

Balises documentaires en HTML5

Table des matières

Introduction	3
I - La structuration logique	4
II - Balisage documentaire en HTML	7
III - Div et span	8
IV - Exercice	9
V - Sections en HTML5	11
VI - Exercice	15
VII - Valider un fichier HTML	17
Solutions des exercices	19
Glossaire	22
Bibliographie	23
Crédits des ressources	24
Contenus annexes	25

Introduction

Durée : 45 minutes

I La structuration logique

« Un document peut être décrit comme une collection d'objets comportant des objets de plus haut niveau composés d'objets plus primitifs. Les relations entre ces objets représentent les relations logiques entre les composants du document. Par exemple [...] un livre est divisé en chapitres, chaque chapitre en sections, sous-sections, paragraphes, etc. Une telle organisation documentaire est appelée **représentation de document structuré**.

(traduit depuis la préface de (André et al., 1989)^{André89 p.23})



Structuration logique (ou structuration documentaire)

Az Définition

On appelle structuration logique d'un contenu une inscription explicitant la structure de ce contenu en fonction de son organisation et des attributs intrinsèques qui le caractérisent plutôt qu'en fonction de propriétés de présentation sur un support.

Synonyme : structuration documentaire

Document abstrait (ou document structuré)

Az Définition

Un document décrit par sa structure logique est appelé document abstrait, on parle aussi de **document structuré logiquement** (et souvent simplement de document structuré).

👁 Exemple

```
1 <exercice>
2   <enonce> Soit un triangle rectangle disposant d'un angle de 30 degrés.
   </enonce>
3   <question> Donner la valeur des autres angles du triangle. </question>
4   <indice> La somme des angles d'un triangle est égale à 180 degrés. </indice>
5   <solution> 90 et 60 degrés. </solution>
6 </exercice>
7
```

Structuration physique (ou mise en forme)

Az Définition

On appelle structuration physique ou mise en forme d'un contenu une inscription décrivant la façon dont ce contenu doit être présenté sur un support donné.

Document formaté

Az Définition

Un document décrit par sa structure physique est appelé document formaté, c'est en général également ce dont on parle quand on parle simplement de document (au sens d'information organisée pour être lisible par un humain).

Exemple



Fondamental

- Il est possible de calculer une ou plusieurs structurations physiques pour une même structuration logique.
- Il est possible de calculer d'autant plus de structurations physiques que la structuration logique est indépendante de ses supports de présentation.

Complément

Exemple de structuration logique (cf. p.25)

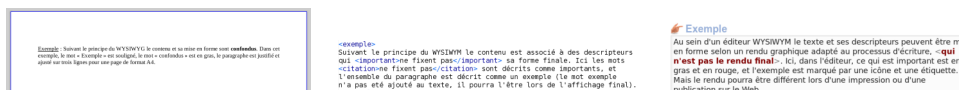
Les avantages de la structuration logique (cf. p.26)

WYSIWYM et WYSIWYG

Complément

- La structuration physique est associée au WYSIWYG.
- La structuration logique est associée au WYSIWYM.

Écriture WYSIWYM : "What you see is what you mean". <https://aswemay.fr/co/010024.html>




Exemple WYSIWYG et WYSIWYM

Complément

Écriture numérique traditionnelle et écriture numérique computationnelle. <https://aswemay.fr/co/010023.html>

Voir aussi

 Complément

Langages XML orienté documents (cf. p.27)

II Balisage documentaire en HTML

Rappel

La structuration documentaire, ou structuration logique, consiste à utiliser des balises qui décrivent le type de contenu plutôt que la façon de le mettre en forme.

La structuration en HTML

Le langage HTML avant sa version 5 possède déjà plusieurs moyens pour exprimer la structuration logique d'une page web :

- balises de titres h1 à h6 ;
- balises d'emphase `` et de contenu important `` ;
- balises de citation `<quote>` et `<blockquote>` ;
- etc.

En réalité, toute balise de HTML non générique (`<div>` ou ``) véhicule une sémantique associée.

Attention

Il est particulièrement déconseillé de détourner la fonction des balises, par exemple l'utilisation d'un tableau (`<table>`) à des fins de mise en forme (alignement de données). Ceci sera le rôle de CSS.

Structures implicites

Complément

Depuis sa création, le langage HTML a pour vocation de structurer le contenu d'une page web. Pourtant, et jusqu'à la version HTML4, les balises de structuration documentaire demeurent peu nombreuses et la balise `div` est trop générique pour bien aider à structurer les pages. Ce manque de solution standard a conduit les développeurs à utiliser des classes spécifiques pour distinguer les éléments : header, top, bottom, footer, menu, navigation, main, container, content, article, sidebar, topnav, etc.).

Les navigateurs ou moteurs d'indexation ne peuvent pas correctement tirer parti de ces structures cachées.

III Div et span

Az Définition

En HTML, l'élément `div` permet de définir un bloc de contenu générique non typé a priori. Il est en général associé à un attribut `class` qui permet de lui associer un rôle dans la structuration de la page.

📄 Syntaxe

```
1 <div class="nom-de-la-classe">...</div>
```

👁 Exemple

```
1 <div class="source"><a href="https://html.spec.whatwg.org/multipage/grouping-
  content.html#the-div-element"> The div element has no special meaning at all.
  </a></div>
2
```

Az Définition

L'élément `span` fonctionne à l'instar de `div` pour les éléments de texte *inline*, à l'intérieur d'un paragraphe de texte (ou d'un `div`).

