

Introduction au HTML5



NSI Première

Académie de Versailles

M. Boehm, C. Poulmaire, P. Remy

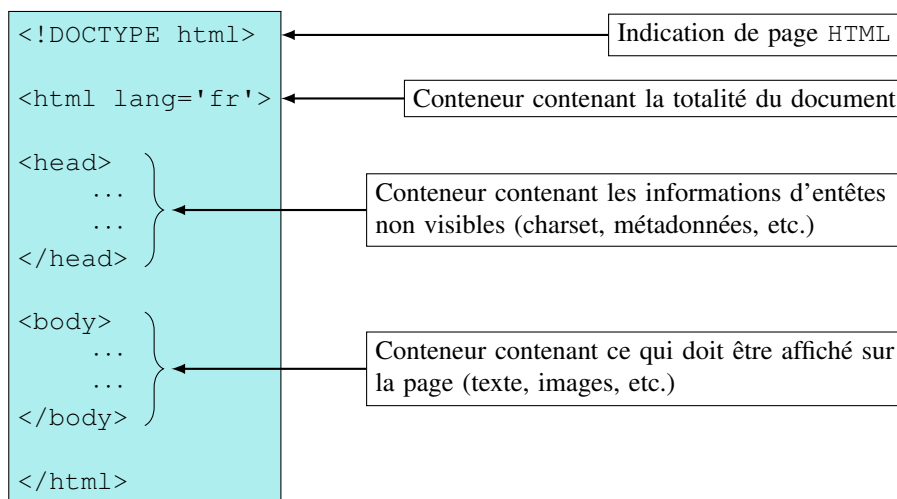
Sommaire

1	Structure générale d'une page HTML5	2
2	Balises d'entête	2
3	Balises structurantes	3
3.1	Quelques règles générales sur les balises structurantes	3
3.2	Balises structurantes du texte	4
3.2.1	Titres et paragraphes	4
3.2.2	Mise en forme du texte	4
3.2.2.1	Exposants et indices.	4
3.2.2.2	Écritures spécifiques.	4
3.2.2.3	Citations.	5
3.2.2.4	Mises en forme particulières.	5
3.2.3	Sauts de ligne	6
3.2.4	Liens hypertextes	6
3.2.4.1	Liens hypertextes internes.	6
3.2.4.2	Liens hypertextes externes.	7
3.2.5	Images, figures, musiques et vidéos	7
3.2.5.1	Insertion d'images.	7
3.2.5.2	Insertion de figures.	7
3.2.5.3	Insertion de musiques.	8
3.2.5.4	Insertion de vidéos.	8
3.3	Balises de listes	9
3.4	Balises de tables	10
3.5	Balises de formulaires	10
3.6	Balises sectionnantes	11
3.7	Balises génériques	12
4	Validation d'une page HTML	12

Le langage HTML (= HyperText Markup Language) a été inventé en 1991 par Tim Berners-Lee. Son principe général est très simple : le texte (= *le contenu*) est structuré par des *balises* qui définissent la structure (= *le contenant*). De plus, c'est un langage *hypertexte*. Cela signifie qu'il est possible de définir des liens entre plusieurs documents ou au sein d'un même document. Les documents pointés par les liens peuvent se situer sur des machines éloignées faisant partie d'internet.

1 Structure générale d'une page HTML5

Toutes les pages HTML5 utilisent la même structure générale de base :



- ★ Le conteneur `<html>` contient la totalité du document HTML. Il admet plusieurs attributs dont l'un, appelé `lang`, est fortement recommandé et permet de spécifier la langue du document (ici le français).
- ★ Le conteneur `<head>` contient des informations n'apparaissant pas sur la page HTML, mais la décrivant. Ces informations sont utilisées par exemple par les moteurs de recherche pour indexer les pages web. Ce conteneur inclut diverses balises appelées *balises d'entête* (voir paragraphe 2 ci-dessous).
- ★ Le conteneur `<body>` contient ce qui doit être affiché : texte, images, sons, etc... et est structuré par différents types de balises appelées *balises structurantes* (voir paragraphe 3).

