

Chapitre 3 : Les éléments de structure HTML5

A. Les éléments de structure en HTML4.	58
B. Les éléments de structure en HTML5.	60
C. L'attribut sémantique role.	63
D. Exemples de structure HTML5.	64
E. Les techniques de mise en page.	66

A. Les éléments de structure en HTML4

1. Les boîtes <div>

Avec le HTML4, le principal élément de structure des pages est la fameuse « boîte <div> ». L'élément <div> permet de créer des zones d'affichage rectangulaires. Chacune de ces zones, chacune de ces boîtes <div> peut être identifiée de manière unique avec un identifiant et être mise en forme avec des styles CSS. Cet identifiant unique est déterminé avec l'attribut `id`.

Exemple :

```
<div id="bandeauHaut">  
  ...  
</div>
```

Vous pouvez aussi utiliser des classes, si la mise en forme n'est pas unique mais est répétée dans la page.

Exemple :

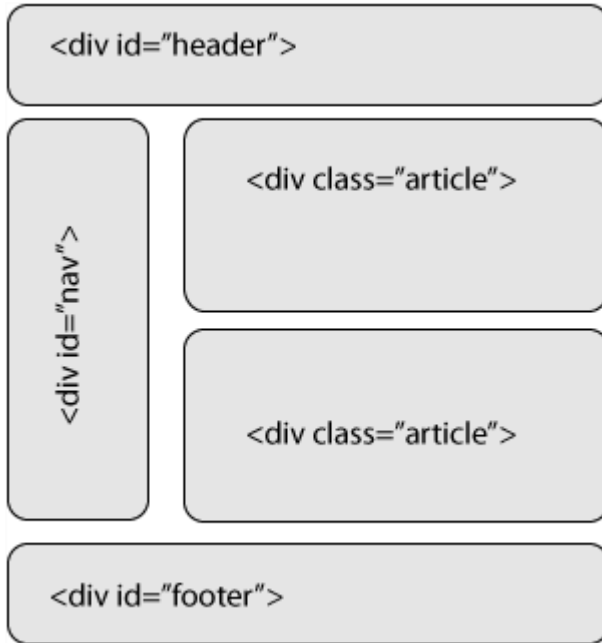
```
<div class="article">  
  ...  
</div>
```

Ensuite, une fois que la structure est déterminée, la mise en page s'effectue avec des styles CSS, avec des sélecteurs d'identification qui reprennent le nom de l'identifiant ou de la classe.

Exemple :

```
#bandeauHaut {  
  ...  
}  
.article{  
  ...  
}
```

Voici un exemple d'une structure de page de type blog avec des boîtes `<div>` :



2. La « divite »

Avec le HTML4 et les CSS 2.1, la structure des pages, quand elle est un peu complexe, peut nécessiter un nombre très important de boîtes `<div>`. Vous pouvez avoir des pages avec des dizaines et des dizaines de boîtes `<div>`. Si vous tombez dans cet excès, vous êtes alors atteint de la maladie de la « divite », soit un abus de boîtes `<div>` !

3. Un contenu non sémantique

Avec toutes ces boîtes `<div>` se pose un autre problème : celui de la sémantique des conteneurs. En effet, chaque boîte `<div>` se différencie par son identifiant unique, donné par le concepteur, selon son « humeur » et ses « envies ».

Ce qui fait que les boîtes `<div>` ne sont pas sémantiques : le contenu attendu n'est en rien défini par un quelconque paramètre. Une boîte `<div>` identifiée avec `id="gauche"` ne prédéfinit en rien son contenu. Nous pourrions y trouver une barre de navigation comme des informations légales ou tout autre contenu.

Or, l'évolution du HTML va vers plus de sémantique.

B. Les éléments de structure en HTML5

1. Les nouveaux éléments de structure

L'avènement du HTML a apporté de nouveaux éléments de structure sémantique. Ces nouveaux éléments ont été définis et nommés suite à une longue analyse du nom des boîtes `<div>`.

2. L'élément `<header>`

Le nouvel élément `<header>` permet d'insérer une zone d'affichage pour les en-têtes. Cet en-tête peut être défini pour la page ou pour une autre zone d'affichage : article, barre latérale...

Il faut considérer cet élément comme utilisable à deux niveaux :

- au niveau de la page, c'est le classique en-tête de page, souvent placé en haut de l'écran, avec un logo, un slogan, une barre de navigation principale...
- au niveau des contenus, cela permet d'avoir une zone d'introduction au contenu qui va suivre.

