

Produire un site web

SAE 105



Mélissa Simon

BUT Métiers du Multimédia et de l'Internet

Année 2022 – 2023

I.Introduction	2
II.La maquette visuelle.....	3
II.1.Lien de la maquette	3
II.2.Design de la maquette	3
III.Le site web	6
III.1.CSS	6
III.1.1Body en css.....	6
III.1.2Header en css	6
III.1.3Les différentes pages en css	7
III.1.4Scrollbar en css.....	9
III.1.5Formulaire en css	10
III.2.HTML.....	11
III.2.1Header en html.....	11
III.2.2Les différentes pages en html	12
III.3.Hébergement du site web	13
III.3.1Lien de mon site web	13
III.3.2Github.....	13
IV.Conclusion	14
V.Ressource.....	14

I. Introduction

Pour la SAE 105, on doit développer un site statique avec HTML et CSS dans un design minimaliste qui doit présenter notre profil, c'est-à-dire notre portefeuille. S. Il devra être conforme au W3C et utiliser des balises de l'HTML5. Et il doit être déployé en ligne. En tant qu'étudiant, on doit mener un travail de conception, d'intégration et de développement des vues d'un site Web en respectant les critères donnés par l'enseignant.

II. La maquette visuelle

II.1. Lien de la maquette

<https://www.figma.com/file/ufHL0kXHueLrwCmucxx9tJ/PorteFolio?node-id=0%3A1&t=qdZknCpuo8AZ6jpG-1>

II.2. Design de la maquette

Pour la conception de la maquette visuelle, j'ai décidé d'utiliser le logiciel Figma. En premier temps, j'ai établi ma typographie en choisissant la police et les couleurs de ma maquette



Figure 1: colorimétrie

Un dégradé de couleur beige tirant vers le marron. J'ai fait ce choix pour avoir un effet neutre et simple sous l'œil de l'utilisateur.

Norican

Norican

Figure 2: Police d'écriture

Ensuite j'ai effectué des recherches de d'exemples de police et j'ai fini par choisir la police Norican qui donne une certaine douceur à la maquette, une légèreté derrière les couleurs que j'ai choisies.

Pour commencer, j'ai d'abord créé l'accueil de mon site web, j'ai créé d'abord le header en blanc en mettant 4 sections, qui suis-je, les compétences, mes projets contact écrivent avec la police Norican. Pour délimiter le header et le reste de la page, j'ai ajouté une barre de la même couleur de la page c'est à dire le beige le plus clair, cela donne aussi un côté soigné. J'ai Ensuite ajouter en haut à gauche mon logo qui est composé d'un cercle noir avec à l'intérieur mes initiales (MS) écrivent avec la police Mogra. Pour finir la page d'accueil, j'ai mis au centre de la page mon prénom est mon nom toujours avec la police Norican, pour montrer à l'utilisateur à qui et de quel sujet va parler ce site web : de moi, de mes compétences et de mes projets.

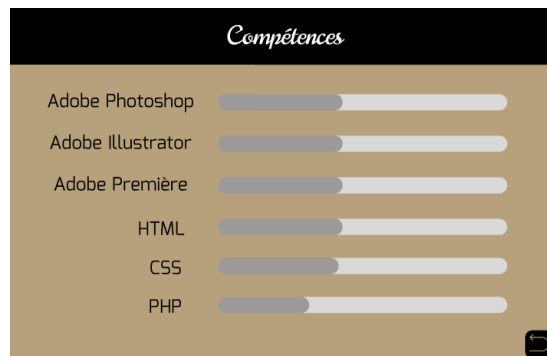


Figure 3: Page d'accueil de la maquette

Après je suis passée à la création de la page « Qui-suis-je ? », pour présenter la page j'ai mis une bannière de couleur noire où j'ai écrit le nom qui suis-je de couleur blanc avec la police Norican, j'ai mis la page entière d'une couleur beige un peu plus foncée que la page d'accueil, au centre j'ai mis du texte qui me décrit de couleur noire avec toujours la même police non c'est réel. En bas à droite de la page nous pouvons trouver un bouton qui permet lorsqu'on clique dessus de retourner à la page d'accueil.