👁 Exemple

```
1 <p>The span element <span class="note">doesn't mean anything</span> on its own.
  </p>
```

div

⊕ Complément

developer.mozilla.org/fr/docs/Web/HTML/Element/div

span

⊕ Complément

developer.mozilla.org/fr/docs/Web/HTML/Element/span

IV Exercice

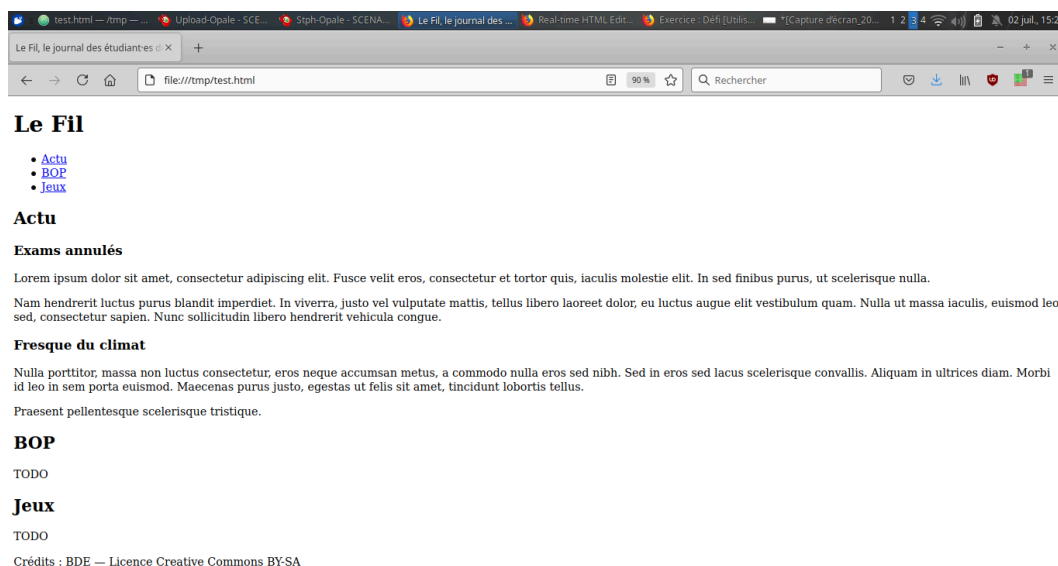
Soit le texte suivant correspondant au journal en cours de rédaction **Le Fil**.

```
1 Le Fil
2
3 Actu
4
5 Exams annulés
6
7 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce velit eros,
  consectetur et tortor quis, iaculis molestie elit. In sed finibus purus, ut
  scelerisque nulla.
8 Nam hendrerit luctus purus blandit imperdiet. In viverra, justo vel vulputate mattis,
  tellus libero laoreet dolor, eu luctus augue elit vestibulum quam. Nulla ut massa
  iaculis, euismod leo sed, consectetur sapien. Nunc sollicitudin libero hendrerit
  vehicula congue.
9
10 Fresque du climat
11
12 Nulla porttitor, massa non luctus consectetur, eros neque accumsan metus, a commodo
  nulla eros sed nibh. Sed in eros sed lacus scelerisque convallis. Aliquam in ultrices
  diam. Morbi id leo in sem porta euismod. Maecenas purus justo, egestas ut felis sit
  amet, tincidunt lobortis tellus.
13 Praesent pellentesque scelerisque tristique.
14
15 BOP
16 TODO
17
18 Jeux
19 TODO
20
21 Crédits : BDE – Licence Creative Commons BY-SA
22
```

Question

[solution n°1 p. 19]

Créer une page HTML en structurant le contenu à l'aide des balises div, h1, h2 et h3 et p, en vue d'obtenir le résultat ci-après.



Indice :

On ajoutera un `<div id="header">` pour le titre principal et un `<div id="footer">` en pied de page pour les crédits.

Indice :

```
1 <!DOCTYPE html>
2 <html lang="fr">
3
4 <head>
5   ...
6 </head>
7
8 <body>
9
10 <div id="header">
11   <h1>...</h1>
12 </div>
13
14 <div id="menu">
15   <ul>
16     ...
17   </ul>
18 </div>
19
20 <div id="content">
21   <h2>Actu</h2>
22
23   <div id="post1">
24     ...
25   </div>
26
27   <div id="post2">
28     ...
29   </div>
30
31 </div>
32
33 ...
34
35 <div id="footer">
36   ...
37 </div>
38
39 </body>
40 </html>
41
```

V Sections en HTML5

Présentation des balises de section

Ci-dessous les principales balises de section introduites par HTML5 :

« `<section>`

Section générique regroupant un même sujet, une même fonctionnalité, de préférence avec un en-tête, ou bien section d'application web.

`<nav>`

Section possédant des liens de navigation principaux (au sein du document ou vers d'autres pages).

`<article>`

Section de contenu indépendante, pouvant être extraite individuellement du document ou syndiquée (flux RSS ou équivalent), sans pénaliser sa compréhension.

`<aside>`

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.

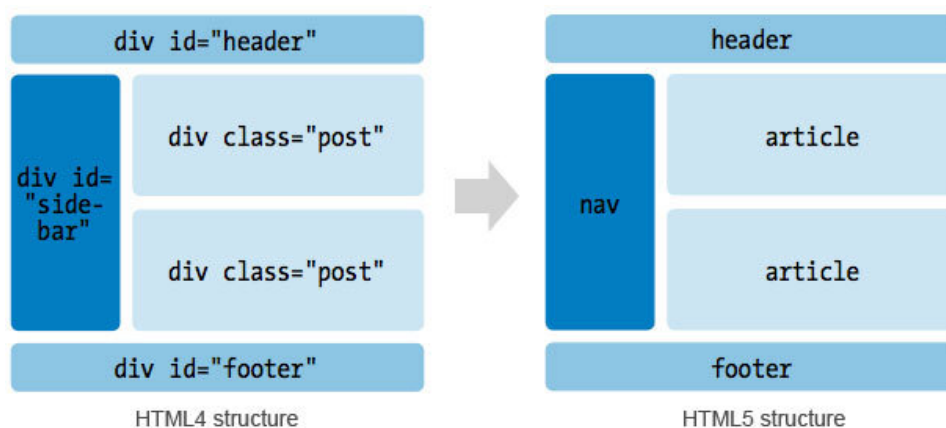
`<header>`

Section d'introduction d'un article, d'une autre section ou du document entier (en-tête de page).

`<footer>`

Section de conclusion d'une section ou d'un article, voire du document entier (pied de page). »

Source : <http://www.alsacreations.com/article/lire/1376-html5-section-article-nav-header-footer-aside.html>.



Sectionnement en HTML4 et en HTML5

Non-unicité des éléments header/footer

Remarque

Comme le suggère la description faite ci-dessus, les éléments header et footer peuvent être utilisés au sein d'une section ; on peut donc trouver plusieurs éléments de ce type dans une même page web.