2 Balises d'entête

Les *balises d'entêtes* sont les balises situées dans le conteneur `<head>` du document HTML. Il en existe de nombreuses dont les plus couramment utilisées sont les suivantes :

- ★ `<meta>` qui permet de définir les métadonnées de la page (encodage, mots-clés, etc.);
- ★ `<title>` qui permet de définir le titre apparaissant dans l'onglet de navigation du navigateur;
- ★ `<link>` qui permet de réaliser des liens, notamment vers des fichiers de style pour la mise en forme.

La structure minimale d'un document HTML doit définir obligatoirement le titre apparaissant dans l'onglet de navigation ainsi que l'encodage utilisé (nous utiliserons dans la suite l'encodage UTF-8). On a ainsi le code minimal suivant :

```
<!DOCTYPE html>
<html lang='fr'>
  <head>
    <meta charset='UTF-8'>
    <title>Titre pour l'onglet de navigation</title>
  </head>
  <body>
    Texte à afficher sur la page web
  </body>
</html>
```

Noter que l'encodage est défini ici par le mot-clé `charset` et non `encoding` comme dans de nombreux langages de programmation.

3 Balises structurantes

Les balises décrites dans ce paragraphe sont toutes situées dans le conteneur `<body>` du document. Elles sont regroupées en différentes catégories suivant qu'elles permettent de structurer du texte, réaliser des listes, des liens, des tableaux, des formulaires, etc...

3.1 Quelques règles générales sur les balises structurantes

- ★ La plupart des balises modifient l'interprétation que fait le navigateur sur leur contenu. Puisque les balises délimitent du contenu, elles sont faites d'une *partie ouvrante*, obtenue par l'instruction `<nom_balise>`, et d'une *partie fermante*, obtenue par l'instruction `</nom_balise>`.

Exemple 3.1 Pour écrire un titre de niveau 1, on utilise la balise `h1`.

Ainsi, pour écrire *Ceci est un titre*, on tape `<h1>Ceci est un titre</h1>`.

- ★ Toutes les balises ne sont pas destinées à avoir du contenu. Certaines ne sont que de simples marqueurs. Ces balises ne nécessitent donc pas de fermeture et sont obtenues à l'aide de l'instruction `<nom_balise>`.

Exemple 3.2 La balise `br` permettant d'effectuer un retour à la ligne n'est pas une balise destinée à recevoir du texte. Pour l'utiliser, on tape donc `
`.

- ★ Les balises sont regroupées en deux catégories –le type « bloc » et le type « en ligne »– suivant leur comportement par défaut vis-à-vis des autres éléments lors de l'affichage de leur contenu sur une page web. Les balises de type « bloc » créent un retour à la ligne avant et après et occupent par défaut la totalité de la largeur de la page. En revanche, les balises de type « en ligne » ne créent pas de retour chariot avant et après et affichent leur contenu dans le flux sans modifier le reste de l'affichage. Ces deux types de comportement peuvent être modifiés à l'aide d'une feuille de style CSS.

Exemple 3.3 La balise `<h1>` est une balise de type « bloc » et la balise `
` est une balise de type « en ligne ».

- ★ Chaque balise a également un comportement par défaut vis-à-vis du navigateur qui les interprète (et peut varier suivant le type de navigateur) en terme de police, couleur, graphisme, etc... Ce comportement peut être modifié à l'aide d'attributs sur lesquels on agit à partir de fichiers de style.
- ★ Toutes les balises admettent deux attributs génériques : l'identifiant `id` et la classe `class`.
 - Pour définir un identifiant, on utilise la syntaxe

```
<nom_balise id='nom_identifiant'>...</nom_balise>
```

Un identifiant est unique et ne peut être utilisé qu'une seule fois dans un document HTML. Il est très utile pour la création de liens internes et de formulaires ainsi que pour l'utilisation de fichiers de style.