Ensuite, et j'ai créé une nouvelle page en dessous des 2 autres pages qui hé correspond à la page compétence. hé j'ai repris le même format que la page qui suis-je en changeant le titre hé en changeant l'art couleur d'arrière-plan en un brin beaucoup plus foncé. J'ai remplacé le texte de qui suis-je par plusieurs barres blanches qui montre la jauge en gris de mes compétences pour Adobe Photoshop, Adobe illustrator, Adobe première, HTML, CCS ET PHP.



Par la suite, j'ai refait le même schéma en créant une nouvelle page pour la section mes projets, j'ai échangé l'arrière-plan hé avec mon couleur qui c'est quoi tire le plus vers le marron. J'ai mis des cases noires où j'ai inséré des images de plusieurs de mes projets et des textes juste en bas qui correspondent à un élément de mes projets : « Logo » et « Photos ».

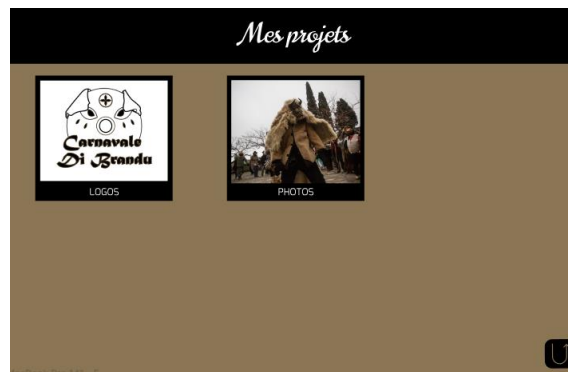


Figure 6: Page Mes Projets de la maquette

Pour finir, toujours avec le même style de page, on utilisant une couleur d'arrière-plan plus foncé que la précédente, j'ai créé la page contact en créant un formulaire gris pastel et gris foncé, on peut remplir son nom son prénom son email et détaillé ce que l'on veut me dire avec un bouton noir « envoyer ».

Figure 7: Page Contact de la maquette*

Toutes ces pages sont les uns en dessous des autres, on peut scroller pour les atteindre. Mais pour atteindre une particulièrement sans scroller on peut cliquer sur les différentes sections qui nous amène à la page qu'on veut, pour retourner à la page d'accueil on clique sur les boutons noirs en bas à droite sur les différentes pages. J'ai positionné chaque boutons ou lien cliquable en fonction de l'accessibilité de l'utilisateur, cela lui permet une facilité d'accès.

Pour terminer, j'ai créé sur la maquette une page qui apparaît lorsqu'on clique sur le mot logo dans la page mes projets, cette page sert à démontrée les différents projets que j'ai faits en créant des logos. Pour cette page de la maquette, j'ai réutilisé les couleurs de la page d'accueil en inversant, le header est beige clair et la page est blanche. On retrouve l'image du logo à gauche et le texte à droite est toujours avec son bouton retour. Je n'ai pas utilisé cette page de la maquette sur mon site web.



Figure 8: Page Logo de la maquette

III. Le site web

Pour coder mon site web, j'ai utilisé le site codesandbox.

Lien du codesandbox :

<https://codesandbox.io/s/060123j-nmxlvj?file=/styles.css>

III.1. CSS

III.1.1 BODY EN CSS

Pour le code CSS, j'ai écrit une page styles.css où je l'applique. J'ai d'abord créé un body, le body est une balise qui fonctionne sur l'entièreté de ton site web, j'ai ajouté dans cette balise le style background-color qui donc met en couleur tout l'arrière-plan du site web avec la couleur #b7a07c qui correspond à un beige. J'ai ajouté dans cette balise le style font-family Norican que j'ai importé grâce @import l'url du style via google fonts, mettre ce style dans le body permet aux textes dans tout le site web de prendre la police Norican.

```
body {  
  margin: 0;  
  background-color: #b7a07c;  
  font-family: Norican;  
}
```

