PARLONS DE LA SEMANTIQUE EN HTML

1. Introduction a la sémantique

Quand on code en html on utilise de fois des éléments sémantique et des éléments non sémantique.

Pour utiliser le HTML sémantique on sélectionne les éléments HTML basé sur leur signification mais pas comme ils sont représenté. L’élément div et span sont non sémantique car ils ne donnent aucun contexte au contenu qui se trouve à l’intérieur.

Par exemple au lieu d’utiliser div to contain l’entête du body ça serait mieux d’utiliser l’élément header qui est utilisé pour l’entête.

Pour utiliser le HTML sémantique ?

* Accessibilité : le html sémantique rend les pages accessibles aux mobile ainsi qu’aux personnes handicapées. En effet le lecteur d’écran et les navigateurs sont capable de mieux interpréter le code.
* SEO : le sémantique html améliore le référencement du site web ou l’optimisation de moteur de recherche qui consiste à augmenter le nombre de personnes qui visitent votre page web.
* Easy to understand : le html sémantique facilite la lecture du code source du site web pour les autres développeurs.

Exemple de comparaison : comparons html sémantique avec les allées dans un magasin, plus il y a des insignes plus la recherche des articles est facile et plus il y en a plus ça devient très compliqué car on ne sait pas dans quelle allée on va trouver tel article.

1. Les Elements
   1. Header et Nav

**Le header** contient généralement les liens de navigation ou pour le contenu d’introduction contenant les titres h1 à h6

Sémantique code en utilisant <header>

<header>  
  <h1>  
     Everything you need to know about pizza!  
  </h1>  
</header>

Le même code ci haut peut être compare au code ici en utilisant cette fois la balise <div> mais en donnant quand les ID pour au moins donner du sens au contenu se trouvant entre le div

<div id=”header”>  
  <h1>  
     Everything you need to know about pizza!  
  </h1>  
</div>

**Note : en utilisant l’élément header ou la balise <header> notre code devient de plus en plus facile a lire.**

**Le nav** est utilisé pour définir un block des liens de navigations comme le menu ou la table de matière. Il est important de noter que nav peut être utiliser a l’intérieur de header

<header>   
  <nav>  
    <ul>  
      <li><a href="#home">Home</a></li>  
      <li><a href="#about">About</a></li>        
    </ul>  
  </nav>  
</header>

En utilisant nav ça devient plus facile de comprendre le code non seulement par nous-même aussi d’autre développeur ; les navigateurs et le lecteur d’écran

* 1. Main and Footer
     1. Main

L’élément <main> est utilisé pour encapsuler le contenu dominant de notre page web. Cet élément est différent de <footer> et de <nav> d’une page web parce que ces éléments ne contiennent pas le contenu principal du site.

En utilisant <main> par opposition a un élément <div>, le lecteur d’écran, le navigateur web seront capable de bien identifier que le contenu a l’intérieur c’est le contenu principal ou essentiel du site.

A quoi ressemble <main> lorsqu’il est incorporé dans notre code ?

<main>  
  <header>  
    <h1>Types of Sports</h1>  
  </header>  
  <article>  
    <h3>Baseball</h3>  
    <p>  
      The first game of baseball was played in Cooperstown, New York in the summer of 1839.  
    </p>  
  </article>  
</main>

Comme nous le voyons, <main> contient l’élément <article> et <header> avec des éléments enfant qui contiennent les informations les plus importantes liée à la page.

* + 1. Le footer

Le contenu en bas de la page avec les informations sur le sujet est appelé <footer> et ça contient les informations tels que :

* Information de contact
* Le copyrignt information
* Les conditions d’utilisations
* Le site Map
* Les références de liens au top de la page

<footer>  
  <p>Email me at Codey@Codecademy.com</p>  
</footer>

La balise <footer> est séparé du <main> et se trouve au bas de page

* 1. Article et Section
     1. Section

Une section définit une zone du document ayant le même thème. Donc ça peut contenir par exemple un contenu ayant même thème tel que les articles sur le cricket, peuvent figurer dans une seule section>.

