="style.css" type="text/css" />
9     <script src="geolocalisation.js"></script>
10  </head>
11
12  <body>
13    <h1>Position et distance</h1>
14    <div id="position"></div>
15    <hr>
16    <div id="distance"></div>
17  </body>
18
19 </html>
```

Aperçu du code du fichier index.html

Le code a également été validé par l'outil proposé par W3C (World Wide Web Consortium) afin de vérifier la structure des fichiers HTML :

Document checking completed. No errors or warnings to show.

Résultat de l'analyse du fichier avec l'outil « Markup Validation Service » de W3C

Fichier geolocalisation.js

Le fichier `geolocalisation.js` contient le script qui permet de **récupérer les données de géolocalisation de l'utilisateur** à partir de l'API HTML5 Geolocation. Il est composé de plusieurs fonctions :

- **`coordinate(latitude, longitude)`** : cette fonction permet de créer, d'une certaine manière, un objet de « coordonnées » similaire à celui retourné par l'API Geolocation (comme l'objet « `position.coords` »).
- **`geoSuccess(position)`** : cette fonction remplit les div du document HTML avec les données récupérées depuis l'API. Cette fonction utilise également la fonction « `calculDistance` » pour calculer la distance entre l'utilisateur et l'ESIREM.
- **`geoError(error)`** : cette fonction est « throw » lorsqu'une erreur se produit lorsqu'on utilise la fonction « `getCurrentPosition` » de « `geolocation` ». Elle indique à l'utilisateur (dans les div du document) qu'il n'est pas possible de récupérer sa position, et affiche également l'erreur.
- **`calculDistance(startCoords, destCoords)`** : cette fonction calcule et retourne la distance (en km) entre deux coordonnées passées en paramètre de la fonction.
- **`degresEnRadians(degres)`** : cette fonction convertit et retourne une valeur en degrés (passée en paramètre) en radians.

Le fichier comporte également la méthode « `window.addEventListener()` » : elle est **configurée pour s'exécuter lors de l'évènement « load »** qui se produit lorsque toutes les ressources de la page web ont été chargées.

Dans cette méthode, on vérifie d'abord que l'API Geolocation est disponible à l'aide d'une simple condition. **Si elle est validée, on utilise l'API par le biais de la méthode « `getCurrentPosition()` »**. Si elle n'est pas validée, on exécute la fonction « `geoError()` », et on avertit l'utilisateur avec une **alerte JavaScript** « `window.alert()` » que la géolocalisation n'est pas disponible avec son navigateur.

Une **variable « `options` »** sert à **paramétrer l'API Geolocation** avec les paramètres suivants : « **`enableHighAccuracy`** » défini sur « **`true`** » pour obtenir les meilleurs résultats possibles, ainsi que « **`timeout`** » défini sur « **`9000`** » pour définir la durée maximale (en millisecondes) que le périphérique est autorisé à prendre pour renvoyer une position. Cette variable est utilisée lors de l'appel à la méthode « **`getCurrentPosition()`** ».

Le script JavaScript a été vérifié à l'aide de l'outil de chez JSHint : <https://jshint.com>.

Fichier style.css

Le fichier style.css est divisé en deux media queries :

- La première s'exécute lorsque la taille de l'écran est **inférieure à 600px**
- La seconde s'exécute lorsque la taille de l'écran est **supérieure à 600px**

Il reste néanmoins un sélecteur de classe indépendant des media queries que l'on appelle « boldResult ». Ce sélecteur est **utilisé pour surligner les résultats de l'API en gras (à l'aide de balises)**, il est utilisé dans tous les cas :

```
1  /* Classe pour mettre les résultats en gras */
2  .boldResult {
3      font-weight: bold;
4  }
```

Aperçu du sélecteur de classe boldResult

La feuille de style CSS a été validée à l'aide de l'outil de chez W3C :

<https://jigsaw.w3.org/css-validator/>.