Figure 9: Codage du body

III.1.2 HEADER EN CSS

Pour le haut de page, j'ai utilisé la balise header avec un style height de 14vh, qui correspond à la hauteur de ton header et un style width de 100% qui prend donc toute la largeur de la page. J'ai ajouté un background-color, donc un arrière-plan de couleur blanc. On met le header avec un style position mis sur fixed qui permet au header de ne pas bouger malgré le scroll de la page. J'ai ajouté une balise nav qui à une section d'une page d'avoir un lien vers d'autre page. J'ai aussi ajouté une insidebar noir, c'est-à-dire une barre en dessous du header pour bien démontré la séparation entre le header des pages qu'il survole.

```
header {
  height: 14vh;
  width: 100%;
  background-color: white;
  position: fixed;
}
```

Figure 10: Codage Header

```
.insidebar {
  background-color: black;
  height: 2vh;
  width: 100%;
  overflow: hidden;
  position: absolute;
  bottom: 0;
}
```

Figure 11 : Codage du header

III.1.3 LES DIFFERENTES PAGES EN CSS

Pour différencier les différentes pages, j'ai créé des balises dans les noms différents des différentes pages. Je leur ai tous mis 86vh de hauteur avec le style height pour qu'ils aient tous la hauteur de la page d'accueil avec le header, 86vh car le header fait 14vh de hauteur. Chacun à son background-color, c'est-à-dire d'arrière-plan de couleur différentes pour que l'assemblage des pages fassent un dégradé de beige.

```
.projets {
  height: 86vh;
  background-color: #8a7655;
}
```

Figure 12: Codage de la page projets

Chaque page à ses propriétés différentes, la page qui suis-je contient la balise P qui correspond au style d'un paragraphe, on retrouve dans cette balise le style text-align, qui correspond à la direction du texte, je l'ai mis en center pour le centré, aussi on trouve un font-size qui permet d'indiquer la taille du texte et un font-family (le format de la police) que j'ai changé pour Condensed car j'estimais qu'il permettait une lecture plus facile pour l'utilisateur que la police du body.

```
.quiusisje {
  height: 86vh;
  background-color: #ccb693;
}

.quiusisje p {
  text-align: center;
  font-size: 1.7rem;
  font-family: Fira Sans Condensed;
}
```

Figure13: Codage Page Quiusisje

Pour la page des compétences, j'ai mis une balise Progress, cette balise permet la création d'une jauge, en indiquant une valeur, la jauge monte ou descend. Un webkit-progress-value est mis qui correspond à la jauge qui monte ou décroît, c'est à lui qu'on donne une valeur pour avoir sa largeur, le webkit-progress-bar est l'arrière fond de la jauge.


```

progress {
  appearance: auto;
  box-sizing: border-box;
  display: inline-block;
  block-size: 1.1em;
  inline-size: 15em;
  vertical-align: 3.5em;
  writing-mode: horizontal-tb !important;
  background-color: #b7a07c;
}

```

Figure 14: Codage de la balise Progress

Une balise h4 est ajoutée qui correspond aux différents noms des jauges qui correspondent aux différentes compétences représentées.

```

h4 {
  font-family: auto;
  font-size: auto;
  display: flex;
  margin-block-start: 1.33em;
  margin-block-end: 1.33em;
  margin-inline-start: 0px;
  margin-inline-end: 0px;
  font-size: auto;
  font-weight: bold;
}

```

Figure 15: Codage de la balise h4

Avec la page projets, j'ai mis en place une balise main qui regroupe les différentes propriétés de la page mes projets. J'ai ajouté une balise section pour regrouper les priorités communes sur les 3 carrés que je veux ajouter sur la page pour démontrer 3 types de projets. Pour les éléments qui ne sont pas en commun, j'ajoute la balise section:nth-of-type() en leur donnant chacun un numéro. J'ai développé une animation pour les boutons Photos, Design et Exercices en utilisant une balise section pour le design des boutons et des balises acol et acol:hover qui permettent le changement de couleur lorsqu'on passe la souris sur les boutons.

```

section {
  height: 35vh;
  width: 30%;
  /* max-width: 370px; */
  background-color: #eeeeee;
  padding: 24px 40px;
  text-align: center;
  margin: 5px;
}

```

Figure 16: Codage balise section

```

section:nth-of-type(1) {
  border-top: solid 10px black;
}

section:nth-of-type(2) {
  margin-left: 30px;
  margin-right: 30px;
  border-top: solid 10px black;
}

section:nth-of-type(3) {
  border-top: solid 10px black;
}

