

XHTML, HTML 5 et web sémantique

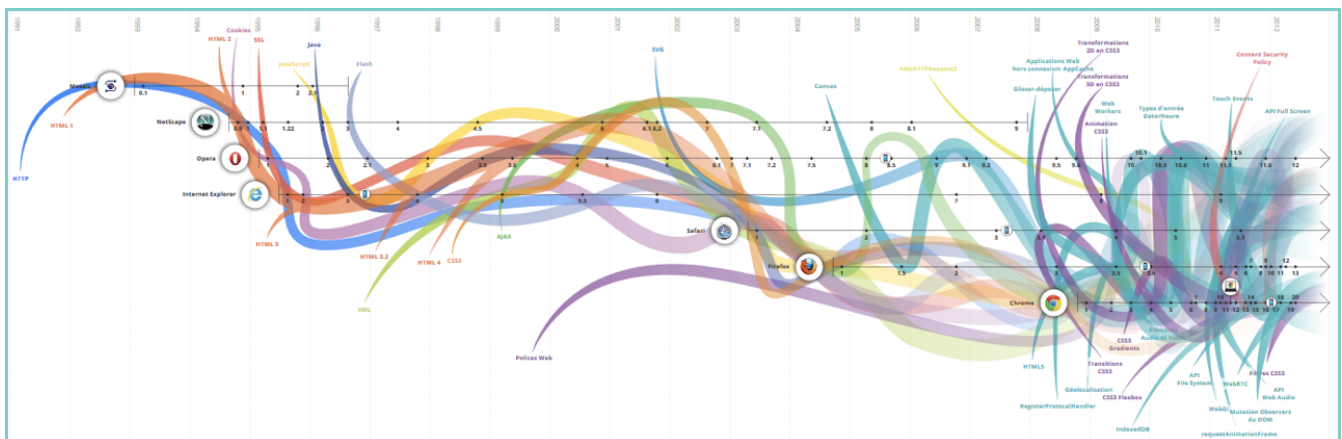
1. Evolution

1.1 Le langage HTML

Le langage HTML a évolué régulièrement depuis sa première standardisation en 1992. Après Mosaïc, la société Netscape crée un navigateur et autorise des nouvelles balises pour répondre aux besoins des développeurs web. Une partie de celles-ci seront intégrées dans la seconde version du langage HTML. En 1994, Tim Berners-Lee fonde le World Wide Web Consortium (ou **W3C**), avec le soutien de la défense américaine DARPA, rassemblant de nombreuses organisations et sociétés informatiques, pour standardiser et diffuser les technologies à la base du Web. Les différentes versions du langage HTML vont résulter des innovations populaires des navigateurs et des besoins exprimés par les membres du consortium. De même, les navigateurs évoluent à leur tour en fonction des standards publiés et de l'évolution des technologies réseaux.

D'une version à une autre du langage HTML, de nouvelles balises sont ajoutées et d'autres sont considérées obsolètes.

En 2009, Tim Berners-Lee fondera la WWW Foundation pour étendre le Web aux pays en voie de développement et garantir un Web ouvert.



Souvenirs, souvenirs

Essayez de vous rappeler en quelle année et avec quel navigateur vous avez surfé sur le web la première fois.

En consultant le site "[Evolution of the web](#)", comparez l'évolution de ce navigateur avec ce que vous utilisez aujourd'hui. Lisez les descriptions des technologies inventées depuis.

2. HTML & XML

2.1 XHTML

Suite à la disparité des versions non standards d'HTML proposées par les différents navigateurs avant les années 2000, les développeurs web étaient en demande d'une norme plus restrictive qui pourrait permettre de

garantir une interprétation correcte des sites webs sur tous les navigateurs.

C'est ainsi qu'en 2000, la première version du standard **XHTML** est publiée. L'XHTML est la fusion de l'HTML et du langage de balises **XML**.

Au même moment est publié HTML 4.0 qui incite à la séparation du contenu (information) et de la forme (rendu visuel).