- Pour définir une classe, on utilise la syntaxe

```
<nom_balise class='nom_classe'>...</nom_balise>
```

Contrairement à un identifiant, une classe peut être réutilisée plusieurs fois à l'intérieur d'un même document HTML et peut même regrouper plusieurs types de balises différentes. Elle est très utilisée lors de l'utilisation de fichiers de style.

Exemple 3.4

- L'instruction `<h1 id='Titre'>` attribue l'identifiant `Titre` à la balise `<h1>`.
- Les instructions `<h1 class='important'>` et `<p class='important'>` permettent d'attribuer dans le même document HTML la même classe `important` aux deux balises `<h1>` et `<p>`.

3.2 Balises structurantes du texte

3.2.1 Titres et paragraphes

- * `<h1>...</h1>` : définit un titre de niveau 1 (type « bloc »).
- * `<h2>...</h2>` : définit un titre de niveau 2 (type « bloc »).
- * `<h3>...</h3>` : définit un titre de niveau 3 (type « bloc »).
- * `<h4>...</h4>` : définit un titre de niveau 4 (type « bloc »).
- * `<h5>...</h5>` : définit un titre de niveau 5 (type « bloc »).
- * `<h6>...</h6>` : définit un titre de niveau 6 (type « bloc »).
- * `<p>...</p>` : définit un paragraphe (type « bloc »).

Exemple 3.5 Un exemple de création de trois titres de niveaux différents et d'un paragraphe :

```
<h1>Titre de niveau 1</h1>
<h2>Titre de niveau 2</h2>
<h3>Titre de niveau 3</h3>
<p>Ceci est un paragraphe</p>
```

On a le rendu suivant à l'écran :

Titre de niveau 1

Titre de niveau 2

Titre de niveau 3

Ceci est un paragraphe

3.2.2 Mise en forme du texte

3.2.2.1 Exposants et indices.

- * `^{...}` : affiche du texte en exposant (type « en ligne »).
- * `_{...}` : affiche du texte en indice (type « en ligne »).

Exemple 3.6 Un exemple d'écriture en exposant et en indice :

```
<p>Texte écrit<sup>en exposant</sup>.</p>
<p>Texte écrit<sub>en indice</sub>.</p>
```

On a le rendu suivant à l'écran

Texte écrit^{en exposant}.

Texte écrit_{en indice}.

3.2.2.2 Ecritures spécifiques.

- * `...` : affiche du texte en valeur ajoutée (généralement en italique ; type « en ligne »).
- * `...` : affiche du texte en valeur forte (généralement en gras ; type « en ligne »).
- * `<code>...</code>` : affiche du texte en mode typographique (utile pour les programmes ; type « en ligne »).

Exemple 3.7 Un exemple d'écriture de textes en valeur ajoutée et en valeur forte :

```
<p>Texte en valeur ajoutée : <em>texte</em>.</p>
<p>Texte en valeur forte : <strong>texte</strong>.</p>
<p>Texte typographique : <code>texte</code>.</p>
```

On a le rendu suivant à l'écran :

Texte en valeur ajoutée : *texte*.

Texte en valeur forte : **texte**.

Texte typographique : `texte`.

3.2.2.3 Citations.

- ★ `<cite>...</cite>` : permet de citer le titre d'une œuvre (généralement en italique ; type « en ligne »).
- ★ `<q>...</q>` : permet de faire une citation courte en incise (type « en ligne »).
- ★ `<blockquote>...</blockquote>` : permet de faire une citation longue (type « bloc »).

Exemple 3.8 Un exemple d'écriture de citations :

```
<p><cite>Notre-Dame de Paris</cite>, Victor Hugo.</p>
<p><q>Etre ou ne pas être</q>, William Shakespeare.</p>
<blockquote>Début de la citation [...] fin de la citation</blockquote>
```

On a le rendu suivant à l'écran :

```
Notre-Dame de Paris, Victor Hugo.

"Etre ou ne pas être", William Shakespeare.

Début de la citation [...] fin de la citation
```

3.2.2.4 Mises en forme particulières.

★ Respecter un contenu pré-formaté.

