BERNIER

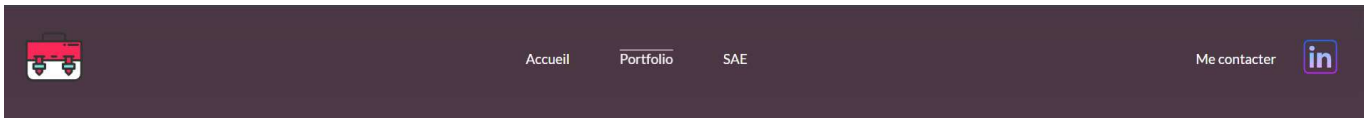
Axel

TP12

SAE 105

Justification de l'AC14.02

J'ai pris soin de bien choisir les balises sémantiques utilisées pour coder mon site. J'ai utilisé une structure classique, avec un `header` pour les éléments en tête de page, une balise `<main>` pour les éléments principaux, et un `<footer>` pour les éléments en bas de page. Cela me permet d'avoir une structure claire. De plus, j'ai utilisé des titres de niveau pour que ma page soit correctement hiérarchisée. Cela permet, au niveau professionnel, d'améliorer l'accessibilité. J'ai pris soin d'utiliser les balises `alt` pour les images, afin de favoriser le référencement naturel de mon site.




```
<header>
    <div class="linkedin">
        <div>
            <a href= mailto:axelbernier53@gmail.com?subject=HTML link >Me contacter</a>
        </div>

        <!--Contient Le Logo linkedin allant sur mon profil-->
        <a href="https://www.linkedin.com/in/axel-bernier-94533b328/" class="linkedinprofil" target="_blank">
            <svg xmlns="http://www.w3.org/2000/svg" x="0px" y="0px" width="60" height="60" viewBox="0 0 64 64">
                <linearGradient id="SUJNhpmDQDF27Y3OfwgfYa_44019_gr1" x1="19" x2="19" y1="24.858" y2="49.041"
                    gradientUnits="userSpaceOnUse" spreadMethod="reflect">
                    <stop offset="0" stop-color="#6dc7ff"></stop>
                    <stop offset="1" stop-color="#e6abff"></stop>
                </linearGradient>
                <path fill="url(#SUJNhpmDQDF27Y3OfwgfYa_44019_gr1)" fill-rule="evenodd"
                    d="M22 48L22 26 16 26 16 48 22 48z" clip-rule="evenodd"></path>
                <linearGradient id="SUJNhpmDQDF27Y3OfwgfYb_44019_gr2" x1="19.382" x2="19.382" y1="15.423"
                    y2="23.341" gradientUnits="userSpaceOnUse" spreadMethod="reflect">
                    <stop offset="0" stop-color="#6dc7ff"></stop>
                    <stop offset="1" stop-color="#e6abff"></stop>
                </linearGradient>
                <path fill="url(#SUJNhpmDQDF27Y3OfwgfYb_44019_gr2)" fill-rule="evenodd"
                    d="M19.358,23c2.512,0,4.076-1.474,4.076-3.554 c-0.047-2.126-1.564-3.649-4.028-3.649c-2.465,0-4.076,1.475-4.076,3.601c0,2.08,1.563,3.602,3.981,3.602H19.358L19.358,23z"
                    clip-rule="evenodd"></path>
                <linearGradient id="SUJNhpmDQDF27Y3OfwgfYc_44019_gr3" x1="37.386" x2="37.386" y1="14.125"
                    y2="49.525" gradientUnits="userSpaceOnUse" spreadMethod="reflect">
                    <stop offset="0" stop-color="#6dc7ff"></stop>
                    <stop offset="1" stop-color="#e6abff"></stop>
                </linearGradient>
                <path fill="url(#SUJNhpmDQDF27Y3OfwgfYc_44019_gr3)" fill-rule="evenodd"
                    d="M26.946,48H34V35.911c0-0.648,0.122-1.295,0.313-1.758 c0.52-1.295,1.877-2.635,3.867-2.635c2.607,0,3.821,1.988,3.821,4.901V48h6V35.588c0-6.657-3.085-9.498-7.826-9.498 c-3
                    clip-rule="evenodd"></path>
                <linearGradient id="SUJNhpmDQDF27Y3OfwgfYd_44019_gr4" x1="32" x2="32" y1="6.5" y2="57.5"
                    gradientUnits="userSpaceOnUse" spreadMethod="reflect">
                    <stop offset="0" stop-color="#1a6dff"></stop>
                    <stop offset="1" stop-color="#c822ff"></stop>
                </linearGradient>
                <path fill="url(#SUJNhpmDQDF27Y3OfwgfYd_44019_gr4)"
                    d="M50,57H14c-3.859,0-7-3.141-7-7V14c0-3.859,3.141-7,7-7h36c3.859,0,7,3.141,7,7v36 C57,53.859,53.859,57,50,57z M14,9c-2.757,0-5,2.243-5,5v36c0,2.757,2.243,5,5,5h36c2.757,0
                </path>
            </svg>
        </a>

    </div>
</header>
```

Voici une capture d'écran de mon `header` où l'on peut voir la disposition de mes éléments. J'ai utilisé des balises `<a>` pour les liens, en faisant attention d'utiliser des liens relatifs. Cela permet d'inclure des adresses fonctionnant en fonction de l'adresse du fichier de destination, pouvant être visualisées sur d'autres machines. De plus, j'ai pris le soin de bien indenter mon code pour favoriser la clarté de celui-ci, notamment en utilisant le raccourci : `Shift + Alt + F`.