Exemple : quelques balises devenues obsolètes : `<center>` (centrer), `<u>` (souligner), ou attributs obsolètes : `<body background="">` ... car la mise en forme doit être spécifiée en CSS (CSS = feuille de styles que vous découvrirez très bientôt!)

2.2 Le langage XML

Le XML (**eXtensible Markup Language**) est un langage de balisage mis au point par le XML Working Group sous l'égide du W3C en 1996.

XML est "eXtensible" car on peut définir de nouvelles balises et attributs en fonction de ses besoins. **Les balises XML décrivent le contenu** (à contrario des balises HTML qui décrivent la présentation)

Afin de pouvoir garantir une syntaxe correcte tout en autorisant des balises personnalisées, un fichier XML peut contenir un lien avec une "grammaire", au format **XSD** ou **DTD**, définissant les balises possibles et leurs imbrications autorisées. L'XSD (XML Schema Definition) est un langage de description de format qui remplace peu à peu les DTD (Document Type Definition) car plus puissant.

La version actuelle est l'XML 1.1 (2006). De très nombreux langages sont basés sur l'XML : XHTML, MathML (représentation de formules mathématiques), SVG (images vectorielles), RSS (syndication de contenus web)

Tout document est transformable dans un autre format grâce à **XSLT** (eXtensible Stylesheet Language Transformations). Ainsi un même document XML peut être visualisé en page web HTML ou fichier PDF.

Un document XML est facilement compréhensible et lisible, par contre, il nécessite beaucoup de caractères supplémentaires en plus des données. D'autres formats existent pour partager une structure de données avec un payload moins important. (i.e.: JSON)

```

1 <?xml version="1.1" ?>
2 <livres>
3   <livre id="BK509">
4     <titre>Think Unix</titre>
5     <auteur>John Lasser</auteur>
6     <genre>Informatique</genre>
7   </livre>
8   <livre id="BK831">
9     <titre>Courir</titre>
10    <auteur>Jean Echenoz</auteur>
11    <genre>Roman</genre>
12  </livre>
13 </livres>

```

Exemple de fichier XML

Les règles principales de validation d'un document XML sont les suivantes:

- Sensible à la casse
- Toute balise est fermée
- Valeur des attributs entre guillemets doubles
- Non interprétation : `<![CDATA[contenu non interprété]]>`

2.3 Comparaison HTML et XHTML

HTML

- Interprétation des erreurs
- Insensible à la casse :
`<tag> = <TAG> = <Tag>`
- Version minimaliste des attributs (i.e: selected)
- La fermeture de certains tags est optionnelle (i.e.: `<i>`)
- Les tags vides peuvent être écrits `<tag>` ou `<tag/>`
- DOCTYPE obligatoire
`<!DOCTYPE html>`

XHTML

- Erreurs de syntaxe fatales
- Sensible à la casse
- Les attributs doivent avoir une valeur
- La fermeture des tags est obligatoire
- Les tags vides peuvent être écrits `<tag></tag>` ou `<tag/>`
- DOCTYPE avec DTD recommandé et espace de noms obligatoire
`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`
`<html xmlns="http://www.w3.org/1999/xhtml">`

Le langage HTML est plus tolérant face aux erreurs de syntaxe, le navigateur fait ce qu'il peut pour interpréter le code HTML erroné. Par contre, en XML (et donc en XHTML), toute erreur de syntaxe empêche l'interprétation du document provoquant soit une page blanche soit un message d'erreur.



XHTML ou HTML ?

Regardez le code source de helmo.be et déterminez dans quelle version du langage HTML il a été conçu.

Faites de même pour learn.helmo.be.

3. HTML 5

3.1 HTML 5



HTML 5 est la version standardisée du langage HTML, finalisée en octobre 2014, en remplacement de XHTML1.x et HTML4.

En décembre 2017, la [dernière version \(5.2\)](#) est publiée par le W3C.