On pourra par exemple inclure dans le header d'une section le titre et sous titre de cette dernière, et dans son footer les références associées.

Page type

 Exemple

Une page type en HTML5 reprenant les nouvelles balises de structuration pourrait avoir le squelette décrit ci-dessous.

```

1 <body>
2
3 <header>
4   <h1>Welcome On Our Website!</h1>
5   <p>Here is our logo and slogan.</p>
6 </header>
7
8 <nav>
9   <header>
10    <h2>Choose Your Interest</h2>
11  </header>
12  <ul>
13    <li>Menu 1</li>
14    <li>Menu 2</li>
15    <li>Menu 3</li>
16  </ul>
17 </nav>
18
19 <article>
20   <header>
21     <h1>Title of Article</h1>
22     <h2>Subtitle of Article</h2>
23   </header>
24
25   <section>
26     <h3>First Logical Part (e.g. "Theory")</h3>
27     <p>Paragraph 1 in first section</p>
28
29     <h4>Some Other Subheading in First Section</h4>
30     <p>Paragraph 2 in first section</p>
31   </section>
32
33   <section>
34     <h3>Second Logical Part (e.g. "Practice")</h3>
35     <p>Paragraph 1 in second section</p>
36     <p>Paragraph 2 in second section</p>
37   </section>
38
39   <footer>
40     <h4>Author Bio</h4>
41     <p>Paragraph in Article's Footer</p>
42   </footer>
43
44 </article>
45
46 <aside>
47
48   <h2>Get To Know Us Better</h2>
49
50   <section>
51     <h3>Popular Posts</h3>
52     <ul>...</ul>

```

```

53     </section>
54
55     <section>
56         <h3>Partners</h3>
57         <ul>...</ul>
58     </section>
59
60     <section>
61         <h3>Testimonials</h3>
62         <ul>...</ul>
63     </section>
64
65 </aside>
66
67 <footer>
68     <ul>
69         <li>Copyright</li>
70         <li>Social Media Links</li>
71     </ul>
72 </footer>
73
74 </body>

```

Source : <http://www.hongkiat.com/blog/html-5-semantic/>.

L'intérêt des balises de section par rapport à HTML4

Quelques problèmes rencontrés en HTML4 et comment HTML5 les résout :

HTML4	HTML5
La transclusion d'un sous-document dans un autre est difficile car il est souvent nécessaire de devoir modifier la hiérarchie des titres. En effet, on ne peut avoir qu'un groupe d'éléments h1 ayant eux mêmes des éléments h2, etc.	Chaque section possède ses propres titres sans conflit avec ceux des sections de même niveau par exemple. Une transclusion peut donc se faire de manière quasi-immédiate sans modifier la hiérarchie des titres.
Chaque section fait partie de la structure du document. Il n'existe pas de section associée au site, comme les menus de navigation, le pied de page principal (avec mentions légales, contact...).	Les éléments tels que nav, header ou footer en dehors de toute section contiennent des informations relatives au site de manière générale. Voir également dans le complément ci-après la balise <main>.

- Avec HTML5, certains navigateurs peuvent :
 - adapter l'affichage du contenu selon le support (mobile ou non) ;
 - générer automatiquement une table des matières du document avec sauts rapides.
 - Ensuite, les technologies d'assistance (personnes aveugles...) peuvent délivrer un contenu adapté à son rôle dans le document (titre, paragraphe...).
- Enfin, les moteurs d'indexation améliorent l'accessibilité des pages bien structurées.

La balise main

[+ Complément](#)

Assez peu connue, la balise `main` introduite par HTML5 permet de spécifier le contenu principal d'une page web. Son contenu devrait être spécifique à la page en cours, et donc non accessible ailleurs sur le site (*a contrario* des liens de navigation générale, du logo du site...). Elle devrait se substituer aux `div` qualifiées "main containers" que l'on retrouve sur de nombreuses pages web écrites en HTML4.

Liste exhaustive des balises de section disponibles

[+ Complément](#)

http://www.w3schools.com/html/html5_semantic_elements.asp

Liste exhaustive des balises HTML

[+ Complément](#)

<http://www.w3.org/TR/html-markup/elements.html>

Mieux comprendre comment utiliser les balises de section en HTML5

[+ Complément](#)

<http://blog.teamtreehouse.com/use-html5-sectioning-elements>

VI Exercice

Soit la page HTML suivante correspondante au journal en ligne **Le Fil** (en cours de rédaction).

```
1 <!DOCTYPE html>
2 <html lang="fr">
3
4 <head>
5   <meta charset="utf-8">
6   <title>Le Fil, le journal des étudiant·es de l'UTC</title>
7 </head>
8
9 <body>
10
11 <div id="header">
12   <h1>Le Fil</h1>
13 </div>
14
15 <div id="menu">
16   <ul>
17     <li><a href="#actu">Actu</a></li>
18     <li><a href="#BOP">BOP</a></li>
19     <li><a href="#Jeux">Jeux</a></li>
20   </ul>
21 </div>
22
23 <div class="content" id="actu">
24   <h2>Actu</h2>
25
26   <div class="post">
27     <h3>Exams annulés</h3>
28     <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce velit eros,
29     consectetur et tortor quis, iaculis molestie elit. In sed finibus purus, ut
30     scelerisque nulla.</p>
31     <p>Nam hendrerit luctus purus blandit imperdiet. In viverra, justo vel vulputate
32     mattis, tellus libero laoreet dolor, eu luctus augue elit vestibulum quam. Nulla ut
33     massa iaculis, euismod leo sed, consectetur sapien. Nunc sollicitudin libero
34     hendrerit vehicula congue.</p>
35   </div>
36
37   <div class="post">
38     <h3>Fresque du climat</h3>
39     <p>Nulla porttitor, massa non luctus consectetur, eros neque accumsan metus, a
40     commodo nulla eros sed nibh. Sed in eros sed lacus scelerisque convallis. Aliquam in
41     ultrices diam. Morbi id leo in sem porta euismod. Maecenas purus justo, egestas ut
42     felis sit amet, tincidunt lobortis tellus.</p>
43     <p>Praesent pellentesque scelerisque tristique.</p>
44   </div>
45 </div>
46
47 <div class="content" id="bop">
48   <h2>BOP</h2>
49   <p>TODO</p>
50 </div>
51
52 <div class="content" id="jeux">
53   <h2>Jeux</h2>
54   <p>TODO</p>
55 </div>
56
57 <div id="footer">
58   <p>Crédits : BDE – Licence Creative Commons BY-SA</p>
```

Exercice


```
51 </div>
52
53 </body>
54 </html>
55
```

Question

[solution n°2 p. 20]

Créer une page HTML5 utilisant les balises : header, nav, section, article, footer.

