

CHARLES PREVPST

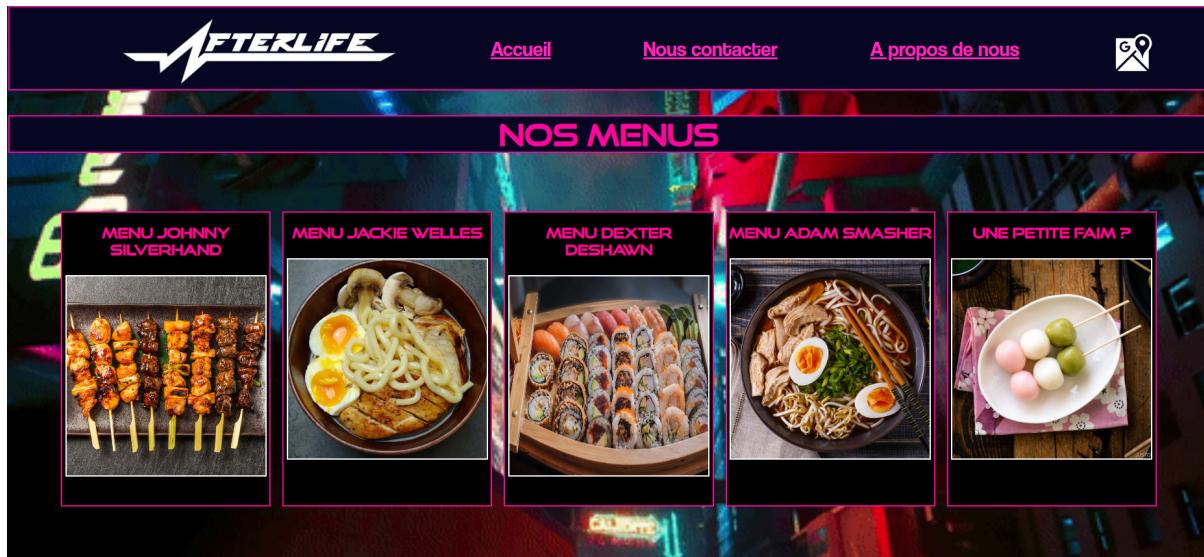
TP6

SAE 105 : Créer un site web.

Nota bene : J'ai travaillé sur différents postes et donc la résolution de la page peut changer selon les écrans, le zoom étant à 90% sur le pc portable utilisé lors de ce saé. Il faudra donc sûrement dézoomer ou zoomer au cran au dessus/au dessous.

Mon projet est de créer un site web de Fast Food qui sert de la StreetFood Japonaise, celui-ci s'appelle L'Afterlife, il est inspiré d'un jeu vidéo intitulé Cyberpunk 2077.

Page d'accueil et de menu du site



Pour les deux pages ci-dessus et les autres du site, afin de mettre les cartes alignées ainsi que la nav bar, j'ai utilisé le flexbox, celui-ci m'a permis de pouvoir les arranger comme bon me semblait tout en gardant un style "épuré".

Connaissant pas tellement le FlexBox, je me suis renseigné sur l'école du web sur youtube (lien en fin pdf) et de camarades connaissant ce système afin de comprendre et de pouvoir l'utiliser.

C'est une des seules commandes que j'ai eu besoin d'utiliser qu'on avait pas vu en cours.

Algorithmes de contenus dynamiques (Javascript)

1. J'ai créé un tableau avec tous les jours de la semaine pour donner jours et les heures sous forme d'un tableau.
2. On récupère l'élément "**table**" avec "**GetElementById**" et on crée des colonnes et lignes afin de placer les informations dans celles-ci.
3. On ajoute le tableau dans le body afin que celui-ci soit sur la page.
4. Puis on utilise "**const horaires**" pour écrire les horaires selon les jours et les emplacements.

