# <PROJET PAGE INTERNET PRODUIT EN HTML CSS/>

**J'ai choisi comme thème de mon site un site personel/CV. Présentant certains de mes hobbies, mais aussi mon milieu scolaire actuel, ainsi que mon CV et quelques créations réalisés depuis le début de l'année.**

**Chaque conteneur ou bouton est cliquable menant à une page internet que j'ai choisi et qui a un lien direct avec sa représentation sur mon site (par exemple lien vers la FFH pour le handball...)**

**Je présenterai le code HTML et la partie CSS qui lui est reliée, l'explication sera appuyée de capture d'écran de mon code.**

***Début du code :***

index.html > html > body > main > div.child-page-listing

```
1  <!DOCTYPE html>
2  <html>
3
4          <link rel="stylesheet" href="Structurescss.css"/>
5          <title> Projet HTML CSS</title>
6          <meta charset="UTF-8">
7          <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
8          <link href="https://fonts.googleapis.com/css2?family=Nerko+One&display=swap" rel="stylesheet">
9          <script src="https://kit.fontawesome.com/d5376e79b9.js" crossorigin="anonymous"></script>
10
```

**Dans cette première partie du code HTML nous posons simplement les bases de la construction. De plus les balises <link> présente dans les débuts du code permettent de faire un lien avec les ressources externe à la feuille HTML . Ici les balises renvoient à la feuille CSS permettant de structurer et designer la feuille HTML ainsi qu'à google fonts, nous permettant d'intégrer la police de notre choix.**
**La balise <script> permet d'intégrer un script executable généralement codés en JS. Ici ce Script proposés par font awesome permet d'insérer des pictogrammes dans notre intégration.**

***Côté CSS :***

# Structurescss.css > html

```
1
2   html, body {
3     width: 100%;
4     height: 100%;
5     margin: 0;
6     padding: 0;
7     background-color: rgb(65, 61, 6
8   }
```

**Pas-grand-chose ici, on règle simplement la largeur (width) et la hauteur (heigth) de la page a 100% pour que cela englobe le tout sur l'affichage du site internet. Permettant de ne pas laisser d'espace vide dans la page internet.**
**Le margin permet de définir la taille de la marge pour les quatre cotés de l'élement concernés : ici pour l'entièreté de la feuille. Le padding lui permet de paramétrer les hauteurs de remplissage. Si l'on veut que tout soit rempli correctement , il faut mettre le tout à 0, cela va de pair avec les deux premières propriétés. On aura ni marge ni hauteurs de remplissage de la feuille HTML . Enfin, on attribue une couleur à la feuille HTML, ici du gris.**

## *II. Le header*

**Le header est l'entête de mon site, dans lequel je vais construire une barre contenant les menus de notre site. Ces menus seront des boutons cliquables menant à des liens qui leur correspondent.**

### *Côté HTML (index 1) :*

```
<body>
    <header>
        <div class="box">
            <div class="div_menu" id="panel1">
                <button class="bouton_header">
                <div class="fs">
                    <i class="fa-solid fa-user" onclick="window.location.href ='file:///C:/Users/lucth/OneDrive/Bureau/structure%20site%20g
                </div>
                    </button>
        </div>
    <div class="div_menu">
            <input class="button_menu" type="button"  onclick="window.location.href ='file:///C:/Users/lucth/OneDrive/Bureau/structure%20si
              </div>

    <div class="div_menu">
            <input class ="button_menu" type="button" onclick="window.location.href ='https://www.telecom-st-etienne.fr/';" value="Ecole"><
              </div>

    <div class="div_menu">
              <input class="button_menu" type="button"   onclick="window.location.href= 'file:///C:/Users/lucth/OneDrive/Bureau/structure%2
              </div>

    <div class="div_menu">
            <input class="button_menu" type="button"  onclick="window.location.href ='https://www.telecom-st-etienne.fr/';" value="à propos
              </div>

        </div>
    </header>
```

