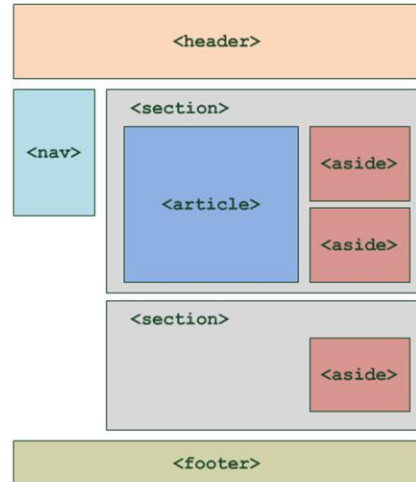


Nouvelle structuration

Les éléments traditionnels

structuration avec les nouvelles balises HTML5

Les éléments de section ([section](#), [article](#), [nav](#), [aside](#), [header](#), [footer](#)) segmentent des portions du document ou de l'application web.



En général, une page web est constituée d'un en-tête (tout en haut), de menus de navigation (en haut ou sur les côtés), de différentes sections au centre... et d'un pied de page (tout en bas).

Nous allons nous intéresser aux nouvelles balises HTML dédiées à la structuration du site. Ces balises ont été introduites par HTML5 et vont nous permettre de dire : « Ceci est mon en-tête », « Ceci est mon menu de navigation »...

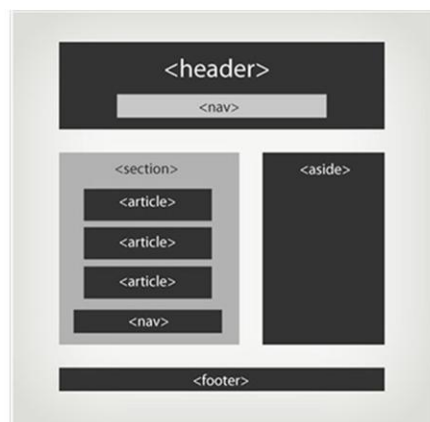
Notes :

Nouvelle structuration

Les éléments traditionnels

De nouveaux éléments sémantiques

Au-delà de la rénovation syntaxique, HTML5 introduit de nouveaux éléments inédits dotés de sens et apportant une alternative aux blocs génériques `<div>` et ``



Nom	Détails
<code><section></code>	Section générique regroupant un même sujet, une même fonctionnalité, de préférence avec un en-tête, ou bien section d'application web
<code><article></code>	Section de contenu indépendante, pouvant être extraite individuellement du document ou syndiquée (flux RSS ou équivalent), sans pénaliser sa compréhension
<code><nav></code>	Section possédant des liens de navigation principaux (au sein du document ou vers d'autres pages)
<code><aside></code>	Section dont le contenu est un complément par rapport à ce qui l'entoure, qui n'est pas forcément en lien direct avec le contenu mais qui peut apporter des informations supplémentaires.
<code><header></code>	Section d'introduction d'un article, d'une autre section ou du document entier (en-tête de page).
<code><footer></code>	Section de conclusion d'une section ou d'un article, voire du document entier (pied de page).

Nouvelle structuration

Les éléments traditionnels

Un Doctype simplifié

```
<!doctype html>
```

Ce doctype allégé a été mûrement testé et réfléchi : il permet toujours pour les anciens navigateurs de rester dans le mode d'interprétation conforme aux standards.

Elle vous est fidèle depuis que vos documents web sont conformes au W3C. Toujours en tête de vos pages, la *Déclaration de type de document* (DTD), appelée par le Doctype, annonce fièrement au navigateur quel est le langage et la grammaire que vous lui faites lire.

Voici à titre indicatif, deux Doctype couramment employés de nos jours, le HTML4.01 strict et le XHTML 1.0 transitionnel.

