

</>

Rémi - Xavier - Romain

Présentation de notre projet !

Map City Weather

Juin 2025

```
    }
    render() {
      return (
        <React.Fragment>
          <div className="py-5">
            <div className="container">
              <Title name="our" title="Our products" />
              <div className="row">
                <ProductConsumer>
                  {({value}) => {
                    console.log(value);
                  }}
                </ProductConsumer>
              </div>
            </div>
          </div>
        <React.Fragment>
```



Sommaire

01. Mise en oeuvre de l'exercice & les outils

02. Structure du projet

03. HTML & CSS

04. Javascript

05. Visuel final

06. Remerciements

</>

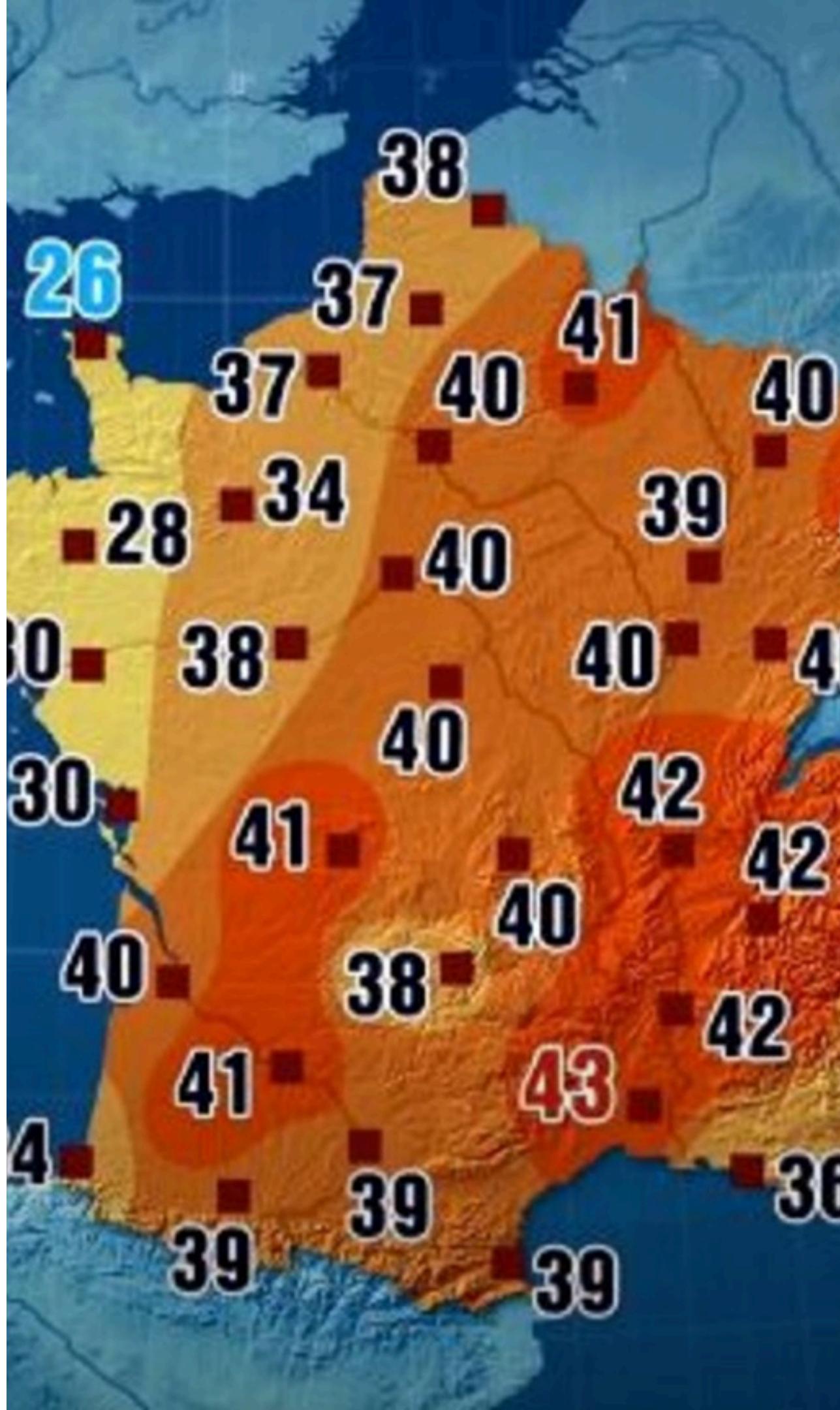
01. Mise en oeuvre de l'exercice & les outils

Consigne : A l'aide des différentes API, lorsque vous cliquez sur une carte, vous allez récupérer le nom de la ville afin d'afficher dans la console la température et le temps actuellement dans cette ville

leaflet : <https://leafletjs.com/>

Nominatim(OpenStreetMap):<https://nominatim.org/release-docs/latest/api/Reverse/>

Weather: [https://goweather.xyz/weather/\\$%7BnomDeLaVille%7D](https://goweather.xyz/weather/$%7BnomDeLaVille%7D)



</>

02. Structure du projet

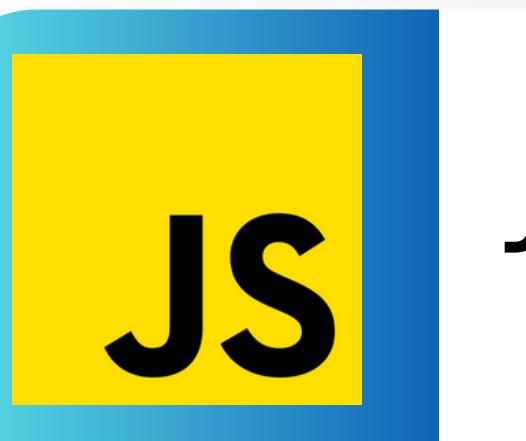
Pour ce projet on a commencé par le html pour préparer le site et nous l'avons relié aux fichiers JavaScript et CSS. Ensuite nous avons suivi les consignes pour implémenter les API dans notre JS et le rendre utilisable selon chacune des spécificités de l'API en question. Enfin après une phase de débug nous avons de nouveau modifié le site pour le rendre plus agréable visuellement et ne pas devoir ouvrir la console pour consulter les résultats météorologique.