VII Valider un fichier HTML

 Fondamental

Les fichiers HTML que l'on écrit et que l'on met en ligne doivent respecter les standard du W3C.

Utiliser un éditeur colorisant

 Rappel

Introduction générale à HTML (cf. p.33)

Afficher la source

 Méthode

Les navigateurs web proposent une fonction permettant d'afficher la source HTML de la page que l'on consulte.

Sous Firefox : Clic droit puis View Page Source (ou ajouter view-source: devant n'importe quelle URL).

 Exemple

```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <!-- Primary Meta Tags -->
5   <meta charset="utf-8">
6 [...]
7   <!-- Page Title -->
8   <title>Gauthier Roussilhe</title>
9 [...]
10 </head>
11 <body>
12   <header class="header">
13     <h1 class="site-title"><a href="../index.html">Gauthier Roussilhe</a>
14   </h1>
15 </header>
16 [...]
17
```

view-source:https://gauthierroussilhe.com/post/paradoxes-enjeux.html (extraits)

W3C Markup Validation Service

 Méthode

Le W3C propose un service permettant de valider le code HTML d'une page en ligne ou locale : <https://validator.w3.org/>

W3C Markup Validation Service
Check the markup (HTML, XHTML, ...) of Web documents

This document was successfully checked as XHTML 1.0 Strict!

Result:	Passed	
Address:	<input type="text" value="https://www.bortzmeyer.org/enfer-numerique.html"/>	
Encoding:	utf-8	(detect automatically) ▼
Doctype:	XHTML 1.0 Strict	(detect automatically) ▼
Root Element:	html	
Root Namespace:	http://www.w3.org/1999/xhtml	

Congratulations

The document located at <<https://www.bortzmeyer.org/enfer-numerique.html>> was successfully checked as XHTML 1.0 Strict. This means that the resource in question identified itself as "XHTML 1.0 Strict" and that we successfully performed a formal validation of it. The parser implementations we used for this check are based on [OpenSP](#) (SGML/XML) and [libxml2](#) (XML).

<https://validator.w3.org/check?uri=https%3A%2F%2Fwww.bortzmeyer.org%2Fenfer-numerique.html...>¹ (extrait)

¹. https://validator.w3.org/check?uri=https%3A%2F%2Fwww.bortzmeyer.org%2Fenfer-numerique.html&charset=%28detect+automatically%29&doctype=Inline&group=0&user-agent=W3C_Validator%2F1.3+http%3A%2F%2Fvalidator.w3.org%2Fservices

Solutions des exercices

Solution n°1

[exercice p. 9]

```
1 <!DOCTYPE html>
2 <html lang="fr">
3
4 <head>
5   <meta charset="utf-8">
6   <title>Le Fil, le journal des étudiant·es de l'UTC</title>
7 </head>
8
9 <body>
10
11 <div id="header">
12   <h1>Le Fil</h1>
13 </div>
14
15 <div id="menu">
16   <ul>
17     <li><a href="#actu">Actu</a></li>
18     <li><a href="#BOP">BOP</a></li>
19     <li><a href="#Jeux">Jeux</a></li>
20   </ul>
21 </div>
22
23 <div class="content" id="actu">
24   <h2>Actu</h2>
25
26   <div class="post">
27     <h3>Exams annulés</h3>
28     <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce velit eros,
consectetur et tortor quis, iaculis molestie elit. In sed finibus purus, ut
scelerisque nulla.</p>
29     <p>Nam hendrerit luctus purus blandit imperdiet. In viverra, justo vel vulputate
mattis, tellus libero laoreet dolor, eu luctus augue elit vestibulum quam. Nulla ut
massa iaculis, euismod leo sed, consectetur sapien. Nunc sollicitudin libero
hendrerit vehicula congue.</p>
30   </div>
31
32   <div class="post">
33     <h3>Fresque du climat</h3>
34     <p>Nulla porttitor, massa non luctus consectetur, eros neque accumsan metus, a
commodo nulla eros sed nibh. Sed in eros sed lacus scelerisque convallis. Aliquam in
ultrices diam. Morbi id leo in sem porta euismod. Maecenas purus justo, egestas ut
felis sit amet, tincidunt lobortis tellus.</p>
35     <p>Praesent pellentesque scelerisque tristique.</p>
36   </div>
37 </div>
38
39 <div class="content" id="bop">
40   <h2>BOP</h2>
41   <p>TODO</p>
42 </div>
43
44 <div class="content" id="jeux">
45   <h2>Jeux</h2>
46   <p>TODO</p>
47 </div>
48
49 <div id="footer">
```

```

50 <p>Crédits : BDE – Licence Creative Commons BY-SA</p>
51 </div>
52
53 </body>
54 </html>
55

```

Solution n°2

[exercice p. 16]

Solution en HTML5

```

1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="utf-8">
6   <title>Le Fil, le journal des étudiant·es de l'UTC</title>
7 </head>
8
9 <body>
10
11 <header>
12   <h1>Le Fil</h1>
13 </header>
14
15 <nav>
16   <ul>
17     <li><a href="#actu">Actu</a></li>
18     <li><a href="#BOP">BOP</a></li>
19     <li><a href="#Jeux">Jeux</a></li>
20   </ul>
21 </nav>
22
23 <section id="actu">
24   <h2>Actu</h2>
25
26   <article>
27     <h3>Exams annulés</h3>
28     <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce velit eros,
consectetur et tortor quis, iaculis molestie elit. In sed finibus purus, ut
scelerisque nulla.</p>
29     <p>Nam hendrerit luctus purus blandit imperdiet. In viverra, justo vel vulputate
mattis, tellus libero laoreet dolor, eu luctus augue elit vestibulum quam. Nulla ut
massa iaculis, euismod leo sed, consectetur sapien. Nunc sollicitudin libero
hendrerit vehicula congue.</p>
30   </article>
31
32   <article>
33     <h3>Fresque du climat</h3>
34     <p>Nulla porttitor, massa non luctus consectetur, eros neque accumsan metus, a
commodo nulla eros sed nibh. Sed in eros sed lacus scelerisque convallis. Aliquam in
ultrices diam. Morbi id leo in sem porta euismod. Maecenas purus justo, egestas ut
felis sit amet, tincidunt lobortis tellus.</p>
35     <p>Praesent pellentesque scelerisque tristique.</p>
36   </article>
37
38 </section>
39
40 <section id="bop">
41   <h2>BOP</h2>
42   <p>TODO</p>
43 </section>
44