```

Figure 17: Codage balise section:nth-of-type()

Par la suite, j'ai créé la page contact avec un footer qui représente le pied de page de la page web , qui dans mon cas représente le footer du site web. Dans le footer, j'ai mis plusieurs style comme `display :flex` qui rend flexible le footer pour qui s'adapte à tout taille qu'on donne au site web via le navigateur. Aussi, avec le style `background-color`, on permet au footer de se colorier entièrement en noir, avec le style `color` on permet au texte dans le footer de se colorier en blanc. On lui donne avec le style `height` une hauteur de `8vh`, avec le style `width` une largeur de `100%` de la page. Pour cette partie du site web, le `font-family` c'est-à-dire la police est en mode auto, la police basique est utilisée pour une meilleure visibilité pour l'utilisateur.

```
footer {
  display: flex;
  justify-content: center;
  margin: 0;
  padding: 0%;
  background-color: black;
  color: #fff;
  align-items: center;
  height: 8vh;
  width: 100%;
  font-family: auto;
}
```

Figure 18: Codage du footer

Comme balise commune qu'on peut retrouver dans le code, c'est le codage de la bannière, tous ces styles fonctionnent sur chaque bannière sur chaque page sauf l'accueil. On peut s'apercevoir que le `width` (la largeur) fait `100%` de la largeur du site, le `height` (a hauteur) fait `14vh` du site. C'est en fonction de la bannière que la taille de la page est choisie, sur `100vh`, la taille d'une page entière, on enlève les `14vh` de la bannière pour garder toujours la taille d'une page entière web. Avec le style `background-color`, la couleur de la bannière est noire et avec le style `color` cela permet d'avoir le texte en blanc. On rend responsive et flexible la bannière avec le style `display` en `flex`, on met au centre la bannière et le texte grâce aux balises `justify-content` et `align-items`.

```
.banniere {
  width: 100%;
  height: 14vh;
  background-color: black;
  color: white;
  display: flex;
  justify-content: center;
  align-items: center;
  font-size: 2rem;
}
```

Figure 19: Codage de la bannière

III.1.4 SCROLLBAR EN CSS

Pour la scrollbar, j'ai inséré 2 méthodes qui permet à la barre de fonctionner sur tous les navigateurs. La méthode la plus simple est celle du navigateur Firefox où on a besoin seulement comme balise de

la scrollbar-width en mode auto, la barre s'adapte automatiquement à la page web et du scrollbar-color qui donne les couleurs de la barre de navigation.

Pour les autres navigations, on a besoin de la balise webkit-scrollbar qui permet de donner la width (largeur) du scrollbar, de la balise webkit-scrollbar-track qui permet de donner la couleur de l'arrière-plan de la barre de navigation et de la balise webkit-scrollbar-thumb.

```
/* ===== Scrollbar CSS ===== */
/* Firefox */
* {
  scrollbar-width: auto;
  scrollbar-color: black #ffffff;
}

/* Chrome, Edge, and Safari */
*::-webkit-scrollbar {
  width: 16px;
}

*::-webkit-scrollbar-track {
  background: black;
}

*::-webkit-scrollbar-thumb {
  background-color: black;
  border-radius: 10px;
  border: 4px solid white;
}
```

Figure 20: Codage Scrollbar

III.1.5 FORMULAIRE EN CSS

Pour le formulaire, qui se trouve dans la page Contact, j'ai d'abord codé un grand carré avec un width (largeur) de 35% de la page, un height (hauteur) de 50vh avec un padding qui permet de définir les différents écarts de remplissages sur les quatre côtés de l'élément, on met évidemment display en position flex pour que le carré soit responsive et les balises margin-left et margin-right en auto pour centré le carré. Background-color permet de mettre l'arrière-plan dans un gris claire. La balise border-radius permet de mettre une bordure sur l'élément avec la taille et sa couleur noire.

```
.carrefourformulaire {
  width: 35%;
  height: 50vh;
  padding: 2%;
  display: flex;
  margin-left: auto;
  margin-right: auto;
  margin-top: 2%;
  bottom: 1vh;
  background: #d9d9d9;
  border: 1px solid #000000;
  border-radius: 63px;
  font-size: 1rem;
}
```