Doctype HTML 4.01 strict

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

Doctype XHTML 1.0 transitionnel

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

A l'ère naissante de HTML5, la syntaxe du Doctype a été remaniée de façon tellement optimisée qu'elle en devient largement plus facile à retenir.

Nouvelle structuration

L'élément header

<header>

L'élément **au moment où une non** représente le bloc d'en-tête d'une section ou d'une page. Il remplace avantageusement son homologue classique **<div id="header">**, mais ne doit pas forcément être considéré comme un élément unique dans le document : toute section est susceptible de disposer de son header

·
<header>

Ici tout le contenu de l'en-tête de document ou de section

</header>

La spécification HTML5 décrit d'éléments **header** comme un conteneur pour « un groupe d'outils d'introduction ou de navigation ». Un document peut comporter plusieurs éléments **header**, vous pouvez par exemple utiliser un élément **header** dans un élément **section**. La spécification décrit l'élément **section** comme regroupement de contenu thématique, comprenant généralement un en-tête.

<section>

<header>

<h1>Spécification</h1>

</header>

<p>Support de cours HTML5 – CSS3</p>

</section>

Un **header** apparaîtra normalement en haut d'un document ou d'une section, mais ce n'est pas obligatoire. Il est défini par son contenu plutôt que par sa position.

Nouvelle structuration

L'élément nav

<nav>

<nav> a pour fonction de regrouper les liens de navigation considérés comme majeurs ou jugés suffisamment pertinents.

```
<nav>
  <ul>
    <li><a href="index.html">Page d'accueil</a></li>
    <li><a href="contact.html">Contact</a></li>
  </ul>
</nav>
```

L'élément <nav> fait exactement ce qu'on attend de lui. Il contient des informations de navigation, généralement une liste de liens.

L'élément <nav> est conçu pour les informations de navigations principales. Une simple liste de liens ne justifie pas l'utilisation de l'élément <nav>. En revanche, les liens permettant de naviguer sur un site doivent presque obligatoirement utiliser l'élément nav.

Bien souvent, un élément nav apparaîtra dans un élément header. Cela paraît logique si l'on considère que l'élément header peut être utilisé pour les aides à la navigation.

Nouvelle structuration

L'élément section

<section>

Une `<section>` représente un bloc générique de contenu ayant la même thématique. Cela concerne les chapitres, les en-têtes et pieds de page, ou tout autre partie dans un document. L'élément `<section>` peut contenir des éléments de titre à h1 à h6 pour une meilleure définition de la structure du document.

```
<section>
  <h1>Articles</h1>

  <article>
    ...
  </article>
</section>
```

L'élément `section` est utilisé pour regrouper du contenu de manière thématique. Cela rappelle l'élément `div`, souvent utilisé comme un conteneur de contenu générique. La différence, c'est que `div` n'a aucune signification sémantique. L'élément `section`, en revanche, est utilisé explicitement pour regrouper du contenu apparenté.

Vous pourrez remplacer quelques-uns de vos éléments `div` par des éléments `section`, mais demandez-vous toujours si tout le contenu est apparenté.

`<section>`

`<header>`

`<h1>Spécification</h1>`

`</header>`

`<p>Support de cours HTML5 – CSS3</p>`

`</section>`

Nouvelle structuration

L'élément article

<article>

L'élément `<article>` désigne une portion du document potentiellement autonome dans le sens où elle pourrait être reprise ou réutilisée, comme un article de journal, de bloc ou de forum.

```
<article>
  <h1>Titre de l'article</h1>
  <p>Contenu de l'article</p>
</article>
```

L'élément `article` est un autre type de section spécialisée. On l'utilise pour le contenu apparenté autonome. Le problème étant de déterminer ce qui constitue un contenu autonome.

Si le contenu a toujours un sens dans ce contexte, `article` est probablement l'élément à utiliser. En fait, l'élément `article` est conçu spécifiquement pour la syndication.

