# AC14.03: Générer des pages Web à partir de données structurées

Plutôt que de coder manuellement chaque élément en HTML, les informations nécessaires (listes de SAE, contenu détaillé, numéros d'AC et ressources) sont stockées dans un fichier de données (DATA). JavaScript est ensuite utilisé pour générer dynamiquement ces éléments en fonction des données, garantissant ainsi que le site reste à jour avec un minimum d'effort. Cette méthode réduit les risques d'erreurs et facilite l'ajout de nouvelles informations.

### 1. Liste SAE dynamique

#### Justification:

La liste des SAE est affichée en étant récupéré dans les données d'un fichier DATA. Cela évite d'écrire le code HTML manuellement pour chaque élément. Lorsqu'un nouvel élément est ajouté au DATA, il est automatiquement inclus dans la liste générée. Cela permet une réduction des erreurs humaines. Le site devient plus facile à maintenir, car l'ajout de nouvelles SAE se fait directement dans les données du DATA sans toucher au code HTML. De plus, cette méthode garantit que la liste reste toujours à jour.

### Exemple:

Base de données :

Script pour générer la liste des SAE :

```
js > J5 scriptjs > ...

SAE["SAE1.01"].titre

Object.keys(SAE).forEach(function (numSAE) {

let compé = '';

SAE[numSAE].compétences.forEach(function (competence) {

compé += '<div class="competence">' + competence + '</div>';

});

document.querySelector(".carre").innerHTML += '<a href="page3.html?sae=' + numSAE + '">' + '<div class="part"><div class="numSAE">' + numSAE + '</div></div
class="titre">' + SAE[numSAE].titre + '</div>' + compé + '</div></a>';

10

11

});

12
```

#### Listes des SAE, rendu final:



## 2. Contenu SAE dynamique

### Justification:

Dans le contenu de chaque SAE, il y a le titre, les compétences, une description, les apprentissages critiques, les ressources et le semestre. Plutôt que de coder tout cela en HTML, on utilise le DATA pour organiser les données et JavaScript pour générer dynamiquement les sections de contenu. Cela simplifie la mise à jour des informations : au lieu de modifier le code HTML, il suffit de mettre à jour le fichier DATA.

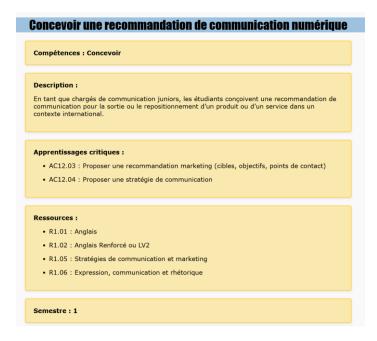
### Exemple:

Script pour afficher le contenu de chaque SAE :

```
let titre = (SAE[sae].titre);
let description = (SAE[sae].description);
let competence = (SAE[sae].compétences);
let ac = ""
let ressource = ""
let semestre = (SAE[sae].semestre)

document.querySelector("h1").innerHTML = titre
document.querySelector(".compétences").innerHTML = "Compétences : " + competence
document.querySelector(".description").innerHTML = "Description : " + description + ""
document.querySelector(".AC").innerHTML = "Apprentissages critiques : " + ac
document.querySelector(".ressources").innerHTML = "Ressources : " + ressource
document.querySelector(".semestre").innerHTML = "Semestre : " + semestre
```

#### Résultat avec la SAE 105 :



## 3. Affichage des numéros des AC et ressources

### Justification:

L'affichage des numéros des AC et des ressources se fait par JavaScript à l'aide de la base de données du DATA. Chaque AC renvoie vers un lien PDF différent expliquant chaque AC de chaque SAE. Cela permet de lier des fichiers PDF à des numéros spécifiques, de manière flexible et dynamique. Cette méthode est plus propre et permet d'intégrer de nouvelles informations sans modifier le code HTML.

## Exemple:

Script pour afficher les numéros des AC et les ressources, avec un lien vers les PDF pour chaque AC :

```
Object.keys(SAE[sae]["AC"]).forEach(function (AC) {
    ac += "<a href='PDF/" + AC + ".pdf' class= 'pdf' target= '_blank'>' + AC + " : " + SAE[sae]["AC"][AC] + "
Object.keys(SAE[sae]["ressources"]).forEach(function (ressources) {
    ressource += "' + ressources + " : " + SAE[sae]["ressources"][ressources] + "
});
```