

# Sémantique avancée

Dans le cours de perfectionnement, nous avons décrit la structure d'une page standard. Cela nous a permis d'aborder les balises structurantes servant à délimiter les zones les plus importantes de la page. Le propos de cette leçon est d'aller un peu plus loin dans les concepts et les balises étudiées. On verra que la problématique sémantique peut parfois s'avérer fort complexe. Naturellement, il ne faut pas penser qu'une seule solution est possible. Il existe, dans de nombreux cas, des choix de balises divergents. Cela n'est pas en soi un problème tant que les options choisies restent cohérentes.

## I. Le concept modulaire

Le concept de module est l'un des plus importants apportés par le HTML5. En réalité, celui-ci était déjà en gestation dans les précédentes versions du langage. Les pages bien conçues fonctionnent en réalité par modules, petits blocs que l'on peut ajouter ou supprimer sur une page. Mais il est pleinement assumé par la dernière version du langage HTML.

On peut reprendre où l'on s'était arrêté et affiner le travail afin de venir travailler désormais à l'intérieur de chaque bloc et se rapprocher de mises en page plus complexes. Pour rappel, nous en étions arrivés ici :

```
<div id="wrapper">
  <header>
    <p>header</p>
  </header>
  <nav>
    <p>menu</p>
  </nav>
  <main>
    <article>article
      <section>section 1</section>
      <section>section 2</section>
      <section>section 3</section>
    </article>
    <aside>
    </aside>
  </main>
  <footer>
    <p>footer</p>
  </footer>
</div>
```

Fig. 1

Avec le rendu suivant :

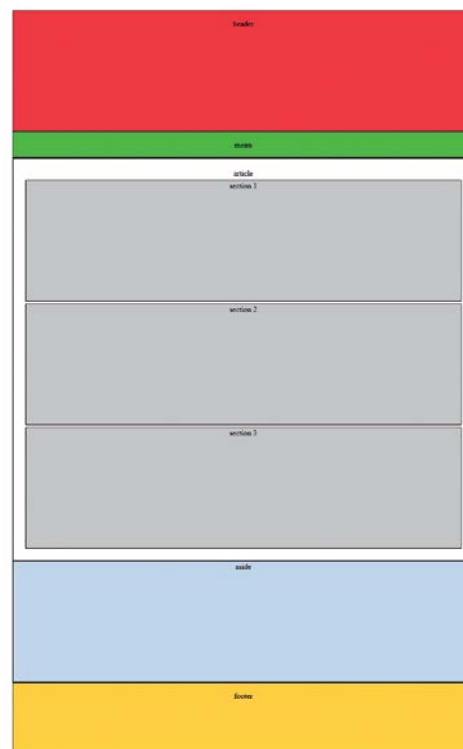


Fig. 2

Bien sûr, il faut tenir compte du fait que nous n'avons pas encore vu le positionnement. Mais ce qui nous intéresse est purement la structure du HTML. Jusqu'ici on n'utilisait qu'un seul **h1** par page, la justification en étant qu'une page possède un titre principal.

Auparavant, et comme nous l'avons appris jusque-là, pour organiser correctement une page il fallait utiliser des balises de titres (typiquement pour l'élément `<article>`) en prenant comme référence la balise de titre principale (**h1**) puis ajouter une balise de niveau inférieur au titre de la page pour aboutir à quelque chose comme :

```

<main>
  <article>
    <h1>titre article</h1>
    <section>
      <h2>titre section 1</h2>
      <p>paragrahe</p>
    </section>
    <section>
      <h2>titre section 2</h2>
      <p>paragrahe</p>
    </section>
    <section>
      <h2>titre section 3</h2>
      <p>paragrahe</p>
    </section>
  </article>
  <aside>
  </aside>
</main>

```

Fig.3 Organiser une page avec les balises titre © Skill and You

Pourtant, depuis HTML5, on n'est plus obligé de se cantonner à un titre par page, on doit travailler en « modules ». C'est-à-dire conserver au mieux la même structure de blocs de code indépendant HTML que l'on retrouve d'une page à l'autre. Par exemple, on a une série d'articles qui remontent en page d'accueil, et sur lesquels on a repris le titre et le sous-titre. Pourquoi ne pas reprendre les traits les plus marquants de la même structure que l'on trouve sur la page dédiée correspondant à l'article. Après tout, nous devons séparer le contenu de la mise en forme et pour un même contenu il ne serait pas anormal que la structure soit identique. Ainsi, à chaque module, au lieu de repartir du **h1** de la page, on met les compteurs à zéro.

