BERNIER

Axel

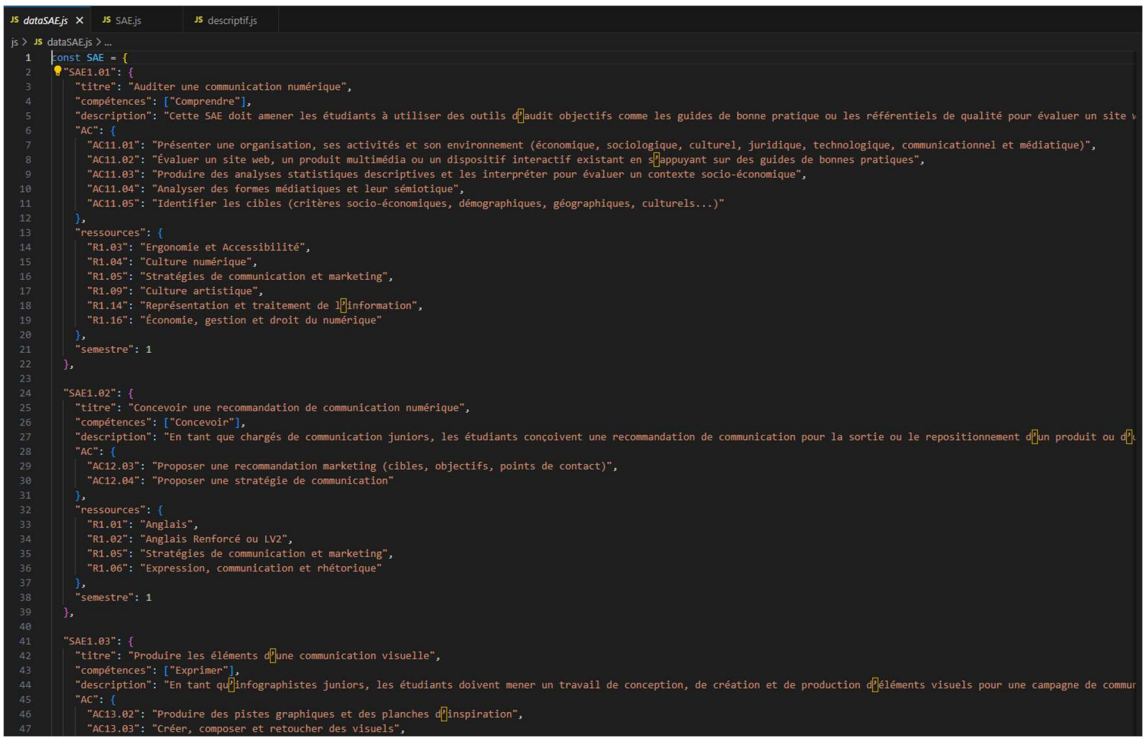
TP12

SAE 105

Justification de l'AC14.03

J'ai pu générer les différentes SAE sur la page qui leur est dédiée en utilisant un objet structuré en JavaScript fourni par le professeur. Cela évite de tout coder en HTML/CSS et permet d'afficher dynamiquement le contenu des SAE. Même en ajoutant de nouveaux objets à la constante SAE par la suite, ils seront automatiquement intégrés à la page, ce qui est très pratique.

