

AC14.02 : Produire des pages Web fluides avec un balisage sémantique

Pour offrir une bonne expérience utilisateur, il est essentiel de créer des pages web fluides et bien structurées. Cela passe par un balisage sémantique avec des balises comme <header>, <h1>, <p>, et <footer> pour organiser le contenu de manière logique.

Une navigation claire et une bonne hiérarchie facilitent l'exploration du site. Une tabulation soignée et des commentaires utiles rendent le code plus lisible.

1. Emploi des bonnes balises

Justification :

Il est essentiel d'utiliser les balises HTML appropriées comme <header>, <h1>, <p>, <footer>, pour améliorer la lisibilité du code et l'accessibilité de la page. Ces balises permettent d'organiser et décrire le contenu de manière sémantique. Elles améliorent la lisibilité du code, mais aussi l'accessibilité et l'optimisation SEO.

Exemple :

```
<> index.html > ...
1  <!DOCTYPE html>
2  <html lang="fr">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8  </head>
9
10 <body>
11     <header>
12         <div>Menu</div>
13     </header>
14     <h1>Titre</h1>
15     <h2>Sous titre</h2>
16     <p>blabla</p>
17     <h3>Sous titre</h3>
18     <p>blabla</p>
19     <footer>Réseaux sociaux</footer>
20 </body>
21
22 </html>
```

<header> : La balise définit l'en-tête du site.

<h1>, <h2>, <h3> : Les balises pour des titres.

<p> : La balise est utilisé pour un paragraphe.

<footer> : La balise définit le pied de page du site.

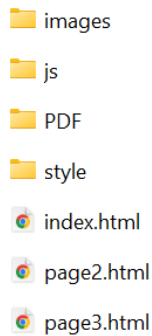
2. Bonne arborescence

Justifications :

Une bonne organisation des dossiers facilite la gestion du projet en séparant les fichiers par catégories (JS, styles, images). Cela rend le projet plus facile à maintenir et à développer.

De plus, l'utilisation des bonnes balises HTML (<header>, <h1>, <p>, <footer>, etc.) et une bonne hiérarchie du contenu assurent une navigation facile, une meilleure accessibilité et un meilleur référencement SEO.

Exemples :



```
index.html > ...
1  <!DOCTYPE html>
2  <html lang="fr">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8  </head>
9
10 <body>
11     <header>
12         <div>Menu</div>
13     </header>
14     <h1>Titre</h1>
15     <h2>Sous titre</h2>
16     <p>blabla</p>
17     <h3>Sous titre</h3>
18     <p>blabla</p>
19     <footer>Réseaux sociaux</footer>
20 </body>
21
22 </html>
```

Les titres sont par exemple organisés, il n’y a pas deux `<h1>` ou d’abord un `<h3>` puis un `<h2>`, ce qui montre une bonne arborescence.

3. Ajout de commentaires

Justification :

La tabulation et l’ajout de commentaire rendent le code plus lisible. Chaque bloc de code doit être tabulé pour montrer sa hiérarchie, et les sections doivent être commentées pour expliquer leur fonction. Un code bien structuré et commenté est plus facile à comprendre, que ce soit pour un autre développeur ou pour soi-même.

Exemple :

```
index.html > html > body > div.ensemble > div.gauche
1  <!DOCTYPE html>
2  <html lang="fr">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>SAE 105</title>
8  </head>
9
10 <!-- Lien vers la feuille de style CSS -->
11 <link rel="stylesheet" href="style/style.css">
12
13 <body>
14
15     <!-- En-tête du site avec des liens de navigation -->
16     <header>
17         <a href="index.html" class="acc">Accueil</a> <!-- Lien vers la page d'accueil -->
18         <a href="page2.html" class="som">Sommaire</a> <!-- Lien vers la page du sommaire -->
19     </header>
```

J’ai tabulé mon code en faisant « shit » + « alt » + « f ». L’ajout de commentaire permet de mieux comprendre le code et facilite la recherche d’une ligne de code.