Par exemple :

```

<main>
  <article>
    <h2>titre article</h2>
    <section>
      <h3>titre section 1</h3>
      <p>paragrahe</p>
    </section>
    <section>
      <h3>titre section 2</h3>
      <p>paragrahe</p>
    </section>
    <section>
      <h3>titre section 3</h3>
      <p>paragrahe</p>
    </section>
  </article>
  <aside>
  </aside>
</main>

```

Fig.4 Organiser une page en module © Skill and You

Évidemment, le modulaire veut dire que nous avons plusieurs modules ou blocs dans la même page. Ainsi, si une barre latérale donnait la possibilité de consulter des articles sur le même sujet, on pourrait avoir :

```

<main>
  <article>
    <h1>titre article</h1>
    <section>
      <h2>titre section 1</h2>
      <p>paragrahe</p>
    </section>
    <section>
      <h2>titre section 2</h2>
      <p>paragrahe</p>
    </section>
    <section>
      <h2>titre section 3</h2>
      <p>paragrahe</p>
    </section>
  </article>
  <aside>
    <h2>sur le même sujet</h2>
    <article>
      <h1>titre article</h1>
      <p>présentation</p>
    </article>
    <article>
      <h1>titre article</h1>
      <p>présentation</p>
    </article>
    <article>
      <h1>titre article</h1>
      <p>présentation</p>
    </article>
  </aside>
</main>

```

Fig.5 Une page html avec plusieurs modules © Skill and You

Attention, vous n'êtes pas obligé d'utiliser plusieurs **h1** par page, l'écriture précédente en **h2** est tout à fait recevable.

Mais nous n'avons pas fini avec les capacités modulaires du langage. Le HTML5, se voulant plus sémantique encore que les versions précédentes du HTML, a introduit des balises spécifiques comme **<header>** ou **<footer>**. Précédemment, nous avons étudié leur utilisation au niveau macro, c'est-à-dire sur l'ensemble de la page. Il s'agit désormais, et dans la même logique que notre problématique concernant la titrairie, de travailler au niveau micro, c'est-à-dire sur un bloc défini.

Pour cela, concentrons-nous sur la principale balise **<article>** de notre page. On va donc, en même temps que le titre en **h1**, indiquer le nom de l'auteur ainsi que la date. Pour ce faire, on va les encadrer dans une balise **<header>** puisqu'ils composent l'en-tête de notre article. Nous allons également nous servir de la balise **time** utilisée pour les dates : **<time>** qui doit s'employer avec un argument, ici nous choisissons **datetime** que l'on écrira au format YYYY-MM-DD (ex. : 2018-12-25).

```
<article>
  <header>
    <h1>titre article</h1>
    <div class="metas">
      <span class="auteur">auteur article</span>
      <time datetime="2023-04-25">25 avril 2023</time>
    </div>
  </header>
  <section>
    <h2>titre section 1</h2>
    <p>paragraphe</p>
  </section>
  <section>
    <h2>titre section 2</h2>
    <p>paragraphe</p>
  </section>
  <section>
    <h2>titre section 3</h2>
    <p>paragraphe</p>
  </section>
</article>
```