La page d’accueil d’un site peut divisée en sections, pour l’introduction, les actualités et les informations de contact.

<section>  
  <h2>Fun Facts About Cricket</h2>   
</section>

Dans le code ci haut nous avions créé une section et nous avons encapsulé un élément h2 dans la section

* + 1. Articles

L’élément article contient du contenu qui a du sens en soi, <article> peut contenir du contenu tel que des articles, des blogs, des commentaires, des magazines.

<section>  
  <h2>Fun Facts About Cricket</h2>  
  <article>  
    <p>A single match of cricket can last up to 5 days.</p>  
  </article>  
</section>

Dans le code ci-dessus, l’élément <article> contenant un fait sur le cricket a été place à l’intérieur de l’élément section. Il est important de savoir que l’élément <section> peut être aussi placé dans l’élément <article> selon le contexte.

* 1. Aside

L’élément <aside> est utilisé pour marquer des informations supplementaires qui peuvent ameliorer un autre élément mais qui ne sont pas necessaire pour comprendre le contenu principal.

Cet element peut etre utilisé avec d’autres elements tels <articles> ou <section>

Certains cas d’utilisations :

* Bibliographies
* Note de fin
* Commentaire
* De citations

<article>  
  <p>The first World Series was played between Pittsburgh and Boston in 1903 and was a nine-game series.</p>  
</article>  
<aside>  
  <p>  
   Babe Ruth once stated, “Heroes get remembered, but legends never die.”   
  </p>  
</aside>

Comme on peut le voir, le contenu dans article constitue les informations importantes, pendant ce temps les informations contenues dans <aside> améliore les informations dans <article> mais pas nécessaire pour le comprendre.

* 1. Figure and Figcaption
     1. Figure

<figure> est un élément utilisé pour encapsulé un media tel qu’une image ; une illustration, un diagramme, un extrait de code qui est référencé dans le flux principal du document

<figure>  
  <img src="overwatch.jpg">  
</figure>

* + 1. Figcaption

<figcaption> est un élément utilisé pour décrire le media dans la balise <figure>. Habituellement <figcaption> ira à l’intérieur de <figure>.

<figure>  
  <img src="overwatch.jpg">  
  <figcaption>This picture shows characters from Overwatch.</figcaption>  
</figure>

Dans le code ci-dessus, nous avons ajouté une <figcaption> dans l’élément <figure> pour décrire l’image de l’exemple précèdent.

* 1. Audio et Attribut

L’élément audio est utilisé pour intégrer du contenu audio dans un document. Comme <video> et <audio> utilise src pour lier la source audio.

<audio>  
  <source src="iAmAnAudioFile.mp3" type="audio/mp3">  
</audio>

L’attribut type permet d’identifier, le type d’audio que nous utilisons, et ça permet aussi navigateur d’identifier le type plus facilement et a déterminer si ce type de fichier audio est pris en charge par le navigateur.

Nous avons lié le fichier audio dans le navigateur, mais nous devons maintenant lui donner des contrôles et c’est là qu’interviennent les attributs. Les attributs fournissent des informations supplémentaires sur un élément.

Il existe plusieurs attributs pour le fichier audio mais l’on va juste se concentrer sur le controls et src

* Controls : affiche automatiquement les commandes audio dans le navigateur tels que la lecture et la sourdine.
* Src : spécifie une url du fichier audio.
  1. Vidéo and Embed

La balise <video> : montre clairement que le développeur a voulu affiché une vidéo à l’utilisateur.

Certains attributs pouvant modifier la lecture d’une vidéo incluent :

* Controls : une fois ajoutée, un bouton de lecture et pause sera ajouté avec le contrôle du volume et une option plein écran.
* autoplay: l’attribut qui entrain la lecture automatique d’une vidéo dès que la page est chargée
* loop : cet attribut entraine la lecture continue de la vidéo en boucle.