**Soutenance**

*Lexique :*

**Balises sémantiques :** \_ aide au référencement du code HTML

\_ permet optimisation du SEO

\_ Ex : section, article, header, footer, nav, form, aside, h2

**SEO :** pour améliorer la position d’un site web sur la page de résultats du moteur de recherches

**La balise canonical :** indique aux moteurs de recherche que ma page web est la vraie, l’unique… (Ex : link rel="canonical"…)

**Media Queries :** permettent de modifier l'apparence d'un site ou d'une application en fonction du type d'appareil

**Meta charset="utf-8" :** pour que le navigateur comprenne les caractères spéciaux (ex : accents)

**Meta viewport :** C’est une fenêtre virtuelle qui améliore l'aspect des sites non optimisés pour les mobiles sur les appareils à écran étroit.

**Grid :** créer des mises en page en remplacement du Flex par un système de grille

**Flex :** c’est une propriété qui modifier les dimensions et emplacement d’un élément à l’intérieur de son conteneur.

*Introduction :*

J’ai commencé ce projet par suivre les cours mis à disposition sur OpenClassrooms ; tel que GitHub, le HTML et CSS, savoir utiliser une ligne de commande dans un terminal, mettre en place un environnement front-end et intégrer et découper une maquette.

Durant l’intégration de ma maquette, j’ai commis plusieurs erreurs sur lesquels je reviendrais plus en détails tout au long de la soutenance. J’ai ainsi pu résoudre les différents problèmes rencontrés avec l’aide mon mentor.

Ce projet a abouti à trois pages : index.html, StyleReservia.css et Responsive.css.

*La page en HTML :*

Tout d’abord, j’ai passé plusieurs heures à appréhender et mettre en place GitHub et mon terminal GitHub Bash.

Ensuite, j’ai codé la page en HTML en suivant une méthodologie : je me suis concentrée sur les éléments généraux et après les éléments les plus spécifiques.

Dans les éléments généraux, dans l’ordre chronologique :

\_ La structure de base, tel que le DOCTYPE, les balises méta et les feuilles de styles associées

\_ Le header

\_ Le body

\_ Le footer

**Ce que j’ai fait :**

J’ai codé les titres en les classifiant, les images, le texte… J’ai d’abord mis les images dans des balises « figure » et avec des « figcaption » mais cela complexifiait trop le code. Je les ai donc enlevés. La balise img est optimisée, notamment avec son attribut alt.

Après mettre concentrée sur la partie globale, j’ai fais des class et des id. J’avais beaucoup d’attribut id, ce qui pouvait perturber l’interprétation et l’affichage de la page web. Alors, j’ai choisi de garder seulement les class. J’avançais section après section, mais lorsque je bloquais sur une partie, je la mettais de côté et en attaquait une autre. Puis je revenais sur le ou les blocages après quelque temps. L’intégration en HTML a été assez simple pour moi, hormis les class, j’ai eu peu de difficultés.

L’utilisation de class m’a amené à faire l’intégration en CSS.

**Les difficultés rencontrées :**

Au début, j’ai commis l’erreur de mettre le header dans le body.

Je ne savais comment mettre plusieurs class dans une même balise. Par exemple, à la ligne 99 de la page index.html, l’attribut class comprend plusieurs valeurs fas, fa-star et blue. J’ai compris que l’espace entre chaque mot signifie une class différentes.

Un autre blocage rencontré a été au niveau des balises meta ; il fallait que je code méthodiquement les balises générales et celles relatives à Twitter ou Facebook. Le site internet de l’exercice étant un site d’information et de réservation en ligne pour trouver des hébergements et des activités dans la ville que l’utilisateur a choisi, les balises twitter et Facebook m’ont semblé indispensable.

*Le page en CSS :*

**Ce que j’ai fait :**

L’intégration en CSS a été moins simples. J’étais un peu perdu concernant la quantité de styles à intégrer. J’ai donc fais régulièrement des points avec mon mentor pour évaluer les priorités, et l’ordre des éléments à intégrer. J’ai donc suivi le même schéma que pour la page HTML ; en traitant mon code section par section. J’ai stylisé chaque class.

L’utilisation du Flexbox, m’a demandé d’approfondir mes connaissances via des sites tel que « Mozilla Developer » ou encore un autre très ludique « Flexbox Froggy ». Il fallait que je comprenne l’utilisation des propriétés et des valeurs de manières plus concrètes.

**Les problèmes rencontrés :**

Une erreur fréquente que je faisais était de nommer les class en français, ou avec des chiffres ou des lettres (1, 2 ou A, B), ou encore avec des noms dont la signification était trop éloignée de l’élément à styliser. En m’aidant de Google Traduction ; j’ai revu l’appellation de mes class pour permettre une meilleur relecture et correction à apporter à mon code.

Au niveau du Flexbox, l’intégration des « div accommodation » sans s, n’a pas été simple. Je n’arrivais pas à faire tenir mes blocs ou fenêtres à l’intérieur de mon autre fenêtre « accommodations » avec une s. Les 6 petits rectangles blancs à l’intérieur du grand rectangles gris, par exemple, dans la section Hébergements à Marseille.

Je me suis aussi retrouvé bloquée avec les propriétés height, width, margin et padding. Dans ce cas-là, je sais maintenant qu’il faut utiliser Inspecteur de Google. Il y a la case flex, ou encore une fois que j’ai sélectionné un élément, les marges intérieures apparaissent en violettes et les marges extérieures apparaissent en jaune.

La page en Responsive :

Lorsque je suis arrivé à la partie Responsive, un énorme problème s’est posé pour savoir comment intégrer cette feuille de style que je ne connaissais pas avant de faire ce projet. Mon blocage était de savoir comment coder à partir de 2 pages qui affichait un style et une organisation différente de celle qui devait s’afficher en responsive. Plusieurs points ont du être vu avec mon mentor.

J’ai intégré plusieurs tailles de Media Queries.

J’avais trop de marges négatives, alors je suis passée par le flex, ce qui m’a permis de quasiment toutes les supprimer.

J’ai intégré 7 Media Queries (avec une max-width à 1700, 1400, 1200, 1050, 991, 576 et 400 pixels. Je les ai codé dans l’ordre décroissant.

Pour rendre mon code plus visible, j’ai indenté les propriétés de chaque Media Queries. Ça me per mettait de vérifier qu’il y avait bien accolade de fin.

*Le Validateur de Firefox :*

Pour finir, j’ai vérifié avec le navigateur Firefox, la validité de mon code sur W3C Validator. Il m’a relevé quelques erreurs tel que l’oublie de guillemets, le niveau des titres (pour les titres, ceux que j’avais mis en h1 sont passés en h2. Et les h2 sont passés en h3. Le message d’erreur portait sur le fait que le h1 « Hébergements à Marseille », « Les plus populaires » et « Activités à Marseille » n’était pas dans le header et ne pouvait pas porter la catégorie h1. Du coup, pour la bonne cohérence je les ai tous descendu d’un échelon.

La dernière erreur rencontrée se situait au niveau des balises sémantiques section, que j’ai remplacé par des div.

*En résumé*

Ce projet, m’a permise d’acquérir de bonnes bases pour pouvoir postuler en entreprise. En continuant à travailler mes cours et connaissances sur des projets personnels, j’aurais de bonnes compétences en HTML, CSS et Responsive. Je me sens capable de travailler seul sur un nouveau projet personnel basique, qui contient peu de difficultés.