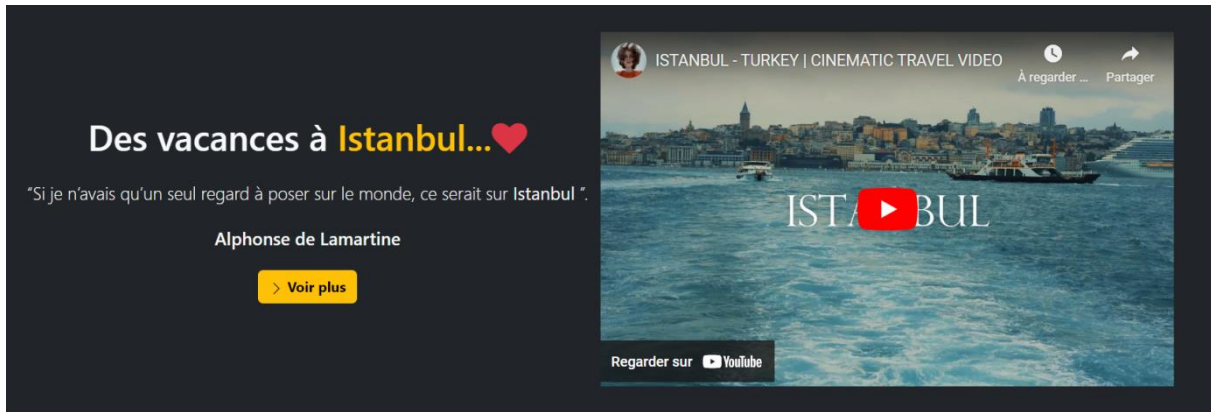


Présentation du projet ECF

Thème choisi : Mon voyage à Istanbul.



Le site parle d'une expérience vécue à Istanbul, et illustre la beauté de cette ville à travers des photos et vidéos et quelques explications et informations qui peuvent être utiles pour les futurs visiteurs de cette ville charmante.

Langages utilisés : HTML, CSS, Bootstrap et JavaScript.

- Manipuler les différents langages et leurs propriétés :
 - Insertion d'images, vidéos, liens..... pour le HTML.
 - Utilisation des accordéons, carrousels, le responsif pour Bootstrap.
 - Applications des fonctions JavaScript pour les API.

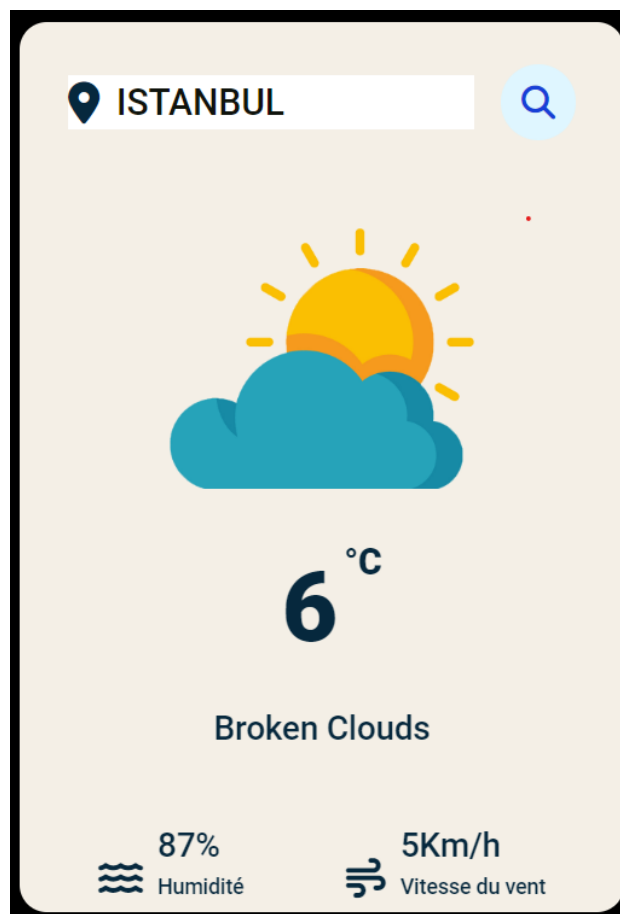
Objectif : Mettre en œuvre ce qu'on a appris durant 2 mois et demi de formation et réaliser un site responsif, valide au niveau du W3C, accessible et répond aux critères imposés par le formateur.

Méthode de travail :

- 1- Rédiger un cahier des charges qui respecte l'énoncé de l'ECF.
- 2- Faire la maquette du site.

- 3- Réaliser les pages du site en HTML et Bootstrap, en prenant comme base une page déjà faite lors d'un exercice Bootstrap.
- 4- S'assurer que le site est valide et accessible.
- 5- Intégrer 2 API au site qui fonctionnent avec le JavaScript, aussi pour mettre en évidence ce qu'on a appris sur les API lors des cours.
- 6- Les API choisies sont dans le thème, une pour la météo et l'autre pour connaître le taux de change bancaire, vu que le site parle d'un voyage à Istanbul, ce genre d'informations est indispensable.