<article>

```
<h1>Mon article</h1>
```

```
<p>Support de cours HTML5 – CSS3</p>
```

</article>

Nouvelle structuration

L'élément aside

<aside>

L'élément `<aside>` représente une portion de contenu contextuelle, directement ou indirectement liées aux éléments qui l'entourent, tel un bloc d'archives relatives au contenu précédent. Par extrapolation, cet élément désigne fréquemment les barres latérales classiques du document et peut remplacer l'ancien `<div id="sidebar">`.

```
<aside>
  <h4>Sources de l'article</h4>
  <ul>
    <li><a href="#">Lien 1</a></li>
    <li><a href="#">Lien 2</a></li>
    <li><a href="#">Lien 3</a></li>
  </ul>
</aside>
```

L'élément `aside` est une sorte d'encadré. Quand je dis encadré, ce n'est pas une question de position. Il ne suffit pas que le contenu apparaisse à gauche ou à droite du contenu principal pour utiliser l'élément `aside`. Une fois de plus, c'est le contenu, et non la position, qui est important.

L'élément `aside` doit être utilisé pour du contenu indirectement apparenté. Si vous avez un morceau de contenu que vous considérez comme différent du contenu principal, l'élément `aside` est probablement le bon conteneur. Demandez-vous si le contenu de l'élément `aside` peut-être retiré sans affecter le sens du contenu principal du document ou de la section.

Nouvelle structuration

L'élément footer

<footer>

L'élément `<footer>` regroupe le contenu du pied d'une section ou d'un document et est destiné à recueillir les informations concernant l'auteur, les mentions légales, etc. Tel `<header>`, l'élément `<footer>` peut apparaître à divers endroits du document.

```
<article>
...
  <footer>
    <p>Posté par Simon, le
      <time datetime="2012-02-02">2 février 2012</time>
    </p>
  </footer>
</article>
```

Comme l'élément `header`, `footer` semble décrire une position, mais comme pour `header`, ce n'est pas le cas. L'élément `footer` contient plutôt des informations au sujet de l'élément qui le contient: auteur, informations de copyright, lien vers des contenus apparentés, etc.

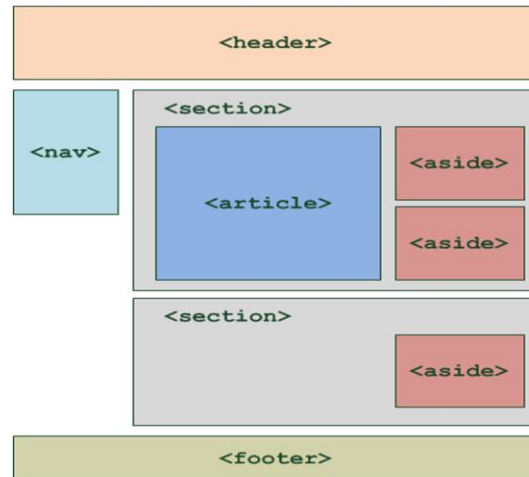
L'HTML5 nous permet également d'en placer à l'intérieur des sections.

```
<section>
  <header>
    <h1>Cours HCS</h1>
  </header>
  <p>HTML5 – CSS3</p>
  <footer>
    <p>Zack's Agency</p>
  </footer>
</section>
```

Nouvelle structuration

Modèle de contenu

Ce code peut vous aider à comprendre comment les balises doivent être agencées. Vous y reconnaissez un en-tête, un menu de navigation, un pied de page... Et, au centre, une section avec un article et un bloc aside donnant des informations sur l'auteur de l'article.



