

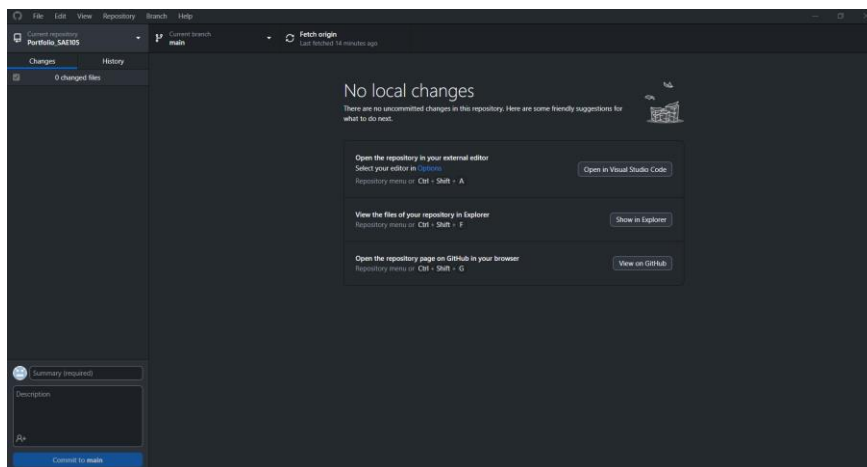
SAÉ 1.05 | Produire un site Web

En tant qu'intégrateurs ou développeurs juniors dans une agence Web, les étudiants doivent mener un travail de conception, d'intégration et de développement des vues d'un site Web. Le cahier des charges comprend la description des éléments statiques de l'application et un jeu de données structurées représentant un échantillon des contenus à mettre en ligne. Les étudiants doivent répondre à la question :

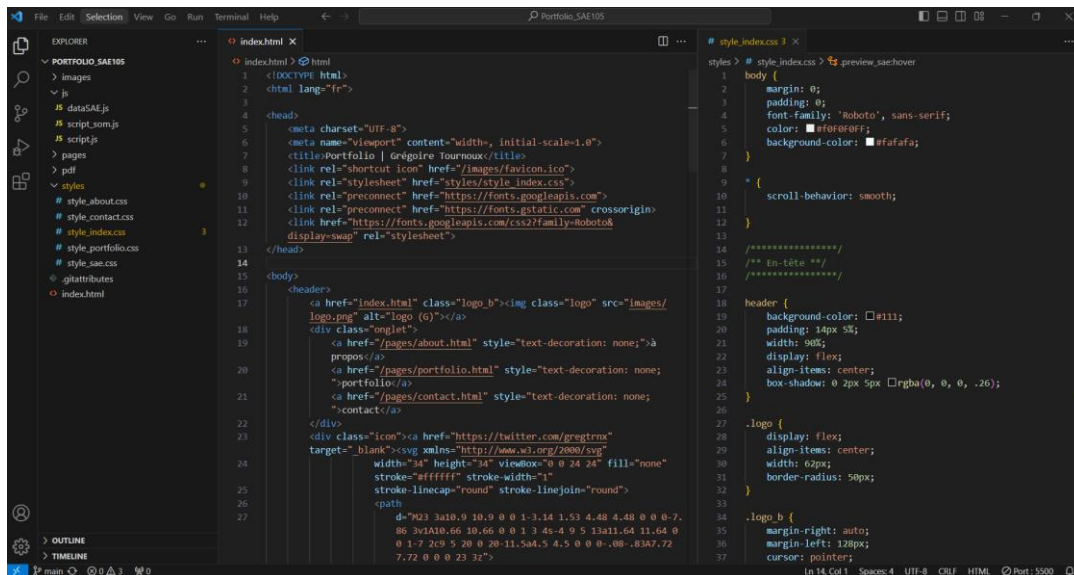
Comment produire efficacement un grand nombre de pages Web, valides et accessibles en automatisant autant que possible le travail d'intégration ?

AC14.01 | Exploiter de manière autonome un environnement de développement efficace et productif

J'ai travaillé de manière indépendante pour configurer un environnement de développement efficace, utilisant des outils tels que « GitHub Desktop » et « Visual Studio Code ». Cela m'a permis de gagner en productivité et d'assurer une intégration fluide du site.



GitHub Desktop me permet de travailler sur n'importe quelle poste de travail (à mon domicile ou à l'IUT de Mulhouse).



Visual Studio Code me permet de réaliser mon intégration avec une interface visuelle agréable et performante, Il est intégré nativement à Git.