- `<pre>...</pre>` : permet de respecter le format du contenu (type « bloc »).

Exemple 3.9 Un exemple de respect de contenu pré-formaté :

```
<pre>Avec cette balise, on peut faire des espaces      et
passer directement à la ligne.</pre>
```

On a le rendu suivant à l'écran :

```
Avec cette balise, on peut faire des espaces      et
passer directement à la ligne.
```

★ Définir des informations de contact.

- `<address>...</address>` : permet de définir des informations de contact (type « bloc »).

Exemple 3.10 Un exemple d'informations de contact :

```
<address>Lycée Les Pierres Vives<br>
Tél : 01 39 14 07 00<br>
1, rue des Alouettes<br>
78420 Carrières-sur-Seine</address>
```

On a le rendu suivant à l'écran :

```
Lycée Les Pierres Vives
Tél : 01 39 14 07 00
1, rue des Alouettes
78420 Carrières-sur-Seine
```

★ Définir une date.

- `<time>...</time>` : permet de représenter une période donnée (type « en ligne »). Cette balise est généralement utilisée avec l'attribut `datetime` afin de traduire la date ou l'instant dans un format informatique pour permettre aux moteurs de recherche d'exploiter ces données ou de créer des rappels.

Exemple 3.11 Un exemple de création de date :

```
<p>Le concert commence à <time datetime='2018-07-07T20:00:00'>20h00</time>.</p>
```

On a le rendu suivant à l'écran :

```
Le concert commence à 20h00.
```

★ Définir des barres de progression et de jauge.

- `<progress>...</progress>` : définit une barre indiquant l'état de complétion d'une tâche (type « en ligne »). Cette balise admet deux attributs `value` et `max` prenant des valeurs décimales positives qui indiquent respectivement la valeur actuelle de la barre de progression et la valeur maximale de la barre.
- `<meter>...</meter>` : définit une jauge indiquant un état, stable ou évoluant au cours du temps (type « en ligne »). Cette balise admet trois attributs `value`, `min` et `max` prenant des valeurs décimales qui indiquent respectivement la valeur actuelle de la jauge, sa valeur minimale et sa valeur maximale.

Exemple 3.12 Un exemple de création de barre de progression et de jauge :

```
<p>Progression : <progress value=70 max=100>70%</progress></p>
<p>Nombre de personnes : <meter min=0 max=100 value=50>50 places</meter></p>
```

On a le rendu suivant à l'écran :



Les textes indiquées entre les balises ouvrantes et fermantes sont des textes alternatifs s'affichant si le navigateur ne prend pas en charge ces éléments.

Les deux balises `<progress>` et `<meter>`, qui peuvent être également intégrées dans des formulaires, sont intéressantes lorsqu'elles sont combinées avec du JavaScript.

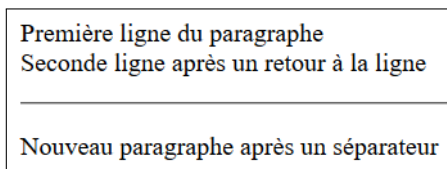
3.2.3 Sauts de ligne

- ★ `
` : effectue un retour à la ligne (type « en ligne »).
- ★ `<hr>` : trace une ligne horizontale (type « bloc »).

Exemple 3.13 Un exemple d'utilisation des sauts de ligne :

```
<p>Première ligne du paragraphe<br>
Seconde ligne après un retour à la ligne</p>
<hr>
<p>Nouveau paragraphe après un séparateur</p>
```

On a le rendu suivant à l'écran :



3.2.4 Liens hypertextes

- ★ `texte du lien` : définit un lien hypertexte de nom `texte du lien` permettant, lorsque l'on clique dessus, d'être redirigé sur la cible référencée dans l'attribut obligatoire `href` (type « en ligne »). Le rendu à l'écran est le suivant :

[texte du lien](#)

On distingue plusieurs types de liens : les liens hypertextes internes et les liens hypertextes externes.