Figure 21: Codage du grand carré du formulaire

Ensuite, je créais une balise formulaire qui regroupe les balises communes pour les différentes étapes du formulaire. Le nom, le prénom et l'Email ont un élément qui sont les mêmes, donc on met à l'élément une hauteur de 100% avec une largeur de 4vh avec un margin-top (position sur la page) de 2vh avec un border-radius de 100px qui met donc des coins arrondis

```
.formulaire {  
  width: 100%;  
  height: 4vh;  
  margin-top: 2vh;  
  border-radius: 100px;  
}
```

Figure 22: Codage du formulaire

Pour finir, pour construire le code du bouton « envoyer », j'ai mis le background (arrière-plan) dans une couleur grise comme les autres éléments du formulaire, j'ai utilisé le style color pour mettre le texte en blanc, une balise border-radius qui arrondis les angles en 20px, un style cursor sur pointer qui permet à la souris de pointer lorsqu'elle passe sur le bouton.

```
button {  
  color: white;  
  background: #9b9999;  
  border-radius: 20px;  
  cursor: pointer;  
  font-weight: 900;  
  transition: 0.5s;  
  
  border: transparent;  
}
```

Figure 23: Codage bouton formulaire

III.2. HTML

En premier temps, dans le html, on utilise la balise head qui fournit des informations générales sur le site web, pas nécessairement lisible, par exemple mon head contient le link qui permet de relier le html au css

III.2.1 HEADER EN HTML

Après, on ajoute la balise body qui représente le contenu principal du document, on ajoute à l'intérieur la balise header qui représente le contenu du site web qui peut être par exemple des titres, logos ou même formulaire. J'ai ensuite ajouté la balise nav qui représente une section d'une page ayant des liens, dans mon cas c'était le header. Dans la nav, j'ajoute une div avec une class qu'on nomme « logo » qui permet de manipuler l'élément qu'on va mettre par le css. Dans cette div, j'ai ajouté la balise a avec son attribut href qui crée un lien hypertexte vers la page accueil pour la balise img qu'on assimile à l'image du logo. Après avoir fermé la div de class logo, j'ajoute la balise ul qui représente une liste d'éléments, dans cette balise ul j'ai ajouté 4 balises li qui est utilisé pour représenter un élément dans une liste, sur chaque balise li j'ai mis à chacun une balise a qui

représente chacun une page du site web hors accueil. Pour finir la liste je referme mes balises ul et nav. Pour terminer le header, j'ajoute une div classe du nom de insidebar avec ses différents propriété dans d'autre div pour relier les informations que j'ai écrit dans le css. Je ferme la balise header.

```
<body>
<header>
  <nav>
    <div class="logo">
      <a href="#logo"></a>
    </div>
    <ul>
      <li><a href="#presentation">Qui suis-je?</a></li>
      <li><a href="#click">Compétences</a></li>
      <li><a href="#point">Mes projets</a></li>
      <li><a href="#pour">Contact</a></li>
    </ul>
  </nav>

  <div class="insidebar">
    <div class="polling_message"></div>
    <div class="cylon_eye"></div>
  </div>
</header>
```

Figure 24: l'html du header

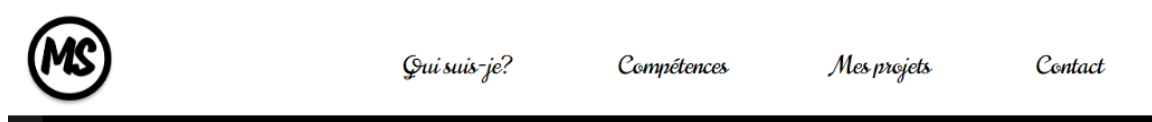


Figure 25: Résultat Header avec HTML + CSS sur le site web

III.2.2 LES DIFFERENTES PAGES EN HTML

A chaque page d'accueil compris, je leur donne une div avec une class avec leur nom de page et chaque élément à l'intérieure qu'elles ont besoin en rapport avec le css