Fig. 6 Exemple de balise time pour les dates © Skill and You

Nous pouvons vouloir faire apparaître une série de mots-clés associés à l'article à la fin de celui-ci. Il ne serait pas pertinent d'ajouter une balise **<section>** puisqu'il ne s'agit pas à proprement parler d'une

partie de l'article, mais bien de l'équivalent d'un pied de page. Heureusement la balise **<footer>**, tout comme la balise **<header>**, peut être utilisée à l'intérieur d'un « module ». Dans ce footer, on va réserver une zone de mots-clés et de catégories dans une **div**.

```
<article>
  <header>
    <h1>titre article</h1>
    <div class="metas">
      <span class="auteur">auteur article</span>
      <time datetime="2023-04-25">25 avril 2023</time>
    </div>
  </header>
  <section>
    <h2>titre section 1</h2>
    <p>paragraphe</p>
  </section>
  <section>
    <h2>titre section 2</h2>
    <p>paragraphe</p>
  </section>
  <section>
    <h2>titre section 3</h2>
    <p>paragraphe</p>
  </section>
  <footer>
    <div class="container-tags">
      <span class="tags">Chien, poil</span>
      <span class="categories">Animaux, soins</span>
    </div>
  </footer>
</article>
```

Fig. 7 Balises header et footer dans un article © Skill and You

Nous omettons de fournir des liens pour chacun des tags et catégories indiqués, mais c'est uniquement pour éviter de surcharger le code.

Maintenant notre page est composée d'un article principal et de liens (même si nous ne les avons pas indiqués ici) vers des articles se rapportant à celui-ci. Sous cet article on pourrait ajouter une seconde zone destinée à accueillir divers « modules ». Pour signifier le titre de rubrique, nous ne voulons pas employer de balise titre car nous ne souhaitons pas donner trop d'importance à cet élément. Heureusement, la balise **<header>** nous permet de réaliser ce qui consiste bien en une présentation de notre zone. Comme nos trois modules sont des contenus indépendants, nous allons les encadrer d'une balise **<article>**.

De plus, nous allons ajouter des images par l'intermédiaire de la balise **<figure>** qui définit une zone pouvant contenir une ou plusieurs images.

```

<section class="les-plus">
  <header>les plus</header>
  <article>
    <h1>Zoom</h1>
    <figure>
      
    </figure>
    <p>Le zoom du mois</p>
  </article>
  <article>
    <h1>Bonnes affaires</h1>
    <figure>
      
    </figure>
    <p>Dénicher les bonnes opportunités</p>
  </article>
  <article>
    <h1>Infographie</h1>
    <figure>
      
    </figure>
    <p>La productivité des séniors européens</p>
  </article>
</section>

```

Fig.8 La balise figure pour illustrer un article à l'aide d'une image

### Option divergente

On pourrait rajouter une couche, en considérant que ces éléments font partie d'une liste, on ferait donc figurer chaque `<article>` à l'intérieur d'un `<li>`. Ceux-ci seront composés d'une balise de titre, d'une image comprise dans la balise figure, et d'un texte dans la balise `<p>`.

### `<figure>` et `<p>`

Pourquoi avoir choisi la balise `p` pour le texte et non `<figcaption>` (pour rappel `<figcaption>` est utilisée dans la balise `<figure>` pour la légende de l'image) ? La réponse est d'ordre sémantique ; en effet, le texte est lié à l'article, c'est-à-dire à la balise `<article>`. Ici, ce n'est donc pas un texte qui légende l'image, c'est l'image qui illustre l'article.

## II. Classes et sémantique

Contrairement à ce que l'on pourrait penser, l'apparition de balises sémantiques ne milite pas pour abandonner ou rendre obsolète le travail des classes et/ou des ID. Il existe d'ailleurs des classes dites sémantiques : les microformats par exemple. Ainsi pour distinguer la principale balise `<article>` de celles qui sont dans le **aside**, nous pourrions ajouter une classe. En effet, nous sommes sur la page de l'article, il n'y aura donc qu'un article correspondant au contenu principal de la page :

```

<article class="main-article">
  <header>
    <h1>titre article</h1>
    <div class="metas">
      <span class="auteur">auteur article</span>
      <time datetime="2023-04-25">25 avril 2023</time>
    </div>
  </header>

```

Fig.9 Distinguer un article principal avec une classe © Skill and You

Pour les autres occurrences, on peut utiliser des classes pour chaque **article**. Par exemple, au sein de la balise **aside**, nous allons utiliser « article-aside » et « article-more » pour la <section> « les-plus ». Ce qui nous donnerait pour **aside** :

```

<aside>
  <h2>sur le même sujet</h2>
  <article class="article-aside">
    <h1>titre article</h1>
    <p>présentation</p>
  </article>
  <article class="article-aside">
    <h1>titre article</h1>
    <p>présentation</p>
  </article>
  <article class="article-aside">
    <h1>titre article</h1>
    <p>présentation</p>
  </article>
</aside>