En somme, ça nous donne ceci :

```
<script>
  // Création d'un tableau contenant les jours de la semaine
  const jours = ['Lundi', 'Mardi', 'Mercredi', 'Jeudi', 'Vendredi', 'Samedi', 'Dimanche'];

  // Récupération de l'élément table dans le HTML
  const tableauJours = document.getElementById('joursSemaine');

  // Boucle pour ajouter chaque jour au tableau
  jours.forEach((jour, index) => {
    // Crée une nouvelle ligne et des cellules pour le jour et son numéro
    const newRow = document.createElement('tr');
    const cell1 = document.createElement('td');
    const cell2 = document.createElement('td');

    // on assigne le contenu aux cellules
    cell1.textContent = jour;
    cell2.textContent = index + 1;

    // on ajoute les cellules à la ligne
    newRow.appendChild(cell1);
    newRow.appendChild(cell2);

    // on ajoute la ligne au tableau
    tableauJours.appendChild(newRow);
  });

  // on ajoute le tableau au body du document
  document.body.appendChild(tableauJours);

  const horaires = [
    '8:00 - 17:00 à Paris', // Lundi
    '8:30 - 18:00 à Paris', // Mardi
    '9:00 - 17:30 à Louviers', // Mercredi
    '8:00 - 16:00 à Louviers', // Jeudi
    '9:30 - 18:30 à Elbeuf', // Vendredi
    '10:00 - 14:00 à Elbeuf', // Samedi
    'Fermé' // Dimanche
  ];

  // Boucle pour remplir la colonne des horaires
  const cellulesHoraire = document.querySelectorAll('.tableau td:nth-child(2)'); // Sélectionne toutes les 2e cellules du tableau

  cellulesHoraire.forEach((cellule, index) => {
    cellule.textContent = horaires[index];
  });
</script>
```

[Accueil](#)[Nos menus](#)[Nous contacter](#)[A propos de nous](#)

Jour	Horaires
Lundi	8:00 - 17:00 à Paris
Mardi	8:30 - 18:00 à Paris
Mercredi	9:00 - 17:30 à Louviers
Jeudi	8:00 - 16:00 à Louviers
Vendredi	9:30 - 18:30 à Elbeuf
Samedi	10:00 - 14:00 à Elbeuf
Dimanche	Fermé

Retrouvez-nous sur : [En collaboration avec CDProject RED](#)

J'ai utilisé ma fiche de style css "**style5.css**" pour la police des cellules et la bordure du tableau en entier.

On obtient donc un résultat concordant avec la page et qui est beau à voir visuellement et sans bugs.

```
.tableau{  
    position: absolute;  
    color: white;  
    border-collapse: collapse;  
    text-align: center;  
    top: 280px;  
    width: 40%;  
}  
  
.tableau th {  
    border: 3px solid #ff1111;  
    padding: 8px;  
    font-size: 30px;  
    font-family: Coolvetica;  
    background-color: #rgba(0, 0, 0, 0.712);  
}  
  
.tableau td {  
    border: 3px solid #ff1111;  
    padding: 8px;  
    font-size: 30px;  
    font-family: Coolvetica;  
    background-color: #rgba(0, 0, 0, 0.712);  
}  
  
.louviers{  
    position: absolute;  
    left: 1000px;  
    top: 280px;  
}
```