La page d'accueil est constituée d'une div de class nommé accueil qui permet un lien avec le sélecteur de class accueil du css avec un id qui est un identifiant qui a comme pouvoir de crée un lien lorsqu'on sur l'élément de la nav dont le href correspond, le nom de l'id de la div class accueil correspond au logo donc l'image du logo, ce qui permet lorsqu'on clique sur le logo n'importe où on se trouve sur le site web de retourner à la page d'accueil en un clic. Toutes les pages ont leur id et href correspondantes.

```
<div class="logo">
  <a href="#logo"></a>
</div>
```

Figure 26: Href logo

```
<div class="accueil" id="logo">
  <p>
    Mélissa Simon
  </p>
</div>
```

Figure 27: HTML de la page d'accueil

Comme dit plus tôt la class accueil permet de relier le la div accueil de l'html à la div accueil du css, les différentes propriétés du sélecteur de classe joue sur la mise en forme de la page d'accueil et de la balise p.

```
.accueil {  
  height: 100vh;  
  background-color: #burlywood;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  font-size: 10rem;  
}
```

Figure 28: CSS de la page d'accueil



Figure 29: Page d'accueil avec HTML + CSS sur le site web

Ces multiples balises, sélecteurs de class permettent un lien entre les propriétés de html et du css.

III.3. Hébergement du site web

III.3.1 LIEN DE MON SITE WEB

<https://melsn03.github.io/>

III.3.2 GITHUB

L'hébergeur utilisé pour mon site web est github.com. GitHub est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git.

Pour héberger mon site sur GitHub, j'ai enregistré les différents fichiers de codage que j'ai effectué sur CodesAndBox, je les ai exportés sur GitHub

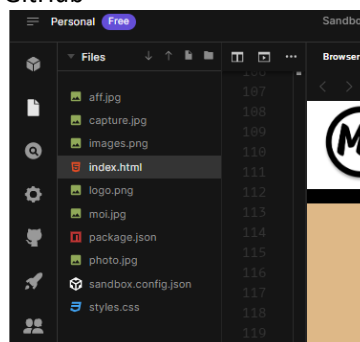


Figure 30: Différents fichiers de codages sur codesandbox

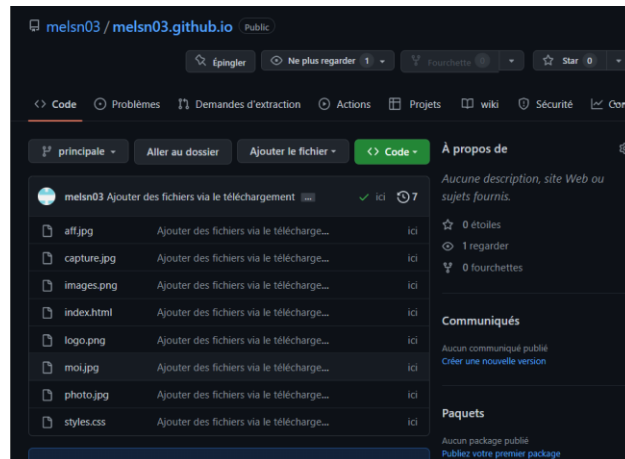


Figure 31: Les fichiers de CodesAndBox exporter sur mon serveur github

IV. Conclusion

Cette SAE 105 avaient pour but de faire acquérir l'étudiant en MMI1 plusieurs compétences comme la combinaison des ressources liées au développement et à l'intégration et comprendre comment elles s'articulent pour automatiser la production de sites web, une sensibilisation à la nécessité de respecter les normes et les bonnes pratiques de l'intégration web, de rendre les étudiants autonomes sur leur poste de travail en maîtrisant leur environnement de développement (édition, validation, inspection du code et hébergement local des sites).

V. Ressource

Figma :

https://www.figma.com/fr/design/?utm_source=google&utm_medium=cpc&utm_campaign=17451562552&utm_term=figma%20design&utm_content=602996802892&gclid=CjwKCAiAoL6eBhA3EiwAXDom5vPGWSSB5Midjv0rX8uIF4adymvdOcM1ajarHeOETNuskeuF_UiNTRoCI9cQAvD_BwE

CodesAndBox :

<https://codesandbox.io/discover>

GitHub :

<https://github.com/>