**Tout au long de mon intégration j'utiliserai des , permettant de contenir plusieurs lignes de code, plusieurs balises. La  ne sert à rien à proprement parler hormis à structurer l'html mais aussi le css. Elle nous permettra de styliser ce qu'elle enveloppe. Mais, pour styliser des composantes particulière au sein d'une  j'utiliserai des <class>.**

**Ici, le code me permettant de créer mon en-tête est intégré dans une grande  nommé box.**

**Premierement je vais créer un boutton de classe «button» dans lequelle je vais intégrer un pictogramme venant du site «font awesome». Ce pictogramme est reliéa ma page d'accueil.**

**Dans cette div je vais créer la  «div_menu». Cette dernière va me permettre de styliser mes boutons. Au sein de cette  que j'ouvrirai pour chaque nouveau bouton je vais insérer une <class> «button_menu» pour la feuille CSS, ainsi que le «type» de la div qui sera «button». Donnant la fonction d'un boutton. Pour que l'utilsateur soit redirigé sur une page donné en cliquant sur le boutton, j'applique l'attribut d'évènement onclick, avec le lien correspondant à la page qui doit s'ouvrir a la suite du clic. La «value» indiquera la valeur et donc le texte inscrit sur ce boutton. Tout cela est intégré dans un <input> un élement permettant de créer un contrôle interactif.**

### *Côté CSS:*

```
.box {
    display: flex;
    }

.fs{
  color:□white;
  font-family: 'Nerko One', cursive;
}

button{
    position: relative;
    height: 50px;
    width: 150px;
    border: 0cm;
    background-color: transparent;

}
```

```
.div_menu {
    position: relative;
    height: 50px;
    width: 150px;
    left: 400px;

}

.button_menu {
    position: relative;
    height: 50px;
    width: 150px;
    border: 0cm;
    background-color: transparent;
    font-family: 'Nerko One', cursive;
    text-align:center ;
    font-weight: bold;
    color: □white
}
```

**Dans ma feuille CSS je vais dans un premier temps donner une couleur à l'arrière-plan de mon header, permettant de designer le tout comme je le veux. Ensuite, je vais donner la div globale un positionnement. Ma div est de class «box»dans le css elle aura donc le sélecteur «.». Le display...**
**La première class de style «button» permet de styliser la div boutton où l'on a intégré un pictogramme.**
**Dans la class «button»je vais paramétrer la largeur (width) et la hauteur (height) du bouton, je vais lui enlever ses bordures et mettre un fond transparent, ce qui me permettra de rajouter un effet par la suite.**

**Le «fs» concerne mon pictogramme, que je vais mettre en blanc. Les pictogramme de fontawesome ont les mêmes fonctionnalités que les polices «font».**

**Les deux class div_menu et button_menu vont me permettre de styliser les <input> contenant les autres boutons.**

**Au travers de la div_menu je vais positionner dans mon header mes input et mon boutton. Ainsi je vais leur donner tous la même hauteur et largeur, et je vais les positionner de 400 pixels en partant de la gauche.**

**Dans le button_menu je paramètre la position exacte de mon contenu de bouton, sa taille et sa hauteur en fonction de div_menu. J'ajoute aussi une police personnalisée qui s'affichera dans le boutton, je la met en gras et en blanc. J'enlève aussi les bordures et je mets le bouton de couleur transparente à défaut du gris automatique. Cela va me permettre d'appliquer un hover au dessus directement.**

***Hover:***

```
.button_menu:hover {
    background-color: rgba(78, 19, 255, 0.729);
}
.button:hover {
    background-color: rgba(78, 19, 255, 0.729);
}


.bouton_header {
    transition: all 0.2s;
}

.bouton_header:hover {
    transform:scale(1.5);
}
```

**Dans la suite de mon CSS, je vais styliser mes boutons avec un hover, c'est-à-dire un effet qui illumine de la couleur que je veux le bouton (comme un voile) quand la souris se passe dessus.**