3.2.4.1 Liens hypertextes internes. Dans un lien hypertexte interne, la cible fait référence à un identifiant d'une balise du document HTML et, lorsque l'on clique sur le lien, le haut de la page web se positionne au niveau de cette balise. Pour définir un tel lien, on utilise la syntaxe suivante dans l'attribut `href` :

```
href='#nom_identifiant'
```

Exemple 3.14 Le code suivant définit un identifiant pour une balise `h1` et le lien `Retour au début`, situé plus bas dans la page, permet de repositionner la page web au niveau de ce titre lorsque l'on clique dessus :

```
<h1 id='deb'>Début</h1>
...
<a href='#deb'>Retour au début</a>
```

3.2.4.2 Liens hypertextes externes. Dans un lien hypertexte externe, la cible est une url (*Uniform Resource Locator*) faisant référence à une page web, un site web, un mail, etc... et, lorsque l'on clique sur le lien, on est redirigé vers cette url. Pour définir un tel lien, on utilise la syntaxe suivante dans l'attribut `href` :

```
href='url'
```

- ★ Si la page à laquelle on accède via le lien est située dans le même site web que la page du lien, on peut simplement donner le chemin relatif par rapport à la page initiale. Par exemple, si la page à laquelle on veut accéder s'appelle `page2.html` et est située dans le dossier `data` situé lui-même dans le dossier courant de la page initiale, alors on saisit le code :

```
href='data/page2.html'
```

- ★ Si la page à laquelle on accède via le lien est un autre site web ou une page spécifique d'un autre site web, on indique l'url complet. Par exemple, si on veut accéder au site `www.MonSite.fr` via notre lien, alors on saisit le code :

```
href='www.MonSite.fr'
```

Remarque 3.15 On peut combiner un lien hypertexte externe avec un lien hypertexte interne de façon à pouvoir accéder directement à un élément particulier d'une autre page. Pour cela, on utilise la syntaxe suivante :

```
href='url_de_la_page#nom_identifiant'
```

- ★ Pour créer un lien vers un email, on utilise la syntaxe suivante :

```
href='mailto:'
```

A la suite de `mailto:`, on peut également préciser l'adresse email, les copies `cc` (= copie carbone) et `cc:` (= copie carbone invisible), le sujet, etc... On consultera internet pour plus de détails.

Remarque 3.16 Il existe également d'autres types d'url que l'on peut passer à l'attribut `href`. En fait, il s'agit des url utilisant n'importe quel protocole supporté par le navigateur. On consultera internet pour plus de détails.

3.2.5 Images, figures, musiques et vidéos

3.2.5.1 Insertion d'images.

- ★ `` : permet de représenter une image dans un document HTML (type « en ligne »). L'attribut `src` est obligatoire et contient le chemin (à partir du fichier HTML courant) de l'image qu'on souhaite afficher; l'attribut `alt` n'est pas obligatoire mais fortement recommandé et contient une description textuelle de l'image qui sera affichée si l'image ne peut être chargée.

Exemple 3.17 Pour représenter l'image `MonImage.png` située dans le dossier courant avec la description textuelle `une image`, on saisit le code :

```
<img src='MonImage.png' alt='une image'>
```

Remarque 3.18 La balise `` doit obligatoirement être placée à l'intérieur d'un paragraphe, c'est-à-dire entre les balises `<p>` et `</p>`, ou dans une *figure* si elle vient enrichir le texte pour compléter les informations de la page (*i.e.*, ne pas être une simple illustration).

3.2.5.2 Insertion de figures.

- ★ `<figure>...</figure>` : permet de représenter un contenu indépendant, habituellement accompagné d'une légende grâce à la balise `<figcaption>...</figcaption>` (type « bloc »).