```

```
45 <section id="jeux">
46   <h2>Jeux</h2>
47   <p>TODO</p>
48 </section>
49
50 <footer>
51   <p>Crédits : BDE – Licence Creative Commons BY-SA</p>
52 </footer>
53
54 </body>
55 </html>
56
```

Glossaire

Sérialisation

Processus consistant à enregistrer des données en mémoire vive (par exemple des objets) sous une forme permettant leur persistance, typiquement sur une mémoire secondaire.

Bibliographie

[André89] Jacques André, Richard Furuta, Vincent Quint, *Structured documents*, Cambridge University Press, 1989.

[Dupoirier, 1995] Gérard Dupoirier, *Technologie de la GED : Techniques et management des documents électroniques*, Hermes, 1995.

Crédits des ressources

Sectionnement en HTML4 et en HTML5 p. 11

<http://ingus.info/blog/introducing-html5-second-edition-book-review.html>

Contenus annexes

1. Exemple de structuration logique

Un exercice structuré logiquement

Exemple

Soit la structuration logique d'un exercice :

```
1 Exercice = {Enonce, Question, Indice, Solution}
2 avec
3 Enonce = Soit un triangle rectangle disposant d'un angle de 30 degrés.
4 Question = Donner la valeur des autres angles du triangle.
5 Indice = La somme des angles d'un triangle est égale à 180 degrés.
6 Solution = 90 et 60 degrés.
```

```
1 <exercice>
2   <enonce> Soit un triangle rectangle disposant d'un angle de 30 degrés.
   </enonce>
3   <question> Donner la valeur des autres angles du triangle. </question>
4   <indice> La somme des angles d'un triangle est égale à 180 degrés. </indice>
5   <solution> 90 et 60 degrés. </solution>
6 </exercice>
7
```

Il est possible à partir de cette représentation de calculer différentes présentations.

1. Pour l'écran on peut générer une présentation HTML, en laissant la solution en hyperlien cliquable.
2. Pour le papier on peut générer une présentation PDF, en affichant la solution sur une page séparée de l'énoncé.
3. Pour un usage multimédia on pourra générer une présentation dynamique, avec affichage graphique de l'énoncé, lecture audio de la question, et affichage de la solution après un temps de pause.

Notons que si l'on avait choisi une des présentations physiques, plutôt que la représentation logique, il n'aurait pas été possible de générer les autres représentations.

Un exercice mis en forme

Exemple

Soit une (mauvaise) mise en forme en HTML du même exercice :

```
1 <html>
2   <body> Soit un triangle rectangle disposant d'un angle de 30 degrés. <br />
3     <b> Donner la valeur des autres angles du triangle. </b>
4     <br/> > <a href="ex001i01.html"> Vous avez besoin d'aide ? </a>
5     <hr/>
6     <a href="ex001s01.html"> Vérifier votre réponse ! </a>
7   </body>
8 </html>
9
```



On voit que dans ce format la structure logique n'apparaît plus explicitement et qu'il n'est plus possible d'identifier l'énoncé, la question et la solution sans comprendre le contenu.

💡 Fondamental

L'exemple montre que l'on peut calculer la mise en forme à partir de la structure logique, **mais non l'inverse** (dans le cas général).

2. Les avantages de la structuration logique

- **La pérennité des contenus**

En effet en étant dissociées d'un format physique, les informations sont indépendantes de l'évolution de ces formats. Il n'est donc pas nécessaire de reproduire régulièrement les contenus pour qu'ils restent compatibles avec les différentes versions de formats, HTML ou Microsoft Word par exemple.

- **La maintenance des contenus**

La représentation logique ne nécessite pas le recours à des experts d'un langage de présentation, comme c'est le cas pour HTML par exemple, puisque le langage de représentation logique est défini par les experts en fonction du domaine.

- **La portabilité des contenus**

L'investissement consenti pour la production des contenus peut-être rentabilisé en dehors d'un système technique donné, puisque des contenus structurés logiquement sont potentiellement interopérables (à condition que cette structure soit effectivement connue et effectivement indépendante d'une structure physique). Par exemple l'information peut être diffusée sur des supports multiples (papier, Internet, mémoire mobile, etc.) ou sur des environnements matériels multiples (systèmes d'exploitation, taille d'écran, capacité de réseau, etc.).

- **La malléabilité des contenus**

La structure logique des contenus étant connue, il est possible de manipuler cette structure et ainsi faire coïncider la présentation à des contextes d'usage différents. Par exemple on pourra proposer une vision synthétique de l'information (en en sélectionnant les éléments essentiels) ou au contraire une vision exhaustive.

- **La séparation des métiers**

La séparation entre structure logique et physique permet également la séparation entre les métiers d'auteur (la production de l'information logique) et les métiers d'éditeur (la mise en forme de cette information pour un support). Cette séparation permet d'augmenter les volumes de production et de réduire les coûts, puisque qu'elle ne nécessite plus le recours à des experts pluridisciplinaires capables de maîtriser à la fois le contenu et son édition. Notons enfin que la séparation des métiers a également tendance à améliorer la qualité puisque les auteurs et les éditeurs peuvent se concentrer respectivement sur leur expertise.

3. Langages XML orientés documents

Az Définition

Ils permettent de représenter informatiquement des documents numériques. Les formats de documents faisant appel à des représentations fortement arborescentes, XML est un candidat idéal pour cet usage. En fait SGML, l'ancêtre de XML, a été inventé pour l'informatique documentaire. Aujourd'hui la très grande majorité des formats de représentation de document sont en XML.