```

Fig.10 Distinguer un article aside avec une classe © Skill and You

Pour « les-plus » :

```

<section class="les-plus">
  <header>les plus</header>
  <article class="article-more">
    <h1>Zoom</h1>
    <figure>
      
    </figure>
    <p>Le zoom du mois</p>
  </article>
  <article class="article-more">
    <h1>Bonnes affaires</h1>
    <figure>
      
    </figure>
    <p>Dénicher les bonnes opportunités</p>
  </article>
  <article class="article-more">
    <h1>Infographie</h1>
    <figure>
      
    </figure>
    <p>La productivité des seniors européens</p>
  </article>
</section>

```

Fig.11 Distinguer les micro formats avec une classe © Skill and You

Ce qui se joue ici, outre que cela permet de sécuriser le CSS, c'est une autre manière d'aborder la sémantique, où les classes et identifiants CSS vont nous servir à décrire, cette fois-ci de manière plus personnalisée, les contenus de nos pages. Après, tout dépend de la stratégie de nommage de nos règles CSS.

Autre exemple, mettons que nous hésitons entre différentes balises et que nous avons bien pensé à donner une classe sur les différents éléments de mon module :

```
<div class="editorial">
  <h1 class="title">Editorial</h1>
  <h2 class="abstract">description</h2>
  <a href="url.html" class="read-more">Lire la suite</a>
</div>
```

Fig. 12 Les classes servent aussi à décrire le contenu © Skill and You

## CSS

```
.editorial {
  width: 33%;
  background-color: #ddd;
}
.title {
  font-size: 1.8em;
  color: #444;
}
.abstract {
  font-size: 1.4em;
  color: #a32;
}
```

Fig. 13 Les identifiants CSS servent aussi à décrire le contenu © Skill and You

Il est possible de modifier le HTML sans avoir à se préoccuper du CSS (ici `<article>` pour `<div>` et `<p>` pour `<h2>`).

```
<article class="editorial">
  <h1 class="title">Editorial</h1>
  <p class="abstract">description</p>
  <a href="url.html" class="read-more">Lire la suite</a>
</article>
```

Fig. 14 Modifier le html est sans conséquence pour le CSS © Skill and You

La règle CSS cible `.editorial` et ce quelle que soit la balise native HTML `<div>`, `<article>`, `<section>`, du moment qu'elle porte le nom de la classe demandée.

À partir de ces quelques exemples, il s'agit de comprendre que les noms donnés aux classes sont loin d'être neutres. Une approche intelligente permet de rendre les fichiers HTML et CSS plus lisibles car il est primordial de penser en matière de souplesse et de maintenabilité du code.

## III. États finaux : les alternatives

Récapitulons l'ensemble du code HTML sémantique :

```
<div id="wrapper">
  <header>
    <p>header</p>
  </header>
  <nav>
    <p>menu</p>
  </nav>
  <main>
    <article class="main-article">
      <header>
        <h1>titre article</h1>
        <div class="metas">
          <span class="auteur">auteur article</span>
          <time datetime="2023-04-25">25 avril 2023</time>
        </div>
      </header>
      <section>
        <h2>titre section 1</h2>
        <p>paragraphe</p>
      </section>
      <section>
        <h2>titre section 2</h2>
        <p>paragraphe</p>
      </section>
      <section>
        <h2>titre section 3</h2>
        <p>paragraphe</p>
      </section>
      <footer>
        <div class="container-tags">
          <span class="tags">Chien, poil</span>
          <span class="categories">Animaux, soins</span>
        </div>
      </footer>
    </article>
  </main>
</div>
```

Fig. 15 Récapitulatif de la sémantique html pour l'article principal © Skill and You

```
<aside>
  <h2>sur le même sujet</h2>
  <article class="article-aside">
    <h1>titre article</h1>
    <p>présentation</p>
  </article>
  <article class="article-aside">
    <h1>titre article</h1>
    <p>présentation</p>
  </article>
  <article class="article-aside">
    <h1>titre article</h1>
    <p>présentation</p>
  </article>
</aside>
```