La constante SAE contient plusieurs objets représentant les différentes SAE, comme SAE1.01 ou SAE1.02. Chaque objet possède des clés qui décrivent ses caractéristiques, telles que les compétences associées, les AC, le titre, etc.



```
const SAE = {
  "SAE1.01": {
    "titre": "Auditer une communication numérique",
    "compétences": ["Comprendre"],
    "description": "Cette SAE doit amener les étudiants à utiliser des outils d'audit objectifs comme les guides de bonne pratique ou les référentiels de qualité pour évaluer un site
    "AC": {
      "AC11.01": "Présenter une organisation, ses activités et son environnement (économique, sociologique, culturel, juridique, technologique, communicationnel et médiatique)",
      "AC11.02": "Évaluer un site web, un produit multimédia ou un dispositif interactif existant en s'appuyant sur des guides de bonnes pratiques",
      "AC11.03": "Produire des analyses statistiques descriptives et les interpréter pour évaluer un contexte socio-économique",
      "AC11.04": "Analyser des formes médiatiques et leur sémiotique",
      "AC11.05": "Identifier les cibles (critères socio-économiques, démographiques, géographiques, culturels...)"
    },
    "ressources": {
      "R1.03": "Ergonomie et Accessibilité",
      "R1.04": "Culture numérique",
      "R1.05": "Stratégies de communication et marketing",
      "R1.09": "Culture artistique",
      "R1.14": "Représentation et traitement de l'information",
      "R1.16": "Économie, gestion et droit du numérique"
    },
    "semestre": 1
  },

  "SAE1.02": {
    "titre": "Concevoir une recommandation de communication numérique",
    "compétences": ["Concevoir"],
    "description": "En tant que chargés de communication juniors, les étudiants conçoivent une recommandation de communication pour la sortie ou le repositionnement d'un produit ou d'
    "AC": {
      "AC12.03": "Proposer une recommandation marketing (cibles, objectifs, points de contact)",
      "AC12.04": "Proposer une stratégie de communication"
    },
    "ressources": {
      "R1.01": "Anglais",
      "R1.02": "Anglais Renforcé ou LV2",
      "R1.05": "Stratégies de communication et marketing",
      "R1.06": "Expression, communication et rhétorique"
    },
    "semestre": 1
  },

  "SAE1.03": {
    "titre": "Produire les éléments d'une communication visuelle",
    "compétences": ["Exprimer"],
    "description": "En tant qu'infographistes juniors, les étudiants doivent mener un travail de conception, de création et de production d'éléments visuels pour une campagne de commu
    "AC": {
      "AC13.02": "Produire des pistes graphiques et des planches d'inspiration",
      "AC13.03": "Créer, composer et retoucher des visuels",
```

Voici une capture d'écran du fichier **data.js** ci-dessous, qui montre la structure de l'objet SAE et ses différentes caractéristiques. Grâce aux données structurées contenues dans cette constante, il est possible d'afficher dynamiquement le contenu des SAE au chargement de la page.

En exploitant cette structure, j'ai créé une fonction appelée clés dans laquelle j'ai défini deux variables :

- **comp** : qui contient les compétences.
- **liste** : pour initialiser la fonction.

```
Object.keys(SAE).forEach(function (cles) {
    let comp = SAE[cles].compétences;
    let liste = "";
```

Je les affiche ensuite dans la section **main** de ma page et crée une div avec la classe SAE afin de leur appliquer un style spécifique.

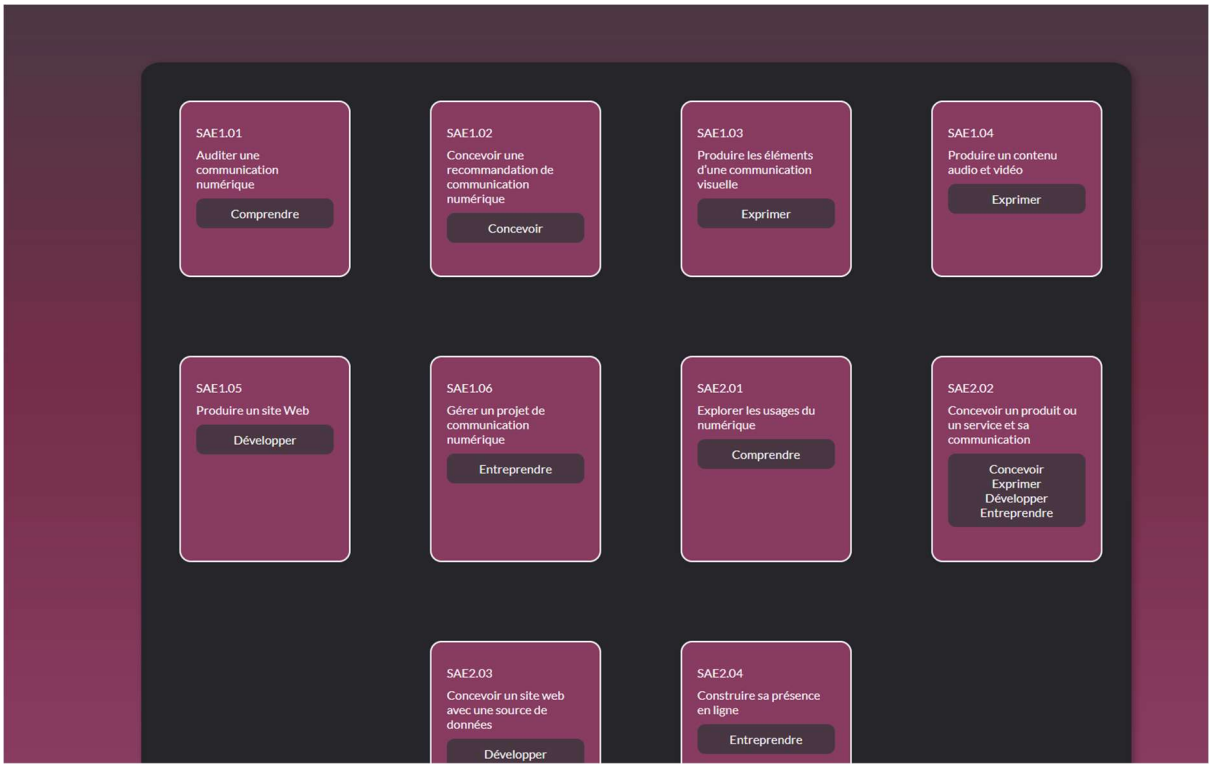
Comme on peut le voir ci-dessous, j'ajoute mes balises HTML pour insérer des liens et afficher les différents objets et leurs clés. J'utilise une boucle **forEach** pour parcourir les éléments de manière efficace, sans avoir besoin d'incréments dans la fonction. L'utilisation