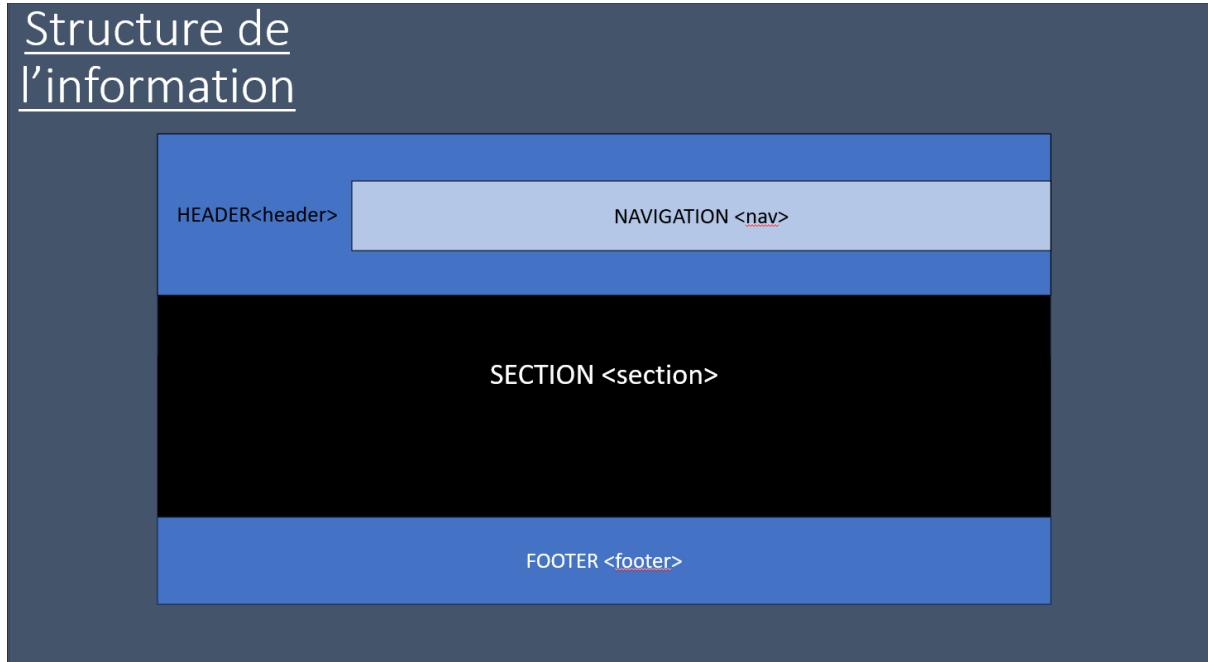
Script du menu

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>NOS MENUS</title>
    <link rel="stylesheet" href="style2.css">
    <script>
        function toggleText(cardId) {
            var card = document.getElementById(cardId);
            var paragraph = card.getElementsByTagName('p')[0];