Fig. 16 Récapitulatif de la sémantique html pour l'article aside © Skill and You

```

<section class="les-plus">
  <header>les plus</header>
  <article class="article-more">
    <h1>Zoom</h1>
    <figure>
      
    </figure>
    <p>Le zoom du mois</p>
  </article>
  <article class="article-more">
    <h1>Bonnes affaires</h1>
    <figure>
      
    </figure>
    <p>Dénicher les bonnes opportunités</p>
  </article>
  <article class="article-more">
    <h1>Infographie</h1>
    <figure>
      
    </figure>
    <p>La productivité des séniors européens</p>
  </article>
</section>
</main>
<footer>
  <p>footer</p>
</footer>
</div>
</body>
</html>

```

Fig. 17 Récapitulatif de la sémantique html pour l'article en plus et avec visuel © Skill and You

Comme dit plus haut, diverses solutions sémantiques peuvent être envisagées, les choix dépendent des contenus de la page et également des objectifs attendus ainsi que de la subjectivité du concepteur de la page.

Nous vous proposons ici une alternative cohérente avec certaines des solutions envisagées plus haut.

```

<div id="wrapper">
  <header>
    <h1>header</h1>
  </header>
  <nav>
    <p>menu</p>
  </nav>
  <div id="contenu">
    <article id="main-article">
      <header>
        <h2>titre-article</h2>
        <p class="metas">
          <span class="auteur">auteur article</span>
          <time datetime="2023-04-25">25 avril 2023</time>
        </p>
      </header>
      <section>
        <h3>titre section 1</h3>
        <h4>paragraphe</h4>
      </section>
      <section>
        <h3>titre section 2</h3>
        <h4>paragraphe</h4>
      </section>
      <section>
        <h3>titre section 3</h3>
        <h4>paragraphe</h4>
      </section>
      <footer id="footer-tags">
        <span class="tag">Chien, Poils</span>
        <span class="categories">Animaux, Soins</span>
      </footer>
    </article>
  </div>
</div>

```

Fig. 18 Solution alternative de la sémantique partie 1 © Skill and You

```

<section class="les-plus">
  <h2>les plus</h2>
  <article class="article-more">
    <h3>Zoom</h3>
    <figure>
      
    </figure>
    <p>Le zoom du mois</p>
  </article>
  <article class="article-more">
    <h3>Bonnes affaires</h3>
    <figure>
      
    </figure>
    <p>Dénicher les bonnes opportunités</p>
  </article>
  <article class="article-more">
    <h3>Infographie</h3>
    <figure>
      
    </figure>
    <p>La productivité des seniors européens</p>
  </article>
</section>
</main>
<footer>
  <p>footer</p>
</footer>
</div>
</body>
</html>

```

Fig. 19 Solution alternative de la sémantique partie 2 © Skill and You

```

<aside>
  <h3>sur le même sujet</h3>
  <ul>
    <li>
      <article class="article-aside">
        <h4>titre article</h4>
        <p>présentation</p>
      </article>
    </li>
    <li>
      <article class="article-aside">
        <h4>titre article</h4>
        <p>présentation</p>
      </article>
    </li>
    <li>
      <article class="article-aside">
        <h4>titre article</h4>
        <p>présentation</p>
      </article>
    </li>
  </ul>
</aside>

```

Fig. 20 Liste ul>li dans le aside © Skill and You

Vous remarquez qu'on préfère ne pas reprendre la hiérarchie des balises de titre à chaque module. Vous remarquez également que l'on utilise une liste **ul>li** dans le **aside**.

## IV. Conclusion

La sémantique est l'un des piliers d'un code propre, maintenable et efficace. Nous avons passé en revue quelques critères permettant de s'y retrouver dans les balises sémantiques, à savoir : utiliser la balise la plus proche du contenu qu'elle encadre. Nous avons vu le nommage des CSS afin d'apporter une surcouche sémantique, qui, si elle n'est pas prise en compte par les moteurs de recherche, nous permet de nous y retrouver dans le code.