**Mon bouton header lui, va être différent avec le «transition» je paramètre un temps d'animation, puis mon hover lui sera un changement de taille «transform:scale». Cela aura pour effet de faire grossir mon pictogramme.**

***Le grid (index 2) :***

**J'ai ensuite construit un grid ou j'ai inséré des images cliquables afin de décrire mes centres d'intérêts.**

***Côté HTML :***

```
<main>
      <div class="child-page-listing">

            <h2><i class="fa-solid fa-circle-info">Centres d'interets</i></h2>

      <div class="grid-container">

            <article class="location-listing">

                <a class="location-title" href="https://www.ffhandball.fr/"> Handball </a>

      <div class="location-image">

                  <img width="300" height="150" src="images/image.jpg" alt="Handball">

            </div>

            </article>
```

**Je vais créer une div «child-page-listing», et une div «grid-container». L'une va régir le bloc complet de mon grid, tandis que la deuxième sera en fait l'intérieur de chacun de mes parties du grid.**
**Je vais ensuite créer un <article> dans ma div me permettant de mettre n'importe quel type de contenu à l'intérieur, c'est au sein de cette <article> que je vais placer mes images, intégré dans une div appelé location d'images. . Je réglerai au préalable la hauteur et la largeur de chacune de mes images, pour obtenir quelque chose d'homogène. une balise <a> au sein de mon article me permettra aussi d'avoir une direction vers une page internet avec un rapport au mot (ici handball).**
**Exceptionnelement dasn cette div, j'intégrerai une balise <h> ou je vais taper mon texte de présentation de mon grid, liés a un pictogramme.**
**Je reproduirai 8 fois la création de l'article «location-listing» et de la div «location-image», me permettant de créer 8 blocs d'images différents.**

***Côté CSS :***

```
.grid-container {
    display: grid;
    grid-template-columns: repeat(auto-fill, minmax(300px, 1fr))
    grid-gap: 1em;
  }
img {
    width: 100%;
    height: fit-content;
}

h2 {
  text-align: center;
  font-family: 'Nerko One', cursive;
  font-weight: bold;
  font-size: medium;
  color : white;
}
```

**Dans le CSS je vais pouvoir positionner ma div grid-container dans un grid, avec la commande display grid, je vais aussi paramétrer l'espace entre chaque bloc ainsi que la commande auto-fill pour mettre le contenant de l'image à la taille donné, soit ici 300 pixel. Le 1 fr désigne la proportionalité de l'espace entre chaque grille dans la colonne.**
**Je règle ma classe image pour qu'elle puisse prendre 100% du cadre par rapport à son format et qu'elle prenne toute la place disponible.**

**Dans «h2», je règle la police et le pictogramme que j'ai inséré, je le met en blanc avec la police qui me convient ainsi que la taille police pour le texte qui accompagne le pictogramme.**

***Effet du grid :***

```css
.location-image img {
  filter: blur(0px);
  transition: filter 0.3s ease-in;
  transform: scale(1.1);
}

.location-title {
  font-size: 1.5em;
  font-weight: bold;
  text-decoration: none;
  font-family: 'Nerko One', cursive;
  z-index: 1;
  position: absolute;
  height: 100%;
  width: 100%;
  top: 0;
  left: 0;
  opacity: 0;
  transition: opacity .5s;
  background: rgba(78, 19, 255, 0.729);
  color: white;

  /* text au milieu*/
  display: flex;
  align-items: center;
  justify-content: center;
}
```

**Dans cette partie du CSS je vais appliquer des effets sur mes cases avec images et texte.**

**Je vais appliquer dans ma class «location-image» un filter «blurr». Cette propriété me permet d'appliquer un flou gaussien sur mon image. La transition ansi que le transform vont me permettre d'avoir une application du filtre sur une durée de 0,3 secondes, puis de voir un petit agrandissement rendant une ésthetique plus lisse quand la souris passe dessus.**