            if (paragraph.style.display === 'none') {
                paragraph.style.display = 'block';
                card.getElementsByTagName('img')[0].style.display = 'none';
            } else {
                paragraph.style.display = 'none';
                card.getElementsByTagName('img')[0].style.display = 'block';
            }
        }
    </script>
</head>
<body>
```

.Structure de l'information du site web

Structure de l'information

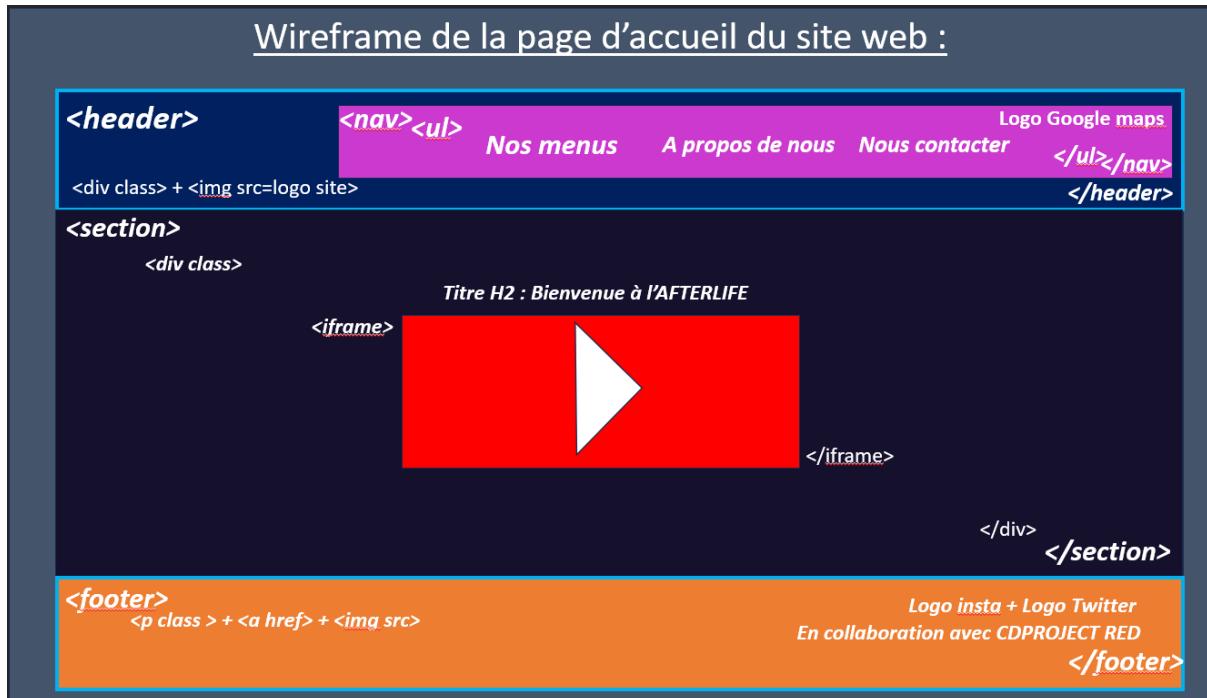


Etape 2 : Lister les solutions techniques d'intégration web envisageables pour mettre en valeur cette architecture.

On pourrait intégrer un “FlexBox” dans le navbar afin d’aligner tout, ou encore de mettre les éléments contenus dedans en “absolute” et les répartir avec des “margin”

On peut mettre la position du header et du footer en fixed afin qu'il reste à sa bonne position.

Etape 3 : Construire les wireframe des pages sous forme de bloc en envisageant les types de conteneur de ces blocs (balise, id, ...).



Etape 6 : Optimisation du code en vue de la validation W3C.

Après vérification, des erreurs avaient été découvertes, j'ai donc corrigé celles-ci afin d'obtenir la validation CSS et HTML.

Code HTML

Vérificateur Nu HTML

Cet outil est une expérience en cours visant à améliorer la vérification HTML et son comportement reste susceptible de changer.

Affichage des résultats pour le fichier téléchargé index.html

Entrée du vérificateur

Montrer source contour Options de rapport d'image

Vérifier par Aucun fichier choisi

Les fichiers téléchargés avec les extensions .xhtml ou .xht sont analysés à l'aide de l'analyseur XML.

Vérification des documents terminée. Aucune erreur ou avertissement à afficher.

J'ai utilisé l'analyseur HTML.
Temps d'exécution total 9 millisecondes.

[À propos de ce vérificateur](#) • [Signaler un problème](#) • Version : 23.11.28

Code CSS

Résultats de la validation W3C CSS de style1.css (CSS niveau 3 + SVG)

Félicitations ! Aucune erreur trouvée.

Ce document est valide conformément à la recommandation [CSS niveau 3 + SVG](#) !

Pour montrer à vos lecteurs votre souci d'interopérabilité lors de la création de cette page Web, vous pouvez afficher cette icône sur toutes les pages valides. Voici le fragment de XHTML que vous pouvez utiliser pour ajouter cette icône à votre page Web:

```
<p>
  <a href="http://jigsaw.w3.org/css-validator/check/referer">
    
  </a>
</p>
```

```
<p>
  <a href="http://jigsaw.w3.org/css-validator/check/referer">
    
  </a>
</p>
```

(fermez le tag img avec > au lieu de /> si vous utilisez HTML <= 4.01)

Sources et Inspirations :

Sources :

1. Openclassroom pour le javascript et le flexbox.
2. W3School pour les polices personnalisées.
3. L'école du web (https://www.youtube.com/watch?v=S3WECGYv_rQ) pour le flexbox.