Langages XML orienté documents formatés

Az Définition

Ils définissent des formats de mise en forme de document (structure physique), en général pour un support donné.

Langages XML orientés documents formatés

👁 Exemple

- XHTML
- XSL-FO
- SMIL
- OpenDocument
- OOXML
- ...

Langages XML orientés documents structurés

Az Définition

Ils définissent des formats de structuration de document (structure logique), en général pour un métier donné, plus ou moins précis selon la généralité du langage.

Langages XML orientés documents structurés

[Exemple](#)

- DocBook
- TEI
- DITA
- ...

Format local orienté document structuré

[Exemple](#)

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <document type="Lorem ipsum">
3   <paragraphe>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
   Pellentesque sit amet libero ac mauris egestas venenatis nec vitae sapien.
   </paragraphe>
4   <paragraphe>Donec a lectus sed augue pellentesque accumsan eu ac justo.
   Etiam est urna, sagittis ac cursus nec, placerat quis velit.</paragraphe>
5 </document>
```

Voir aussi

[+ Complément](#)

Documents structurés et documents formatés (cf. p.28)

4. Documents structurés et documents formatés

Document formaté

[Az Définition](#)

On appelle document formaté un document dont le fichier informatique source décrit la façon de le mettre en forme. C'est la façon la plus courante de traiter avec les documents informatiques, telle qu'elle est mise en œuvre dans les traitements de texte ou sur le Web avec HTML.

Un document formaté nécessite un logiciel capable d'en interpréter le format pour permettre de le lire.

XML est un excellent candidat à la *sérialisation* ^{p.22} de documents formatés.

Document formaté (format OpenDocument : extrait simplifié)

[Exemple](#)

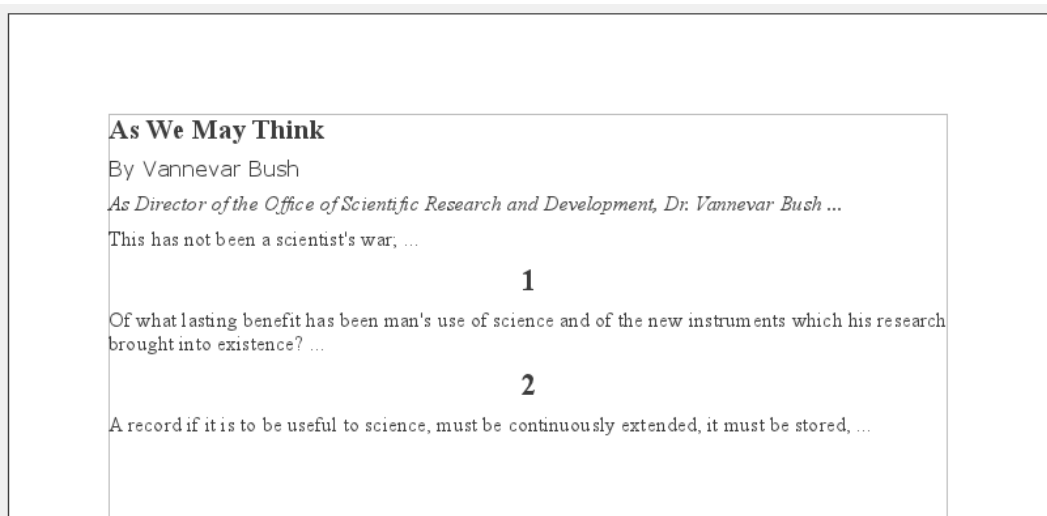
```

1 <document-content>
2   <automatic-styles>
3     <style name="P1" family="paragraph">
4       <text-properties font-name="Times New Roman" fo:font-size="12pt"/>
5     </style>
6     <style name="P2" family="paragraph">
7       <text-properties font-name="Times New Roman" fo:font-size="16pt"
fo:font-weight="bold"/>
8     </style>
9     <style name="P3" family="paragraph">
10      <paragraph-properties fo:text-align="center"/>
```

```

11         <text-properties font-name="Times New Roman" fo:font-size="18pt"
fo:font-weight="bold"/>
12     </style>
13     <style name="P4" family="paragraph">
14         <text-properties font-name="Times New Roman" fo:font-size="12pt"
fo:font-style="italic"
15         />
16     </style>
17     <style name="T1" family="text">
18         <text-properties font-name="Verdana"/>
19     </style>
20 </automatic-styles>
21 <body>
22     <text>
23         <p style-name="P2">As We May Think</p>
24         <p style-name="P1">
25             <span style-name="T1">By Vannevar Bush</span>
26         </p>
27         <p style-name="P4">As Director of the Office of Scientific Research
and Development, Dr.
28             Vannevar Bush ...</p>
29         <p style-name="P1">This has not been a scientist's war ...</p>
30         <p style-name="P3">1</p>
31         <p style-name="P1">Of what lasting benefit has been man's use of
science ...</p>
32         <p style-name="P3">2</p>
33         <p style-name="P1">A record if it is to be useful to science ...</p>
34     </text>
35 </body>
36 </document-content>

```



Visualisation dans OpenOffice.org Writer de l'extrait de l'article "As We May Think"

Notion de documents structurés

Az Définition

On appelle document structuré un document dont la structure logique est décrite plutôt que la mise en forme physique (*Structured documents* André⁸⁹ p.23, p.7).

Après SGML qui avait été inventé pour cela, XML est aujourd'hui le candidat quasi-unique pour la réalisation de documents structurés.

Document structuré (format DocBook, légèrement simplifiée)

👁 Exemple