**Dans ma class «location-title» je vais d'abord paramétrer la taille de la police et sa couleur ainsi que le gras. le «text-decoration» me permet de paramétrer le fait de ne rien ajouter de plus, comme un soulignement ou autre. J'ajoute aussi la police que j'ai posée dans le reste de mon code, afin de personnaliser encore plus le tout. Je règle la hauteur et la largeur de ma police, son opacité pour venir contraster avec le hover posé précedemment. Les paramètres «transition: opacity» et «background» me permettent de régler la couleur de mon hover, ici en bleu.**

**Enfin la dernière ligne de code me permet d'imposer une position «flex» a mon texte, qui va me permettre de le manipuler librement. Je le justifie et le place au centre de mes grilles de grid.**

***Carrousel (index 3) :***

**Dans la dernière partie de mon code j'ai décidé d'ajouter un petit carrousel pour présenter quelques petits projets réalisés en graphismes.**

***Côté HTML :***

```
<br>
<br>
<br>
<br>
<br>

<h1> PROJETS REALISES</h1>
  <div class="slider-container">
      <div class="menu">
          <label for="slide-dot-1"></label>
          <label for="slide-dot-2"></label>
          <label for="slide-dot-3"></label>
      </div>

      <input class="slide-input" id="slide-dot-1" type="radio" name="slides" checked>
      <img class="slide-img" src="images/chat_wars.jpg">

      <input class="slide-input" id="slide-dot-2" type="radio" name="slides">
      <img class="slide-img" src="images/snk.png">

      <input class="slide-input" id="slide-dot-3" type="radio" name="slides">
      <img class="slide-img" src="images/logo.png">

      </div>
```

**Les <br> nous permettent de sauter des lignes dans la page, permettant à tout les éléments de ne pas se coller. Je crée une commande <h1> dans laquelle je vais mettre le texte préentant le contenu du carrousel.**

**Par la suite je vais crée une class «slider-container». C'est au sein de cette dernière que nous allons construire le carrousel.**
**Je vais créer une div «menu» qui va nous servir a insérer des «label». Le «label» est une légende pour un objet, de plus il permet un contrôle sur cet objet à l'aide du for. je vais donc controler les «slidde-dot» qui correspondent au point permettant de défiler le carrousel grâce au <label>.**
**Ensuite, je vais créer des objets manipulable avec l'<input>, je vais donne rune class a ces objets et je vais les faire correspondre au label grâce à l' «id». De plus je vais insérer des images dedans en leur créant une class me permettant de les manipuler par la suite.**
**Notre carrousel est crée, sans l'utilisation de Js il est fonctionnel mais peut développé.**

***Côté CSS:***

```
.slider-container {
  max-width: 500px;
  position: relative;
  margin: auto;
  height: 350px;
  overflow: hidden;
}
.menu {
  position: absolute;
  left: 0;
  z-index: 11;
  width: 100%;
  bottom: 0;
  text-align: center;
}
.menu label {
  cursor: pointer;
  display: inline-block;
  width: 10px;
  height: 10px;
  background: #ccc;
  border-radius: 50px;
  margin: 0 0.2em 1em;
}
```

```
.menu label:hover,.menu label:focus {
  background: #1c87c9;
}
.slide-input{
  opacity: 0;
}
.slide-img {
  width: 100%;
  height: 300px;
  position: absolute;
  top: 0;
  left: 100%;
  z-index: 10;
  transition: left 0s 0.75s;
}
[id^="slide"]:checked + .slide-img {
  left: 0;
  z-index: 100;
  transition: left 0.65s ease-out;
}
```

Je vais dans mon CSS, styliser mon carrousel afin de le rendre propre et permettre aux images de rentrer correctement dedans.