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="style.css" />
    <title>Cours HCS</title>
  </head>

  <body>
    <header>
      <h1>Cours Zack's Agency</h1>
      <h2>Nouveaux éléments HTML5</h2>
    </header>

    <nav>
      <ul>
        <li><a href="#">Accueil</a></li>
        <li><a href="#">Blog</a></li>
        <li><a href="#">CV</a></li>
      </ul>
    </nav>

    <section>
      <aside>
        <h1>À propos de l'auteur</h1>
        <p>C'est moi, Zozor ! Je suis né un 23 novembre 2005.</p>
      </aside>
      <article>
        <h1>Je suis un grand voyageur</h1>
        <p>Bla bla bla bla (texte de l'article)</p>
      </article>
    </section>

    <footer>
      <p>Copyright Zozor - Tous droits réservés<br/>
      <a href="#">Me contacter !</a></p>
    </footer>

  </body>
</html>
```

Nouvelle structuration

Les autres balises de structure

<hgroup>

La balise `<hgroup>` est utilisée pour grouper des éléments de titres.

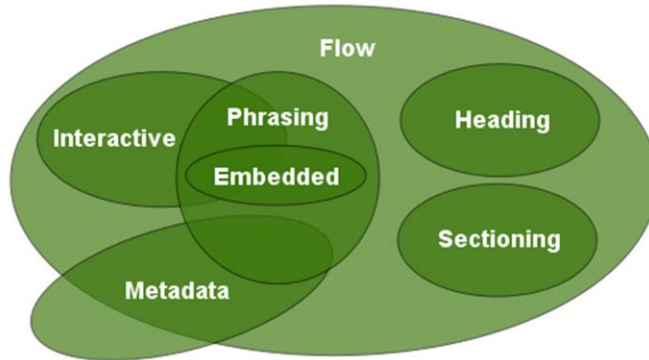
L'élément `<hgroup>` permet de regrouper un ensemble de balises `<h1>` à `<h6>`, quand un titre à plusieurs niveaux (sous-titre).

Exemple d'utilisation de la balise `<hgroup>`

```
<hgroup>  
  <h1>Cours HCS</h1>  
  <h2>HTML5 – CSS3</h2>  
</hgroup>
```

Nouvelle structuration

Les nouvelles catégories de contenus



Le **flux (flow)** regroupe la plupart des éléments courants, c'est-à-dire les autres sous-modèles cités ci-après, ainsi que le contenu texte simple.

Le contenu **interactif** (*interactive*) est destiné à tout ce qui permet une interaction avec l'utilisateur : les liens, les éléments média, les contrôles de formulaire, bref, tout ce qui peut être piloté au clavier, à la souris ou au doigt.

Le contenu de **phrasé** (*phrasing*) correspond à la plupart des éléments pouvant apparaître dans un flux de texte, et qui sont pour la plupart d'entre eux affichés **en-ligne** (par exemple en CSS `display:inline` ou `display:inline-block`). On peut y remarquer entre autres `<audio>`, `<video>`, `<iframe>`, `<canvas>`, ``, les éléments de formulaire tels que `<input>`, `<textarea>`, `<button>`, `<select>` et le balisage du texte ``, ``, `<i>`, ``, etc.

Le contenu **embarqué** (*embedded*) est plus spécialisé : `<audio>`, `<canvas>`, `<embed>`, `<iframe>`, ``, `<object>`, `<video>`, `<svg>`, `<math>`.

A titre indicatif, son apparence CSS est définie par des espaces possédant une hauteur et une largeur, souvent affichés avec un type `display:inline-block`.

La **titraile** (*heading*) comprend tous les titres hiérarchiques (`<h1>` à `<h6>` et `<hgroup>`), qui sont eux aussi par défaut affichés en **bloc** par les navigateurs.

Le **contenu sectionnant** (*sectioning*), définit les grandes zones du document HTML ou de l'application web : `<article>`, `<aside>`, `<nav>`, `<section>`. Par consensus, les navigateurs ont choisi de conférer à ces éléments un rendu CSS de type **bloc**.

Les **métadonnées** (*metadata*) ne relèvent pas du contenu principal mais participent à la définition des informations gravitant autour de ce dernier, par exemple le titre du document (`<title>`), le style (`<style>`), les relations externes (`<link>`), et les scripts (`<script>`).