HTML



CSS



Javascript



03. HTML & CSS

The image shows a code editor interface with two tabs: 'index.html' and '# style.css'. The 'index.html' tab contains the following HTML code:

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6      <title>Map City Weather</title>
7      <link href="style.css" rel="stylesheet" />
8      <link href="reset.css" rel="stylesheet" />
9      <link
10         rel="stylesheet"
11         href="https://unpkg.com/leaflet@1.9.4/dist/leaflet.css"
12         integrity="sha256-p4NxAoJBhIIN+hmNHzRCf9tD/miZyoHS5obTRR9BMY="
13         crossorigin=""
14     />
15     <script
16         src="https://unpkg.com/leaflet@1.9.4/dist/leaflet.js"
17         integrity="sha256-20nQCchB9co0qIjJZRGuk2/Z9VM+kNiyxNV1lvTlZBo="
18         crossorigin=""
19     ></script>
20 </head>
21 <body class="flex background">
22     <h1>Map City Weather</h1>
23     <div id="map"></div>
24     <div class="info"></div>
25     <script src="js.js"></script>
26 </body>
27 </html>
```

The '# style.css' tab contains the following CSS code:

```
1  :root {
2      --color-blue: #074756;
3      --color-lightBlue: #69bfca;
4      --color-1: #0d7899;
5      --color-2: #4099b2;
6  }
7
8  .flex {
9      display: flex;
10     flex-direction: column;
11     text-align: center;
12     align-items: center;
13     justify-content: center;
14     gap: 1em;
15 }
16
17  body {
18     padding: 1em;
19     font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
20 }
21
22  .background {
23     background-image: url("./image/cloud.webp");
24     background-position: center;
25     background-repeat: no-repeat;
26     background-size: cover;
27     min-width: 100vw;
28     min-height: 100vh;
29 }
30
31  #map {
32     padding: 1em;
33     width: 1500px;
34     height: 700px;
35     border-radius: 1em;
36     border: solid 3px var(--color-1);
37     transition: 0.3s ease;
38 }
39
40  #map:hover {
41     border: solid 3px var(--color-lightBlue);
42 }
43
44  .info {
45     padding: 0.25em 2.5em;
```



04. Javascript

Pour la première api, le site:

<https://leafletjs.com/> nous a guidé pour implémenter une carte OpenStreetMap en JS sur notre page web et nous a fourni également la manière de récupérer les coordonnées au clique de la souris avec .latlng.lat et .latlng.lng pour pouvoir les réutiliser plus tard une fois insérer dans des variables

```
var map = L.map("map", {
  center: [47, 2],
  zoom: 6,
  maxBounds: [
    [-85, -180],
    [85, 180]
  ],
  maxBoundsViscosity: 1.0
});

L.tileLayer("https://tile.openstreetmap.org/{z}/{x}/{y}.png", {
  maxZoom: 19,
  minZoom: 3,
  noWrap: true,
  attribution:
    '&copy; <a href="http://www.openstreetmap.org/copyright">OpenStreetMap</a>'
}).addTo(map);
```

```
let latitude;
let longitude;
let city;
let info = document.querySelector(".info");

map.addEventListener("click", (e) => {
  latitude = e.latlng.lat;
  longitude = e.latlng.lng;

  console.log(latitude);
  console.log(longitude);
```



En second lieu, le site <https://nominatim.org/release-docs/latest/api/Reverse/> nous présente l'api avec des valeurs à remplacer par des coordonnées. Avec un fetch nous allons chercher les informations de localisations précises pour les enregistrer lors du clique. Une fois ces informations misent au format json nous pouvons les mettre dans notre valeur "city" après lui avoir indiqué la valeur recherchée.

```
fetch(`https://nominatim.openstreetmap.org/reverse?format=geocodejson&lat=${latitude}&lon=${longitude}`)
  .then((response) => response.json())
  .then((data) => {
    console.log(data.features[0].properties.geocoding.city);
    city = data.features[0].properties.geocoding.city;
```

Pour l'API 3, le site [https://goweather.xyz/weather/\\${%7Bcity%7D](https://goweather.xyz/weather/${%7Bcity%7D), nous envoie les données météorologique d'une ville, or avec l'API 2 nous avons justement une valeur avec le nom d'une ville qui est désormais utilisable. Il ne nous reste plus qu'à l'utiliser avec un nouveau fetch et afficher dans la console et en HTML les résultats obtenus.

```
fetch(`https://goweather.xyz/weather/${city}`)
  .then(response => response.json())
  .then(data => [
    console.log(data),
    info.textContent = `la ville est ${city} la météo est ${data.temperature}`]);
});
```

</>

05. Visuel final





**Merci pour votre
attention**

**N'oubliez pas de
consulter notre
carte interactive !**