

TD n°1 – HTML/CSS/Javascript/GitHub

Contexte : Application web Météo

TRAVAIL A FAIRE

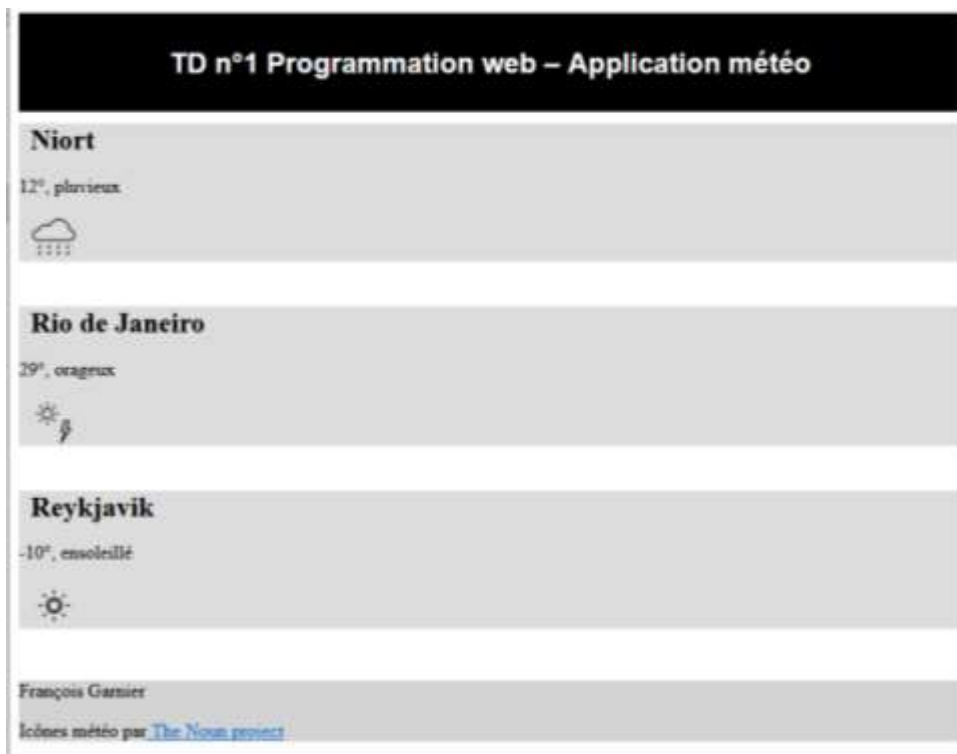
1 Créez un fichier `td1.html` contenant le code suivant:


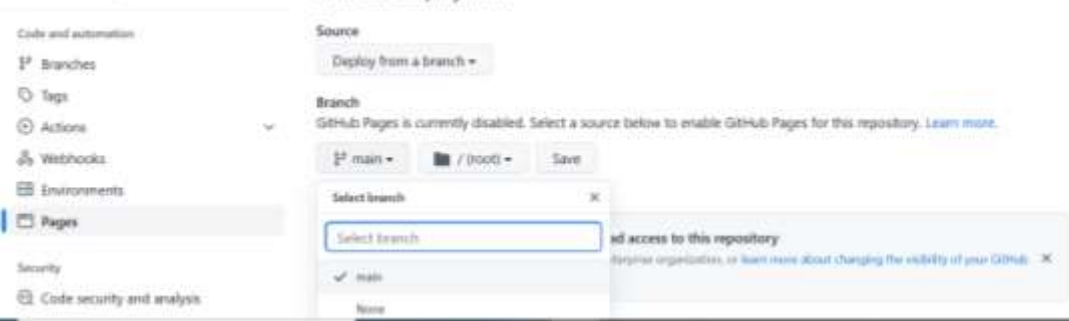
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
  </head>
  <body>
    <h2>Niort</h2>
    <p>12°, pluvieux</p>
    