Exemple 3.19 Pour représenter l'image `MonImage.png` située dans le dossier courant avec la description textuelle `une image` et la légende `Légende de l'image`, on saisit le code :

```
<figure>
  <img src='MonImage.png' alt='une image'>
  <figcaption>Légende de l'image</figcaption>
</figure>
```

Remarque 3.20 Les figures peuvent contenir plusieurs images. Par exemple, le code


```
<figure>
  <img src='Ed.png' alt='Logo Edge'>
  <img src='Firefox.png' alt='Logo Mozilla Firefox'>
  <img src='Chrome.png' alt='Logo Google Chrome'>
  <figcaption>Logos de différents navigateurs</figcaption>
</figure>
```

permet d'afficher trois images de différents logos de navigateurs internet.

Remarque 3.21 Les figures peuvent contenir autre chose que des images comme, par exemple, des codes sources, des citations, des vidéos, etc... On consultera internet pour plus de détails.

3.2.5.3 Insertion de musiques.

- ★ `<audio src='...' controls>...</audio>` : permet d'intégrer un contenu sonore dans un document HTML (type « en ligne » si l'attribut `controls` est indiqué). L'attribut `src` contient le chemin (à partir du fichier HTML courant) du contenu sonore; l'attribut `controls` est optionnel et permet d'ajouter sur la page web les boutons « Lecture », « Pause » et la barre de défilement. Entre les balises `<audio>` et `</audio>`, on peut rajouter un texte alternatif qui sera affiché si le navigateur ne prend pas en charge le contenu.

Exemple 3.22 Pour intégrer la musique `MaMusique.mp3` située dans le dossier courant avec le texte alternatif Mettez à jour votre navigateur, on saisit le code :

```
<audio src='MaMusique.mp3' controls><p>Mettez à jour votre navigateur</p></audio>
```

Remarque 3.23 La balise `<audio>` admet plusieurs attributs optionnels parmi lesquels :

- ★ `autoplay` : la musique est jouée dès le chargement de la page web,
- ★ `loop` : la musique est jouée en boucle.

Remarque 3.24 La balise `<source src='...' type='...'>` permet de définir plusieurs formats audio possibles, le navigateur sélectionnant la première ressource dont il prend en charge le format. On a par exemple le code suivant :

```
<audio controls>
  <source src='MaMusique.mp3' type='audio/mp3'>
  <source src='MaMusique.wav' type='audio/wav'>
  <p>Mettez à jour votre navigateur</p>
</audio>
```

3.2.5.4 Insertion de vidéos.

- ★ `<video src='...' controls>...</video>` : permet d'intégrer un contenu vidéo dans un document HTML (type « en ligne » si l'attribut `controls` est indiqué). L'attribut `src` contient le chemin (à partir du fichier HTML courant) du contenu vidéo; l'attribut `controls` est optionnel et permet d'ajouter sur la page web les boutons « Lecture », « Pause » et la barre de défilement. Entre les balises `<video>` et `</video>`, on peut rajouter un texte alternatif qui sera affiché si le navigateur ne prend pas en charge le contenu.

Exemple 3.25 Pour intégrer la vidéo `MaVideo.mp4` située dans le dossier courant avec le texte alternatif Mettez à jour votre navigateur, on saisit le code :

```
<video src='MaVideo.mp4' controls><p>Mettez à jour votre navigateur</p></video>
```

Remarque 3.26 La balise `<video>` admet plusieurs attributs optionnels parmi lesquels :

- ★ `autoplay` : la vidéo est lancée dès le chargement de la page web,
- ★ `loop` : la vidéo est lancée en boucle,
- ★ `width` : permet de modifier la largeur de la vidéo (en pixels),
- ★ `height` : permet de modifier la hauteur de la vidéo (en pixels).

Remarque 3.27 La balise `<source src='...' type='... '>` permet de définir plusieurs formats vidéo possibles, le navigateur sélectionnant la première ressource dont il prend en charge le format. On a par exemple le code suivant :

```
<video controls>
  <source src='Mavideo.mp4' type='video/mp4'>
  <source src='MaVideo.webm' type='video/webm'>
  <p>Mettez à jour votre navigateur</p>
</video>
```

3.3 Balises de listes

★ Listes ordonnées et non ordonnées.