Premièrement je vais styliser ma class «slider-container». Je vais lui donner une hauteur de 350 pixels ainsi qu'une contenance maximale d'une largeur de 500 pixels. une marge auto, ainsi qu'une propriété «overflow : hidden» qui cachera les parties du contenu qui dépasseront de mon slide.

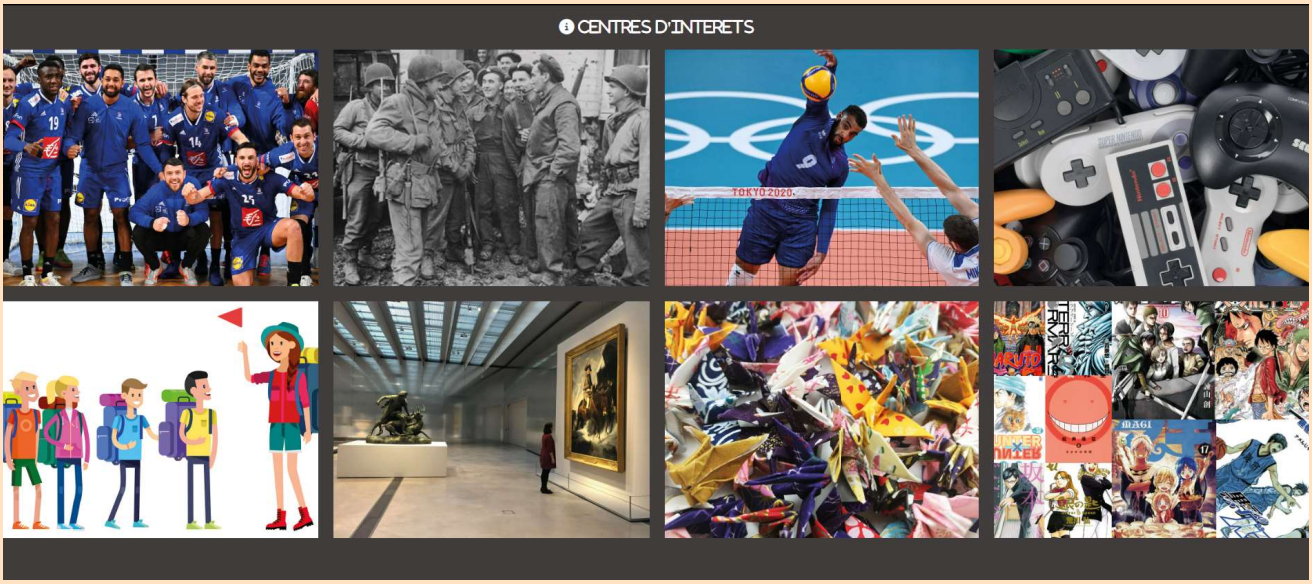
La class menu sera paramétré de manière à ce que sa position soit absolue, avec un texte centré. J'insére un Z-index permettant à cet élément de se mettre au premier plan. Tout ce qui entrera dans la class «menu» sera au premier plan par rapport aux restes.

Dans mon «menu label» je vais gérer la taille des «slide-dot», des petits points me permettant de faire défiler mes images. Je leur donne une taille, une couleur ainsi qu'une propriété «cursor : pointer» qui va changer l'apparence du curseur quand il passera sur le point.  J'applique ensuite un hover sur ces points.
Ensuite dans «slide img», je vais régler la taille des image, leur position, ainsi que le temps d'animation d'un slide a l'autre. Je fais bien attention à ce que mes images s'adaptent au format de mes slides précedemment paramétré.

Enfin, ma dernière ligne de CSS me permet de styliser complétement le slide. Je met un Z-index très haut qui va faire en sorte que l'image qui suit la premiere se mette complétement au dessus, au premier plan. je lui dis aussi d'aller vers la gauche, en paramétrant le temps d'arrivée de l'objet.

# <INDEX 1/>

# <INDEX 2/>



# <INDEX 3/>

PROJETS REALISES