Voilà la définition du W3C pour l'élément `<header>` : « *The header element represents a group of introductory or navigational aids. A header element is intended to usually contain the section's heading (an h1–h6 element or an hgroup element), but this is not required. The header element can also be used to wrap a section's table of contents, a search form, or any relevant logos.* »

Notez bien que l'élément `<header>` n'est pas obligatoire dans la structure des pages.

3. L'élément `<footer>`

Le nouvel élément `<footer>` permet d'insérer une zone d'affichage pour les pieds de page. Nous retrouvons la même sémantique que pour les en-têtes. Ces pieds de page peuvent être définis pour la page ou pour une autre zone d'affichage, ou pour un article.

Voilà la définition du W3C : « *The footer element represents a footer for its nearest ancestor sectioning content or sectioning root element. A footer typically contains information about its section such as who wrote it, links to related documents, copyright data, and the like.* »

Nous retrouvons un usage semblable à celui du `<header>`, mais cette fois pour des pieds de page. Nous ne devons donc pas prendre « au pied de la lettre » la traduction littérale « pied de page ». C'est plutôt un pied de zone d'affichage, la zone d'affichage pouvant être une page, une section, un article...

4. L'élément `<nav>`

L'élément `<nav>`, comme son nom le laisse supposer, est dédié à l'affichage d'une navigation avec des liens hypertextes. Mais attention, ne vous sentez pas obligé de n'avoir qu'une seule zone de navigation par page ou de créer autant d'éléments `<nav>` que vous avez de navigations dans vos pages, du moment que chacun d'entre eux est identifié. L'élément `<nav>` est plutôt dédié à la navigation principale du site, à la création de liens entre les pages du site.

La définition du W3C est claire : « *The nav element represents a section of a page that links to other pages or to parts within the page: a section with navigation links. Note: Not all groups of links on a page need to be in a nav element – the element is primarily intended for sections that consist of major navigation blocks. In particular, it is common for footers to have a short list of links to various pages of a site, such as the terms of service, the home page, and a copyright page. The footer element alone is sufficient for such cases; while a nav element can be used in such cases, it is usually unnecessary.* »

5. L'élément <section>

L'élément <section> permet de regrouper des éléments partageant une même thématique. Cela permet de regrouper dans un même élément un contenu, avec son en-tête et son pied de page.

Voilà la définition du W3C : « *The section element represents a generic section of a document or application. A section, in this context, is a thematic grouping of content, typically with a heading.* »

6. L'élément <article>

L'élément <article> permet d'insérer un contenu autonome, car il peut être réutilisé ailleurs dans le site, sans que sa compréhension soit nulle.

Voici la définition du W3C : « *The article element represents a self-contained composition in a document, page, application, or site and that is, in principle, independently distributable or reusable, e.g. in syndication. This could be a forum post, a magazine or newspaper article, a blog entry, a user-submitted comment, an interactive widget or gadget, or any other independent item of content.* »

Un article peut parfaitement avoir un en-tête (<header>), un pied de page (<footer>) et des titres (<hx>).

7. L'élément <aside>

L'élément <aside> permet d'afficher un contenu lié au contenu auquel il est associé. Cela peut correspondre à des barres latérales (sidebar), à des zones de widgets, à des compléments sur des articles.

8. L'élément <main>

Dernier arrivé dans les brouillons du W3C, l'élément <main> permet de cibler le contenu principal de la page.

La définition du W3C l'indique bien : « *The main element represents the main content of the body of a document or application. The main content area consists of content that is directly related to or expands upon the central topic of a document or central functionality of an application.* »

Ainsi, une page pourra être parfaitement structurée avec un élément `<main>` pour le contenu de la page, avec un en-tête `<header>`, des contenus avec des éléments `<article>` et un pied de page `<footer>`.



Le W3C indique des restrictions pour l'utilisation de l'élément `<main>` :

- un seul élément `<main>` par page,
- il ne doit pas être utilisé à l'intérieur (élément descendant) des éléments `<article>`, `<aside>`, `<footer>`, `<header>` ou `<nav>`,
- il devrait être associé avec le rôle `main` (`<main role="main">`) pour une meilleure accessibilité des sites web.

9. Les boîtes `<div>`

Enfin pour conclure, ce n'est pas parce que vous utilisez du HTML5, que vous devez abolir l'utilisation des boîtes `<div>` ! Elles sont toujours utilisables et ont toujours leur utilité !