de littéraux de gabarits (template literals) m’a permis de maintenir un code clair, propre et bien indenté.

```
    document.querySelector("main").innerHTML +=
        `<div class="sae">
        <a href='../page/descriptif.html?SAE=${cles}' class='lien'>
                    <div>${cles}</div>
                    <div>${SAE[cles].titre}</div>
                    <div class="style">${liste}</div>

        </a>
    </div>`
});
```

Voici le rendu final :



De plus, chaque SAE est composée d'apprentissages critiques (AC), eux-mêmes justifiés par des fichiers PDF. Ces derniers sont affichés dynamiquement en fonction des paramètres passés, tout comme la description de chaque SAE.

Pour cela :

- Les variables **listeapr** et **listress** me permettent d'afficher les AC et les ressources des SAE.
- La variable **lienpdf** génère dynamiquement un lien pour récupérer les fichiers PDF.

```
let param = new URLSearchParams(location.search);
let cles = param.get('SAE');

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} : ${SAE[cles]['AC'][element]}</div></a>`;
});

let listress = "" //permet d'afficher les ressources
Object.keys(SAE[cles]['ressources']).forEach(function (contenu) {
    listress += `<div class = "div-ressource"> ${contenu} : ${SAE[cles]['ressources'][contenu]}</div>`
});
```

L'utilisation de la propriété « console.log() » m'assure de vérifier dans l'inspecteur de mon navigateur de m'assurer que mon code JS fonctionne.

L'utilisation de la propriété **console.log()** m'a permis de vérifier, via l'inspecteur de mon navigateur, que mon code JavaScript fonctionnait correctement.

Enfin, j'ai ajouté les balises nécessaires pour structurer l'affichage dans la section **main**, tout en leur appliquant le style souhaité.

```
document.querySelector("main").innerHTML += `
    <h2>${cles}</h2>
    <div class='titreSAE'>${SAE[cles].titre}</div>
    <div class='compétences'>Compétence(s) : ${SAE[cles].compétences}</div>
    <div class='flex'>
        <div class='description'>
            <div>Description :</div><br>
            ${SAE[cles].description}
        </div>
        <div class='semestre'>
            Semestre :
            ${SAE[cles].semestre}
        </div>
    </div>
    <div class='flex-2'>
        <div class='AC'>
            <div class='titreac'>Apprentissages critiques :</div>
            ${listeapr}
        </div>
        <div class='ressources'>
            <div class='titreress'>Ressources :</div>
            ${listress}
        </div>
    </div>
`;
```

Ainsi, mes pages web ont bien été générées à partir de données structurées, ce qui favorise le dynamisme de mes pages et améliore l'expérience utilisateur.