```

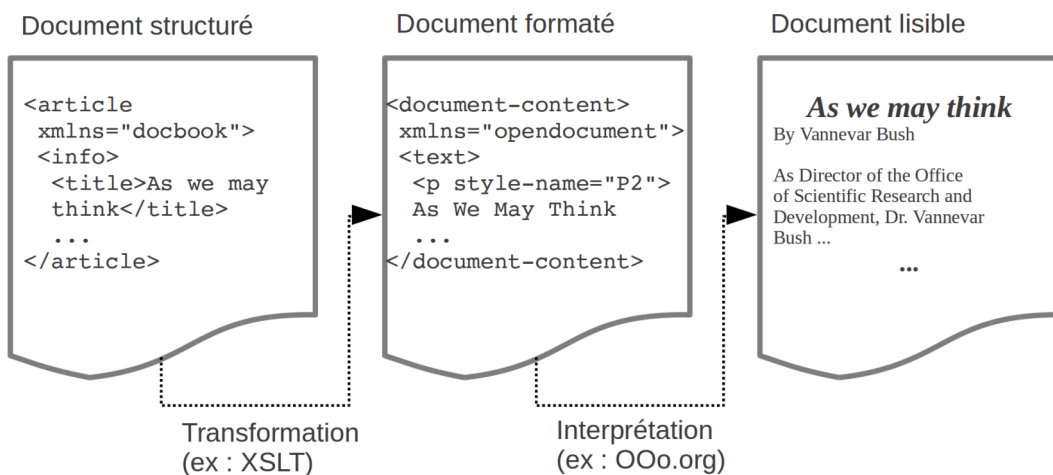
1 <article xmlns="http://docbook.org/ns/docbook">
2   <info>
3     <title>As we may think</title>
4     <author>
5       <personname>Vannevar Bush</personname>
6     </author>
7   </info>
8   <epigraph>
9     <para>As Director of the Office of Scientific Research and Development,
10    Dr. Vannevar Bush
11     ...</para>
12   </epigraph>
13   <para>This has not been a scientist's war ...</para>
14   <sect1>
15     <para>Of what lasting benefit has been man's use of science ...</para>
16   </sect1>
17   <sect1>
18     <para>A record if it is to be useful to science ...</para>
19 </sect1>
20 </article>

```

Transformation des documents structurés

Az Définition

Un document structuré n'est pas destiné à être directement utilisé pour la lecture humaine, il doit être d'abord transformé dans un document formaté pour être utilisé (visualisé, imprimé, ...).



Processus de transformation d'un document structuré en document formaté

Chaîne XML complète

💬 Remarque

La technologie XML peut en fait être utilisée tout au long de la chaîne :

- Langage XML orienté document structuré en amont (DocBook, DITA, TEI, langage local, ...)
- Langage XML de programmation pour la transformation (XSL-XSLT)
- Langage XML orienté document formaté pour l'aval (OpenDocument, XHTML, ...)

Voir aussi

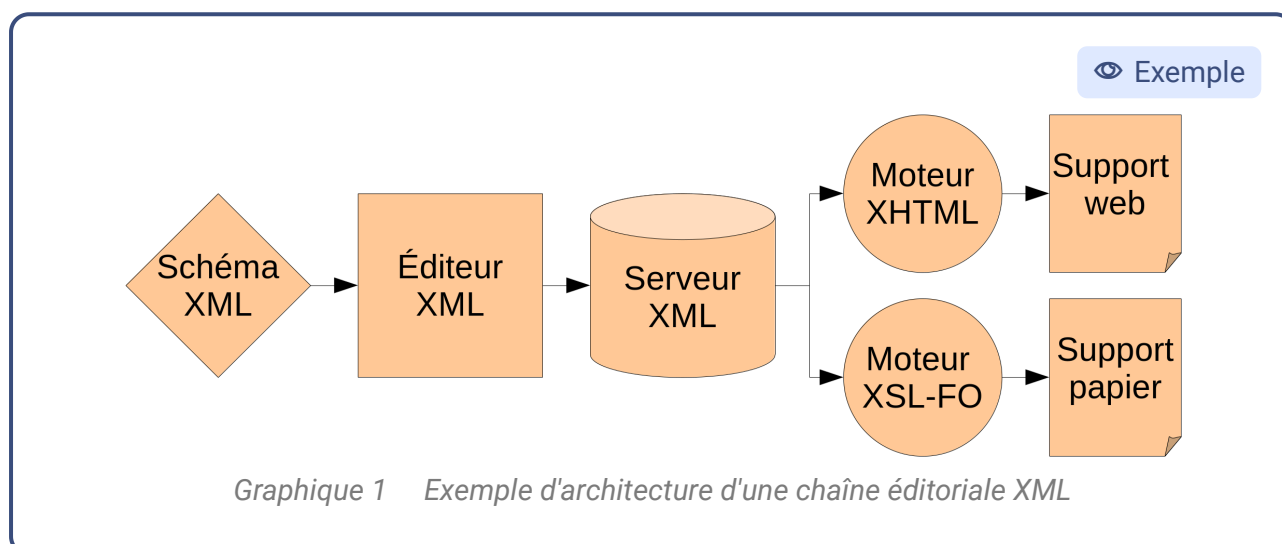
⊕ Complément

La structuration logique (cf. p.4)*Exemple de structuration logique* (cf. p.25)*Architecture des chaînes éditoriales XML* (cf. p.31)*Un langage pour publier les documents XML* (cf. p.32)*Définition de XSL-XSLT* (cf. p.32)**Bibliographie**

⊕ Complément

Technologie de la GED Dupoirier, 1995 p.23 : Structure logique et structure physique (pp58-61)

5. Architecture des chaînes éditoriales XML



L'approche classique pour réaliser une chaîne éditoriale XML est la suivante :

1. Formalisation du schéma documentaire, avec un langage de modélisation de schéma XML (XML Schema, Relax NG, etc.)
2. Utilisation d'un éditeur XML standard et stylage de cet éditeur (il existe de très nombreux éditeurs XML, plus ou moins graphiques, qui se paramètrent automatiquement lorsqu'on leur fournit un schéma : Oxygen, XMetal, Arbortext, etc.).
3. Utilisation de serveurs de fichiers XML pour la gestion centralisée des contenus.
4. Réalisation de moteurs de transformation (par exemple avec les technologies XSL-XSLT combinées avec des langages de rendu comme HTML).

L'ensemble est en général intégré avec un langage applicatif tiers (Java, PHP, etc.).

Implémentations

 Méthode

On peut distinguer trois grandes modalités pour implémenter une chaîne éditoriale :

- L'implémentation « sur mesure » consiste à mobiliser différentes briques logicielles et à programmer un logiciel spécifique pour un besoin particulier. Par exemple en couplant un éditeur du marché avec un outil de stockage et en réalisant des moteurs de publication sous la forme de feuilles XSL-XSLT.
- L'implémentation « modèle dédié » consiste à implémenter en dur un modèle de chaîne éditoriale pour un besoin assez transversal, tel que la publication de livres ou de modules de cours universitaires.
- L'implémentation « générique » consiste à réaliser un système paramétrable, indépendant d'un modèle en particulier, et à le configurer en fonction des besoins. Il s'agit alors de ce que l'on peut appeler un Système de Gestion de Chaînes Éditoriale par analogie aux Systèmes de Gestion de Bases de Donnée

6. Un langage pour publier les documents XML

XML lorsqu'il est utilisé pour définir des formats documentaire métier est un format de **représentation** de l'information, et non un format de **publication** (de présentation) de cette information : donc un tel fichier XML **n'est pas utilisable tel que par un lecteur**.

XML ne peut donc être utilisé pour des langages abstrait que si l'on est capable de transformer les documents sources en document publiés lisibles grâce à un format de présentation : HTML par exemple dans le cas de publication Web, ou PDF pour l'impression.

7. Définition de XSL-XSLT

XSL-XSLT

 Az Définition

XSL-XSLT est une partie du standard W3C XSL qui a trait à la transformation des documents XML (l'autre partie étant XSL-FO).

XSL-XSLT est un langage de programmation déclaratif écrit en XML (un programme XSL-XSLT est un document XML).

- XSL-XSLT est langage de manipulation de document XML (fondé sur XPath et sur le modèle arborescent de représentation des documents XML)
- XSI-XSLT est utilisé pour transformer un document XML source dans un autre format, typiquement HTML, mais aussi tout autre format codé sur des caractères dont la syntaxe est connue.
- XSL-XSLT est aussi utilisé pour faire des changements de schéma XML (export d'un XML vers un autre XML structuré différemment) par exemple pour échanger des données selon un standard.

XSL-XSLT, XSL-FO, XSLT, XSL, FO

Remarque

On parle souvent (par simplification) de XSL ou de XSLT pour désigner XSL-XSLT et de FO pour désigner XSL-FO.

XSL utilisé seul désigne donc par convention XSL-XST (et non XSL-FO).

8. Introduction générale à HTML

Rappel

HTML est l'un des trois standards qui ont permis de fonder le Web :

- HTTP est le standard qui permet de transférer des pages web depuis un serveur web vers un client web (navigateur web) ;
- URL (ou adresse web) est le standard qui permet d'identifier de façon unique une page web dans le monde ;
- HTML est le standard de formater les pages web.

HTML est un standard du W3C

Rappel

- Le W3C est un consortium international composé notamment d'industriels du domaine de l'informatique (IBM, Microsoft...), des éditeurs de navigateurs (Mozilla, Google...) et de centres de recherche (MIT, INRIA...).
- Son rôle est de standardiser les technologies utilisées dans le domaine du Web : HTML, CSS, JavaScript, mais aussi XML, DOM, PNG, SVG...
- Le W3C a été fondé en 1994 par Tim Berners-Lee au MIT avec le soutien du département de la défense états-unienne. L'INRIA rejoint le W3C en 1995. En octobre 2021 le W3C compte 452 membres.

Une petite histoire d'HTML

Rappel

- 1990 : La première page HTML est publiée sur le premier serveur web au Cern.
- 1991-1994 : HTML est développé au sein des navigateurs NCSA Mosaic puis Netscape Navigator sans être standardisé.
- 1995 : Le W3C spécifie le langage HTML 2.0 via la RFC 1866.
- 1996-2014 : Le W3C fait évoluer régulièrement le standard : HTML 3.2, HTML 4.0, XHTML 1.1.
- 2014 : Le W3C publie HTML5.

HTML5

Az Définition

Le HyperText Markup Language, abrégé HTML, **HTML5** dans sa dernière version, est le langage de **balisage** conçu pour écrire les pages **web**.

HTML5 permet :

- d'écrire du texte et des liens entre des pages (hypertexte),
- d'inclure des ressources multimédias (images, des vidéos...),
- de structurer la page (balisage sémantique),
- de mettre en forme le contenu (balisage physique avec CSS),
- de programmer des interactions avec l'utilisateur (par exemple des formulaires de saisie, avec JavaScript).

(Source modifiée : Wikipédia²)

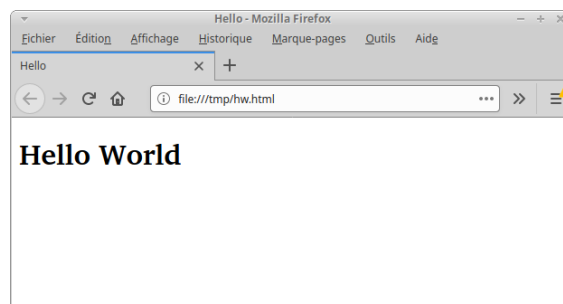
Hello World

Exemple