AC14.02 | Produire des pages Web fluides incluant un balisage sémantique efficace et des interactions simples

J'ai élaboré des pages Web avec une approche axée sur la fluidité de l'expérience utilisateur. Le balisage sémantique a été appliqué de manière efficace pour améliorer l'accessibilité et la compréhension du contenu. Les interactions simples ont été intégrées pour garantir une navigation intuitive. Voici un exemple d'interactions simples :



Exemple de balisage sémantique (<main> ; <section> ...) :

```
<main>
  <div class="texte">
    <div class="marg">Découvrez</div>
    <div>mon</div>
    <div class="marg">parcours</div>
    <a href="/pages/portfolio.html" style="text-decoration: none;">
      <div class="btn">En savoir plus</div>
    </a>
  </div>
</main>
<section class="bas_main">
  <div class="last_info">
    <div class="last_head">
      <div class="border_deco"></div>
      <h3>certains de mes derniers travaux</h3>
      <div class="border_deco"></div>
    </div>
    <div class="last_content">
      <a href="/pages/sae.html?number=SAE1.01" style="text-decoration: none;">
        <div class="preview_sae">
          <div class="img_preview"></div>
          <div>Audit communication numérique</div>
        </div>
      </a>
      <a href="/pages/sae.html?number=SAE1.02" style="text-decoration: none;">
        <div class="preview_sae">
          <div class="img_preview"></div>
          <div>Design Graphique</div>
        </div>
      </a>
    </div>
  </div>
</section>
```

En intégrant ces balises, je décris la structure logique et le rôle de chaque partie de ma page, améliorant ainsi la qualité et l'efficacité de mon code HTML.

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

Pour automatiser le processus d'intégration, j'ai mis en place des scripts pour générer des pages Web à partir de données structurées. Cela a permis une gestion efficace du contenu et la possibilité d'ajouter rapidement un grand nombre de pages tout en maintenant la cohérence.

```
JS script.js > ...
// Paramètres URL
let urlParams = new URLSearchParams(location.search);
let saeNumber = urlParams.get('number'); // 'number' vient de sae.html?number=SAE1.01.
// Prendre les infos de l'objet SAE en fonction du nombre
let saeInfo = SAE[saeNumber];
let presaElement = document.querySelector('.presa');

// Mettre à jour le contenu de l'élément ciblé
presaElement.innerHTML = '<h1>' + saeInfo.titre + '</h1>' +
  '<h2>Compétences : ' + saeInfo.compétences.join(', ') + '</h2>' +
  '<h3>Description : ' + saeInfo.description + '</h3>' +
  '<p>AC : ' + Object.values(saeInfo.AC).join(', ') + '</p>' +
  '<p>Ressources : ' + Object.values(saeInfo.ressources).join(', ') + '</p>' +
  '<p>Semestre : ' + saeInfo.semestre + '</p>' +
  '<a class="pdf" href="' + saeInfo.pdfUrl + '" target="_blank" style="text-decoration: none;">Lien vers le PDF</a>';
```

AC14.04 | Mettre en ligne une application Web en utilisant une solution d'hébergement standard

J'ai opté pour l'utilisation de WinSCP, qui me permet de mettre en ligne mon site localement sur tout le réseau de l'IUT. Cela à nécessiter très peu d'étape voir ci-dessous :

Pour mettre en ligne le site que vous êtes en train de développer, il suffit de transférer les fichiers HTML, CSS, JS, ... sur le serveur **iutmmmiweb03.uha.fr**.

Vous disposez d'un accès FTPS par authentification sur ce serveur avec les identifiants UHA habituels.

Ouvrez l'application WinSCP et créez un "Nouveau site" en indiquant :

- Protocole de fichier : **FTP**
- Chiffrement : **TLS/SSL explicite**
- Nom d'hôte : **iutmmmiweb03.uha.fr**
- Numéro de port : **21**
- Nom d'utilisateur : **prénom.nom@uha.fr**
- Mot de passe : laisser ce champ vide
- Cliquez sur **Connexion**.
 - A la première connexion, WinSCP vous demande d'Enregistrer le certificat.
 - Cliquez sur **Oui**.
- Indiquez votre mot de passe habituel.

Après vous être connecté, WinSCP présente deux fenêtres :

- A gauche : l'arborescence de votre machine locale.
- A droite : l'espace de stockage distant contenant votre site web.

L'espace de stockage distant contient le fichier **index.php** dont le contenu produit l'affichage ci-dessus.

Pour mettre votre site en ligne :

- Commencez par supprimer le fichier **index.php** (ou renommez-le en **_index.php**).
- Déplacez-vous dans l'arborescence de votre machine locale, jusque dans le dossier de votre site.
- Sélectionnez les fichiers et dossiers de votre site, puis cliquez sur le bouton **Envoyer**. Les fichiers doivent apparaître dans la fenêtre de droite.

Vérifiez la mise en ligne de votre site sur l'URL : **prénomnom.iutmmmiweb03.uha.fr**