3.1.1 Support d'HTML5 par les navigateurs

HTML 5 et compatibilité des navigateurs



(Vidéo: "HTML5 & les navigateurs", durée: 4min36)

Les 2 sites présentés dans la vidéo:

- [HTML 5 test](#)
- [HTML 5 please](#)

3.2 Nouveautés apportées par HTML 5

Ci-dessous, voici une vue d'ensemble des apports d'HTML 5. Ils seront présentés et utilisés soit durant les séances à venir, soit dans le cadre du cours de Javascript en bloc 2.

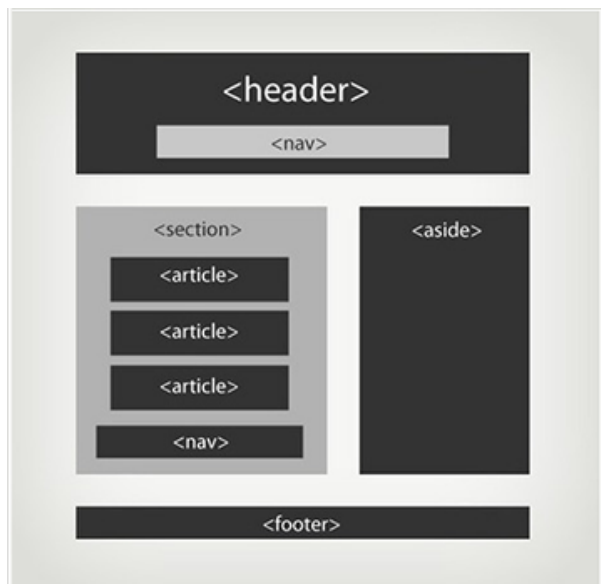
- Des **composants riches** pour les formulaires:
 - Curseur (slider), palette de couleurs, calendrier, jauge, barre de progression, ...
 - Aide pour l'utilisateur (attribut **placeholder**)
 - **Validation** native (email, url, nombre, dates, orthographique)
 - Liste de valeurs : `<datalist>`
- Support multimédia **sans plugin**:
 - Audio : `<audio>`
 - Vidéo : `<video>`
- Support **graphique** (dessin à la volée):
 - canvas2D, canvas3D, SVG : `<canvas>`
- Balises **sémantiques**:
 - `<main>`, `<section>`, `<footer>`, `<header>`, `<nav>`, `<article>`, `<aside>`, ...
 - `<figure>`, `<figcaption>`, `<details>`, `<summary>`, ...
- Stockage **local**:
 - WebSQL, IndexedDB, ApplicationCache...
- **Communication bidirectionnelle**:
 - WebSocket, Notification
- Intégration **desktop**:
 - drag & drop de fichiers, accès partiel au système de fichiers,...
- **Géolocalisation**

3.2.1 Nouveautés d'HTML 5.2

- Une nouvelle balise pour gérer les **fenêtres modales** : `<dialog>`
- La facilitation de liaison avec des systèmes de **paiement en ligne** : `<iframe allowpaymentrequest>`
- Plusieurs `<main>` autorisés sur un même document pour autant qu'un seul soit visible à la fois
- La balise `<style>` est autorisée dans `<body>` même s'il est toujours recommandé de la placer de préférence dans le `<head>`
- Le mode strict d'XHTML est déprécié.

4. Web sémantique

4.1 L'intérêt du Web Sémantique



Depuis HTML 4.0, il faut distinguer les données, en HTML, de leur représentation visuelle (en CSS). Plus important encore depuis HTML 5, il faut donner du sens, la sémantique, aux données qui soit compréhensible par les navigateurs et les robots.

En effet, un navigateur dont l'espace d'affichage est limité ou un logiciel de lecture de contenu pour mal-voyants qui doit rythmer sa lecture doivent pouvoir déterminer comment découper et proposer le contenu intelligemment. Il est donc important de pouvoir les aider à déterminer les zones d'informations et leurs associations (corrélation, redondance, importance, ...).