  </body>
</html>
```

Récupérer l'image d'un nuage pluvieux sur updago et tester cette page HTML.

2 Améliorations (fichier css externalisé pour la mise en forme) :

- insérer un footer en couleur de fond « lightgrey » comportant votre nom et prénom ainsi qu'une référence aux créateurs des icônes météo <https://thenounproject.com>
- insérer un header à fond noir et texte en blanc centré suivant « TD n°1 Programmation web – Application météo »
- visualisation pour 3 villes avec les données et images de votre choix en fond « gainsboro » dans des `<div>` permettant la représentation suivante :

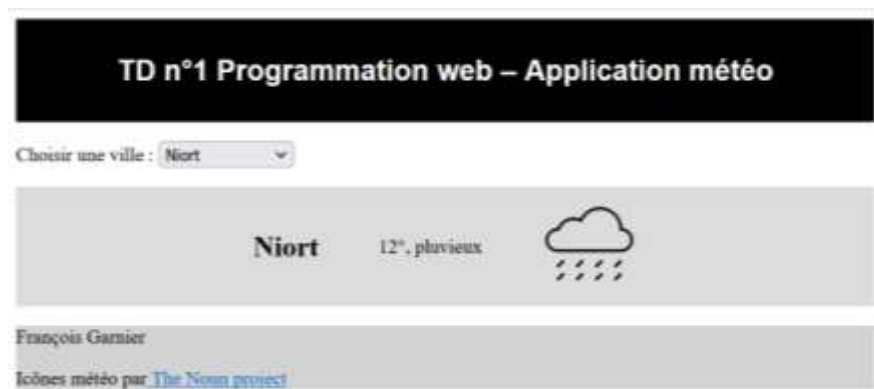


| | |
|---|--|
| 3 | <p>Maintenant nous allons rendre la page plus flexible, en nous appuyant sur les flexbox CSS :</p> <ul style="list-style-type: none"> • Tutoriel vidéo 35 minutes : https://grafikart.fr/tutoriels/flexbox-806 • Guide flexbox de Mozilla : https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox • Tutoriel en anglais : https://css-tricks.com/snippets/css/a-guide-to-flexbox/ <p>L'objectif est de mettre les trois villes côte à côte avec une visualisation proche de celle-ci :</p>  <p>Avec la flexbox, si la taille de la fenêtre diminue, ou que vous ouvrez la page sur un smartphone (testez avec l'inspecteur de code : F12 de votre navigateur), vous devriez avoir un rendu qui réduit automatiquement les villes pour qu'elles tiennent sur la même ligne.</p> |
| 4 | <p>Nous allons mettre en ligne sur GitHub notre application. Vous pourrez donc la tester !</p> <p>GitHub est un service en ligne permettant d'abord de versionner son code (versioning) et de collaborer à plusieurs développeurs sur un dépôt nommé « repository ». Cet outil permet donc un travail collaboratif sur un projet.</p> <p>GitHub offre également la possibilité d'héberger un site Web associée à un projet.</p> <ul style="list-style-type: none"> • Créez un compte Github. Attention votre nom d'utilisateur apparaîtra dans l'URL de votre site sous la forme <code>https://nom.github.io/</code> • Créez un nouveau projet « create repository » et nommer le impérativement nomUtilisateur.github.io C'est ce qui permettra à GitHub de comprendre qu'il s'agit d'un projet web. • Créer un fichier <code>index.html</code> dans l'onglet code « create new file » en lien bleu et copier/coller le contenu du fichier <code>td1.html</code> ; Enregistrer le fichier en <code>td1.html</code> • Se rendre dans les Settings (préférences) puis menu Pages à gauche et activer le mode Github pages associé à la main branch :  <ul style="list-style-type: none"> • Retourner dans l'onglet « code », puis bouton « Add Files » puis « Upload Files ». Téléverser tous les fichiers du td1 (png et css). • Votre projet web est en ligne et accessible du monde entier ! Tester votre site web météo hébergé sur GitHub en tapant dans votre navigateur : https://nomUtilisateur.github.io |

5

L'application web « météo » évolue car nous souhaitons faire en sorte de pouvoir sélectionner une des trois villes dans une liste déroulante et afficher la météo uniquement de cette ville (les autres seraient cachées). Pour réaliser cela, suivre les étapes suivantes dans vos fichiers en local sur votre PC :

1. Créer une balise `<script>` dans le `<head>`.
2. Créer une fonction : `function afficheVille() { alert(villeChoisie); }`
3. Créer une variable : `let villeChoisie = "Niort"`
4. Connecter la fonction à l'événement de chargement de la page. Pour cela modifier la balise body de la façon suivante `<body onload="afficheVille()">`
5. Tester !
6. Ajouter un menu déroulant avec la balise html `select` contenant les 3 villes juste en dessous du header. Placer un attribut `id` à ce `select`.
7. Supprimer l'appel à `onload` dans le body et rajouter l'appel à `afficheVille()` à la balise `select`: `<select onchange=afficheVille()>`
8. Placer un attribut `id=nomVille` à chaque bloc correspondant à une ville (div) et un appel de classe css nommée `ville`, même si la classe n'existe pas !
9. Nous allons procéder à la modification de la fonction `afficheVille` pour que seule la ville sélectionnée dans le menu s'affiche :



Pour cela, référez-vous à la partie 11 sur le DOM du cours n°1 javascript et compléter la fonction `afficheVille()` :