Mode d'emploi de l'api météo :



```
// Déclare les variables et les récupérer en appelant leurs classes.

const container = document.querySelector('.container');
const search = document.querySelector('.search-box button');
const weatherBox = document.querySelector('.weather-box');
const weatherDetails = document.querySelector('.weather-details');
const error404 = document.querySelector('.not-found');

// Créer l'événement en appuyant sur le bouton et faire appel à une ville.

search.addEventListener('click', () => {

    const APIKey = '51bf887c1ee1cc8ac544c981e5386a92';
    const city = document.querySelector('.search-box input').value;
```

```
// Si le nom de la ville est correcte, on demande au Json de nous retourner les infos
// à travers l'url de l'api.
    if (city === '')
        return;

    fetch(`https://api.openweathermap.org/data/2.5/weather?q=${city}&units=metric&appid=${APIKey}`)
        .then(response => response.json())
        .then(json => {

// Si la ville n'existe pas le Json retourne un message d'erreur et il affiche l'image 404
        if (json.cod === '404') {
            container.style.height = '400px';
            weatherBox.style.display = 'none';
            weatherDetails.style.display = 'none';
            error404.style.display = 'block';
            error404.classList.add('fadeIn');
            return;
        }
    }
}
```

```
// Faire disparaître l'image d'erreur "404"
    error404.style.display = 'none';
    error404.classList.remove('fadeIn');

    const image = document.querySelector('.weather-box img');
    const temperature = document.querySelector('.weather-box .temperature');
    const description = document.querySelector('.weather-box .description');
    const humidity = document.querySelector('.weather-details .humidity span');
    const wind = document.querySelector('.weather-details .wind span');
```

```
// appeler les infos météo de l'api en utilisant le "main" de l'api qui nous renvoie les données météo.

/* "parseInt" pour transformer la donnée récupérée de string en chiffre */
description.innerHTML = `${json.weather[0].description}`;
temperature.innerHTML = `${parseInt(json.main.temp)}<span>°C</span>`; /*<span> pour affichage particulier du °C */
humidity.innerHTML = `${json.main.humidity}%`;
wind.innerHTML = `${parseInt(json.wind.speed)}Km/h`;

// Donner le style (d'animation) d'affichage pour la (div class = container) qui contient les info météo
weatherBox.style.display = '';
weatherDetails.style.display = '';
weatherBox.classList.add('fadeIn');
weatherDetails.classList.add('fadeIn');
container.style.height = '590px';

});

})
```

```
// Récupérer les images correspendantes au temps qu'il fait dans la ville demandée.

switch (json.weather[0].main) { /* les data sont stokés sous forme de tableau dans json*/
  case 'Clear':
    image.src = 'img/clear.png';
    break;

  case 'Rain':
    image.src = 'img/rain.png';
    break;

  case 'Snow':
    image.src = 'img/snow.png';
    break;

  case 'Clouds':
    image.src = 'img/cloud.png';
    break;

  case 'Haze':
    image.src = 'img/mist.png';
    break;

  default:
    image.src = '';
}
}
```

Mode d'emploi de l'api change :

Convertisseur de devises

EUR

▼

TRY

▼

Convertir

100 EUR équivalent à 2106.06 TRY

```
// Créer la fonction qui fait appel aux options (la monnaie de chaque pays) à travers l' ID du formulaire.
```

```
document.querySelector('#form').onsubmit = () =>{
```

```
    const base = document.querySelector('#currency-from').value;
```

```
// appeler le lien de l'api pour fournir les réponses nécessaires à la demande du change monétaire.
```

```
fetch(`https://api.exchangerate.host/latest?source=ecb&base=${base}`)
```

```
    .then((response) => response.json()) /* demander une réponse de json*/
```

```
    .then((data) => {
```

```
        const amount = document.querySelector("#amount").value; /* le montant saisi */
```

```
        const currencyTo = document.querySelector("#currency-to").value; /* la monnaie choisie pour le change */
```

```
        const rate = data.rates[currencyTo]; /*récupérer le taux de change de la monnaie du pays sélectionné depuis l'api sous forme de tableau */
```

```
// Fonction qui va faire le calcul du change en récupérant le montant saisi et le multiplier par la monnaie du pays sélectionné.
```

```
    // Elle se met en exécution qu'on appuie sur le bouton
```

```
function convert(){
```

```
    return amount * rate; /* exemple: (12 euros * 20 try) */
```

```
}
```

```
// Afficher le message résultat, en récupérant l'id pour montant saisi + la devise (équivalent à) le montant et la devise du pays qu'on cherche,
```

```
// ici c'est pour Istanbul (TRY). toujours en faisant appel à l'id {currencyTo}
```

```
    document.querySelector(".display-result").innerHTML = `${amount} ${base.toUpperCase()} équivalent à ${convert().toFixed(2)} ${currencyTo}`;
```

```
    })
```

```
    .catch((error) => {
```

```
        console.log("Error: ", error);
```

```
    });
```

```
    return false;
```

```
}
```