De même, les robots indexeurs de contenu vont

rechercher des mot-clefs dans l'ensemble de la page et les pondérer en fonction de l'importance sémantique des zones dans lesquelles ils se trouvent. Ainsi un mot présent dans le titre principal de la page est plus important qu'un mot-clef présent dans un paragraphe d'une zone optionnelle de contenu.

Certaines balises comme `` et `<div>` sont des conteneurs d'éléments HTML (ou de texte brut) mais n'ont aucune signification sémantique. Ces balises sont utilisées pour aider à appliquer des styles CSS. Le `` est un conteneur de mots (d'un paragraphe, d'un titre, ...) tandis que `<div>` peut contenir n'importe quel élément HTML. Nous reviendrons plus précisément sur la distinction entre ces balises avec le positionnement en CSS.

```
1 <div>
2   <p>Un premier <span>paragraphe</span></p>
3   <p>Un second paragraphe lié</p>
4 </div>
```

4.2 Balises HTML 5 principales

`<main>` : contenu principal de la page supposé varier de page en page d'un même site,
`<article>` : contenu qui peut garder son sens s'il est utilisé en dehors de la page,
`<section>` : groupement de contenus au sein de la page, d'une section, ou d'un article. Bien que non obligatoire, il est conseillé de spécifier un titre aux sections,
`<header>` : en-tête de page, d'article ou de section,
`<footer>` : pied de page, d'article ou de section,
`<nav>` : groupement d'éléments de navigation (menus),
`<aside>` : information supplémentaire facultative liée à la page, une section ou un article, non nécessaire à la compréhension du contenu lié. Il est recommandé que le contenu d'un `<aside>` puisse se suffire à lui-même (sur mobile, `<aside>` peut être affiché à la demande et masquer le contenu lié!)

En HTML 5.0, `main`: une seule balise `main` par page ! Depuis HTML 5.2, plusieurs `<main>` autorisées pour autant qu'un seul soit visible à la fois.

```
1 <body>
2   <header></header>
3   <main></main>
4   <footer></footer>
5 </body>
```

Structure classique de page web

4.2.1 Les titres

Pour faciliter la compréhension d'un document écrit, une structuration de l'écrit grâce aux titres, sous-titres, ... est importante. C'est également le cas pour une page web. C'est d'autant plus le cas que les titres sont nécessaires aux lecteurs humains, aux robots indexeurs et aux logiciels de lecture pour mal-voyants.

Un lecteur de contenu web balaye la page du regard (lecture en diagonale) pour évaluer rapidement si le contenu l'intéresse. Les robots indexeurs de contenus utilisent les mots-clefs présents dans les différents niveaux de titre et leur accordent une importance relative aux niveaux. Les systèmes de lecture automatisée de contenu web pour mal-voyants exploitent également les titres pour rythmer correctement le parcours de la page.

Les recommandations de l'usage des différentes niveaux de titres (h1 à h6) sont les suivantes :

- **Hiérarchie correcte** des niveaux de titres (Exemple: pas de <h3> s'il n'y a pas un <h2> et un <h1>),
- **Un seul <h1>** pour une même page. Les <article> et les <section> commencent, si possible, par un <h2>.
- Les titres sont rédigés **pour le lecteur** en premier lieu et non comme un agrégat de mots-clés pour manipuler les robots indexeurs de contenus,
- Les balises de titre sont utilisées pour **donner du sens** au contenu et non pour l'aspect graphique d'un texte. Les feuilles de style CSS servent à définir le rendu visuel de n'importe quel élément HTML.
- Pas trop de titres, ni trop peu ! En général un titre par page, article, section ; des sous-titres si nécessaire

Incorrect

```

1 <body>
2   <h1>Un exemple de titre</h1>
3   <h2>1) Sous titre</h2>
4   <h3>2) Sous titre</h3>
5   <h4>3) Sous titre</h4>
6   <h5>3.1) Sous sous titre</h5>
7 </body>