J'ai également ajouté des commentaires dans mon code HTML, CSS et JavaScript pour pouvoir naviguer plus rapidement à l'intérieur de celui-ci. Cela facilite les modifications ultérieures sans perdre de temps à me rappeler comment j'ai réalisé la structure. Voici quelques captures d'écran attestant de la mise en place des différents commentaires.

```
<main>
    <div class="info"> <!--Div contenat le texte de présentation-->
        <div class="flex">
            <div class="profil">
```

```
<footer> <!-- Contient tous mes projets personnel (photos, vidéos...)-->
        <div class="projet" id="section1">
            <div class="cercle">
```

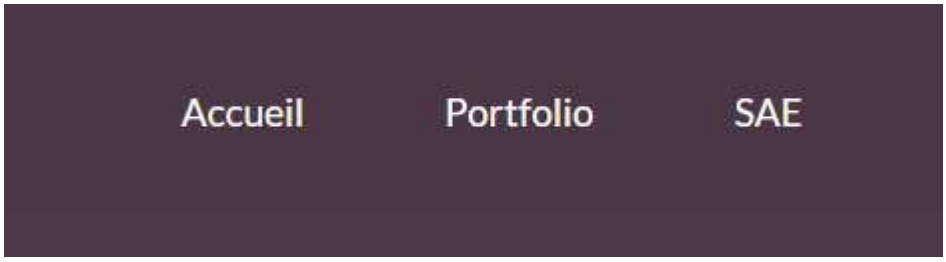
```
/*Barre de navigation*/
header {
    display: flex;
    align-items: center;
    justify-content: space-between;
    padding: 20px;
    margin: 10px;
    gap: 20px;
}
```

```
/* contient les liens relatifs de navigation*/
.centre {
    display: flex;
    gap: 50px;
}
```

```
let listeapr = "" // permet d'afficher les AC
Object.keys(SAE[cles]['AC']).forEach(function (element) {// permet d'afficher dynamiquement le contenu des AC
    let lienpdf = `../pdf/${element}.pdf`; //création du lien pour récupérer dynamiquement les PDF
    listeapr += `<a href="${lienpdf}" target="_blank" class = "link-apr"><div class = "div-apr">${element} : ${
});
```

J'ai aussi pris soin d'ajouter des animations et une navigation intuitive à mon site web. Par exemple, sur la page des SAE, il y a un bouton de retour à l'accueil, et sur la page de description des SAE, un bouton permet de revenir vers la page des SAE, assurant une navigation fluide et intuitive.

Également, la barre de navigation présente dans le `header` reprend une structure simple et classique, permettant d'accéder aux différentes pages de manière claire et sans fioritures.



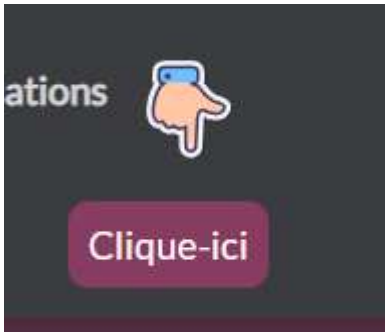
J'ai aussi ajouté des animations pour dynamiser mon site et améliorer l'expérience utilisateur. Par exemple, avec l'utilisation de `hover` en CSS, j'ai pu modifier le style d'un bouton lorsqu'on le survole.

```css
.button-crea {
   display: flex;
   justify-content: end;
   ;
   margin-top: 20px;
   margin-right: 100px;
   text-decoration: none;
   color: white;
}

html {
   scroll-behavior: smooth;
}

.button-crea>div {
   background-color: #893c62;
   padding: 10px;
   border-radius: 10px;
   transition: box-shadow 0.3s ease;
}

.button-crea>div:hover {
   transform: scale(1.1);
   transition-duration: 0.2s;
   box-shadow: 0 0 10px #893c62, 0 0 20px #893c62;
```



Lors du survol du bouton, j'ai décidé de changer la couleur, de rajouter un effet d'ombre et des effets de transformation avec les commandes `transform` et `transition-duration` pour animer le bouton. J'ai utilisé des unités relatives comme `em` et des pourcentages pour que mes éléments s'adaptent à l'écran. De plus, j'ai employé les commandes `position: absolute` et `position: relative`.

Ces propriétés permettent de créer des mises en page dynamiques et fluides en positionnant précisément des éléments tout en respectant la sémantique du document HTML. Par exemple, `position: relative` a été utilisée pour ajuster des éléments tout en conservant leur contexte dans le flux du document, garantissant ainsi une structure compréhensible. En complément, `position: absolute` a servi à placer des éléments tels que des boutons ou des menus à des emplacements précis, créant des interactions visuelles simples et intuitives.

Je réponds donc à tous les critères pour créer des pages web fluides, respectant les normes basiques d'un balisage sémantique efficace, avec l'ajout d'interactions simples comme les animations CSS, des liens relatifs permettant de naviguer entre différentes pages en cliquant sur un bouton, et l'utilisation de propriétés de mise en page dynamique.