```
function afficheVille()
{
    // 1. Récupération de la valeur courante du select avec son id
    let nomVilleChoisie = ...

    // 2. Récupération de la liste de toutes les div correspondant à une div de météo
    // grâce à la classe « ville »
    let villes = ...

    // 3. Parcours de la liste des div météo. On cache celles non sélectionnées.
    for (i=0; i<...; i++)
    {
        // l'attribut id de l'élément est comparée à la ville choisie
        if (villes[i].id == nomVilleChoisie)
        {
            // garder l'élément visible -> ELEMENT.style.display= "";
            ...
        } else
        {
            // cacher l'élément -> ELEMENT.style.display= "none";
            ...
        }
    }
}
```

| | |
|---|--|
| 6 | Externaliser tout le javascript dans un fichier <code>td1.js</code> (comme pour un fichier css), la partie javascript de la question précédente du fichier <code>td1.html</code> <i>Chercher sur le net...</i> |
| 7 | Publier sur GitHub votre site web et tester en ligne l'opérationnalité de votre application web. |
| 8 | <p>BONUS : Améliorer l'application pour la rendre dynamique via une API : vous serez en mesure d'utiliser une API pour récupérer les données réelles de météo. Par exemple en utilisant celle-ci : https://openweathermap.org/current Il faudra :</p> <ol style="list-style-type: none"> 1. Créer un compte 2. Obtenir une clé API 3. Faire un nouveau programme html basé sur le précédent et qui fera une lecture sur : https://api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}&appid={API key} <p>mon API key : <code>eccf8200898239c01b81626293da9f1d</code></p> <p>Exemple pour Niort https://api.openweathermap.org/data/2.5/weather?lat=46.3167&lon=-0.4667&appid=eccf8200898239c01b81626293da9f1d</p> <p><u>Conseils :</u></p> <ul style="list-style-type: none"> . Vous pouvez utiliser postman pour tester l'API et visualiser le json résultat (voir support de cours « saé collecte web ») . Suivre la vidéo suivante pour comprendre comment récupérer l'API et la traiter en javascript : https://www.youtube.com/watch?v=C8bKthavr6E . Mutualiser l'API pour les coordonnées des 3 villes, rajouter les deux premières lettres du pays (ou d'autres informations que vous trouverez dans le fichier json récupéré) et publier sur GitHub ! |