```

Correct

```

1 <body>
2   <h1>Un exemple de titre</h1>
3   <h2>1) Sous titre</h2>
4   <h2>2) Sous titre</h2>
5   <h2>3) Sous titre</h2>
6   <h3>3.1) Sous sous titre</h3>
7   <h2>4) Sous titre</h2>
8 </body>

```

4.2.2 Comparaison sans ou avec balises sémantiques

Sans

```

1 (...)
2 <div class="monMenu">
3   <a href="accueil.html">accueil</a>
4   <a href="contact.html">contact</a>
5 </div>
6 <h1>Vos derniers messages</h1>
7 <div class="message">
8   <table>
9     <tr><td>Bienvenue</td></tr>
10    <tr><td>Salut Marc, (...) </td></tr>
11    <tr><td>Ecrit par John</td></tr>
12  </table>
13  <div class="profil">Infos sur John</div>
14 </div>
15 <div class=" message ">(...)

```

Avec

```

1 (...)
2 <header>
3   <nav class="monMenu">
4     <a href="accueil.html">accueil</a>
5     <a href="contact.html">contact</a>
6   </nav>
7 </header>
8 <main>
9   <h1>Vos derniers messages</h1>
10  <article class="message">
11    <header><h2>Bienvenue</h2></header>
12    <section>Salut Marc, (...) </section>
13    <footer>Ecrit par John</footer>
14    <aside class="profil">Infos sur John</aside>
15  </article>
16  <article class=" message ">(...)

```

Qu'en pensez-vous ? Quelle structure vous paraît la plus compréhensible ?

4.2.3 Les éléments avec légende

- <figure>: Conteneur pour une unité de contenu (images, vidéos, tableaux, blocs de texte)
- <figcaption>: Légende associée au conteneur parent

```

1 <figure>
2   
3   <figcaption>Jean Dupont</figcaption>
4 </figure>

```



Jean Dupont

4.2.4 Éléments escamotables

Les éléments escamotables permettent d'afficher ou masquer le **détail** d'un élément en **cliquant** sur une version **résumée**.

```

1 <details open>
2   <summary>5 utilisateurs</summary>
3   <ul>
4     <li>J. Dujardin</li>
5     <li>O. Sy</li>
6     <li>V. Lemercier</li>
7     <li>J. Reno</li>
8     <li>A. Chabat</li>
9   </ul>
10 </details>

```

5 utilisateurs

- J. Dujardin
- O. Sy
- V. Lemercier
- J. Reno
- A. Chabat



Recherche des balises sémantiques

Regardez le code source de [helmo.be](https://dartagnan.cg.helmo.be/~p150107/tutoriels/html-5/) et regardez l'usage qui est fait des balises sémantiques. Qu'en pensez-vous ?

Faites de même pour learn.helmo.be.

5. Bonnes pratiques

5.1 Écriture du code HTML

Lorsque vous rédigez du code HTML, veillez à respecter les conventions suivantes :

- Les balises et attributs écrits en **minuscules**
- Spécifier la **version HTML** (DOCTYPE)
- Utilisation de " et non ' pour les attributs
- Fermer chaque balise (même les non obligatoires comme , <body>...)
- Les éléments imbriqués sont indentés
- Spécifier l'encodage des caractères en UTF-8 (meta charset)
- Spécifier la langue utilisée pour le contenu (html lang="fr")
- Pas de / pour fermer les balises autofermantes.(img, br, hr,...)
- Respecter la hiérarchie des titres (h1...h6) et la sémantique des balises (i.e. : pas de tableau pour mise en page)

Spécifier la langue utilisée pour le contenu est utile aux synthèses vocales (pour adopter le bon accent) et aide au référencement par une meilleure interprétation des termes par les robots indexeurs.

5.1.1 Exemple

```

1 <!DOCTYPE html>
2 <html lang="fr">
3   <head>
4     <meta charset="UTF-8">
5     <title>Titre de page</title>
6   </head>
7   <body>
8     <main>
9       <h1 class="maClasse">Mon titre</h1>
10      <h2>Mon sous-titre</h2>
11      
12      <hr>
13      <br>
14    </main>
15  </body>
16 </html>

```

5.1.2 Réduire les balises au minimum

Plus vous utilisez des balises et plus vous les imbriquez et plus cela risque d'augmenter le temps de chargement de la page. De plus, une structure de balises simplifiée est également plus facile à maintenir.