```

1 <!DOCTYPE html>
2 <html lang="en-US">
3   <head>
4     <meta charset="utf-8">
5     <title>Hello</title>
6   </head>
7   <body>
8     <h1>Hello World</h1>
9   </body>
10 </html>

```



Créer un document HTML

Méthode

Un document HTML est un simple fichier texte, il peut donc être créé avec n'importe quel éditeur de texte.

On utilise a minima des éditeurs colorisant comme :

- *notepad++* sous Windows
- *gedit* sous Linux

² https://fr.wikipedia.org/wiki/Hypertext_Markup_Language

On préfère des éditeurs colorisant et validant comme :


- <https://atom.io> (libre)
- <https://code.visualstudio.com> (partiellement propriétaire, collecte de données par défaut, désactivable)

Consulter un document HTML

 Méthode

- Un document HTML se consulte avec un navigateur comme Firefox ou Chrome à l'aide du menu permettant d'ouvrir un fichier (par exemple CTRL+O avec Firefox).
- On peut toujours consulter le code source d'un fichier HTML en cours de consultation que ce soit un fichier local ou un fichier en ligne (par exemple clic droit sur la page avec Firefox).

Outils d'édition HTML en ligne

 Complément

- replit.com
- jsfiddle.net
- codepen.io/pen³
- htmledit.squarefree.com⁴

³. <https://codepen.io/pen/>

⁴. <https://htmledit.squarefree.com/>