Version complexe

```
<div class="photoProfil"></div>
```

Version simplifiée

```

```

5.1.3 Ordre des attributs

Pour le navigateur, l'ordre des attributs des balises n'a aucune importance. Par contre, pour les développeurs et les web designers qui manipulent ce code, avoir une convention d'ordre des attributs permet de faciliter la lecture du code. Comparez vous-mêmes.

Attributs dans un ordre quelconque

```
1 <ul>
2   <li data-auteur="J.R.R Tolkien" class="livre" id="livre1">Le Hobbit</li>
3   <li id="livre2" data-auteur="F. Herbert" class="livre">Dune</li>
4   <li class="livre" id="livre3" data-auteur="H.P. Lovecraft">L'appel de Cthulhu</li>
5 </ul>
```

Attributs ordonnés

```
1 <ul>
2   <li class="livre" id="livre1" data-auteur="J.R.R Tolkien">Le Hobbit</li>
3   <li class="livre" id="livre2" data-auteur="F. Herbert">Dune</li>
4   <li class="livre" id="livre3" data-auteur="H.P. Lovecraft">L'appel de Cthulhu</li>
5 </ul>
```

L'ordre des attributs recommandé est le suivant:

1. **class** : liaison la plus courante avec les feuilles de styles CSS,
2. **id, name** : identifiants généralement manipulés par Javascript ou par soumission de formulaire,
3. **data-*** : attributs de données personnalisables
4. **src, for, type, href, value** : attributs contenant une valeur de taille réduite
5. **title, alt** : attributs contenant une expression plus ou moins longue
6. **role, aria-*** : attributs liés à l'amélioration de l'accessibilité ou au web sémantique.

5.2 Respect des standards

5.2.1 Orthographe

De trop nombreuses fautes d'orthographe dans un rapport ou un site web décrédibilise son auteur ou la société qu'il représente. Il existe suffisamment d'outils de correction orthographique pour remédier aux fautes d'inattention ou à des lacunes de maîtrise de langue.

De plus, des mots-clefs écrits incorrectement ne favorisent pas non plus un bon référencement.

5.2.2 Langage HTML & CSS

Des erreurs de syntaxe ou un non respect des standards HTML et CSS risquent d'induire un rendu visuel fort variable d'un navigateur à l'autre voir rendre une partie des contenus non visibles aux utilisateurs.

Pensez à tirer profit des vérificateurs mis à disposition par le W3C :

- [Vérificateur HTML](#)
- [Vérificateur CSS](#)

6. Exercices & QCM

6.1 Exercices

6.1.1 Exercice html5.1

Créez une page "**exercice-semantique.html**" avec l'éditeur Notepad++.

Sur base de la maquette "fil de fer" nommée [exercice-semantique.pdf](#), écrivez la structure du code HTML en tirant parti au maximum des balises sémantiques.

6.1.2 Exercice html5.2

Téléchargez le fichier "[exercice-debug-W3C.html](#)" et copiez-collez le code HTML dans le [validateur W3C](#).

Tant qu'il reste des erreurs:

- Lisez les explications liées à la première erreur,
- Corrigez l'erreur,
- Copiez-collez la nouvelle version du code et validez-la à nouveau...

Ne corrigez qu'une erreur à la fois ! Le but de l'exercice est de vous confronter aux différents messages d'erreur du validateur !

6.2 Questionnaire formatif

Lorsque vous avez terminé le tutoriel et les exercices, testez vos connaissances et votre compréhension avec le questionnaire QCM.

Questionnaire QCM

Vous pouvez passer ce questionnaire autant de fois que vous le désirez et la note obtenue n'intervient pas dans la note année.

Une cote de 14/20 minimum est nécessaire pour libérer le chapitre suivant.