- `...` : définit une liste d'éléments ordonnés généralement affichés avec un indicateur ordinal (type « bloc »).
- `...` : définit une liste d'éléments non ordonnés généralement affichés avec une liste à puces (type « bloc »).
- `...` : définit un élément d'une liste ordonnée ou non ordonnée (type « bloc »). Il doit avoir pour parent une balise `` ou `` et peut également les avoir pour enfant (listes imbriquées).

Exemple 3.28 Un exemple de création d'une liste imbriquée avec une liste ordonnée contenant une liste non ordonnée :

```
<ol>
  <li>Premier item</li>
  <li>Second item avec ouverture d'une liste non numérotée :
    <ul>
      <li>Premier sous-item</li>
      <li>Second sous-item</li>
    </ul>
  </li>
</ol>
```

On a le rendu suivant à l'écran :

```
1. Premier item
2. Second item avec ouverture d'une liste non numérotée :
   ○ Premier sous-item
   ○ Second sous-item
```

★ Listes de descriptions.

- `<dl>...</dl>` : définit une liste de descriptions sous la forme d'une liste de paires associant des termes fournis par des éléments `<dt>` et leurs descriptions ou définitions fournies par des éléments `<dd>` (type « bloc »). Une telle liste est intéressante pour implémenter, par exemple, un glossaire.
- `<dt>...</dt>` : permet d'identifier un terme dans une liste de descriptions (type « bloc »). Il est obligatoirement un enfant de `<dl>` et est généralement suivi d'un élément `<dd>`.
- `<dd>...</dd>` : permet de donner la définition d'un terme au sein d'une liste de définitions (type « bloc »). Il est obligatoirement précédé de `<dt>`.

Exemple 3.29 Un exemple de création d'une liste de descriptions :

```
<dl>
  <dt>Premier terme</dt>
  <dd>Définition du premier terme</dd>
  <dt>Second terme</dt>
  <dd>Définition du second terme</dd>
  <dd>Informations complémentaires</dd>
</dl>
```

On a le rendu suivant à l'écran :

Premier terme
Définition du premier terme
Second terme
Définition du second terme
Informations complémentaires

3.4 Balises de tables

- * `<table>...</table>` : permet de représenter un tableau de données, c'est-à-dire des informations exprimées sur un tableau en deux dimensions (type « bloc »).
- * `<caption>...</caption>` : définit le titre du tableau. Il doit être le premier élément parmi les descendants de l'élément `<table>`.
- * `<tr>...</tr>` : définit une ligne de cellules dans un tableau. Une ligne peut être constituée d'éléments `<th>` ou `<td>`.
- * `<th>...</th>` : définit une cellule d'un tableau comme une cellule d'entête.
- * `<td>...</td>` : définit une cellule d'un tableau qui contient des données.

Remarque 3.30 Les éléments `<th>` et `<td>` admettent plusieurs attributs parmi lesquels :

- * `colspan` : entier indiquant le nombre de colonnes sur lesquelles s'étend la cellule.
- * `rowspan` : entier indiquant le nombre de lignes sur lesquelles s'étend la cellule.

Exemple 3.31 Un exemple de création de tableau :

```
<table>
  <caption>Titre du tableau</caption>
  <tr>
    <th>Entête 1</th>
    <th colspan=2>Entête 2 sur 2 colonnes</th>
  </tr>
  <tr>
    <td>Donnée 1-1</td>
    <td>Donnée 1-2</td>
    <td>Donnée 1-3</td>
  </tr>
  <tr>
    <td>Donnée 2-1</td>
    <td colspan=2>Donnée 2-2/3</td>
  </tr>
</table>
```

On a le rendu suivant à l'écran :

Titre du tableau		
Entête 1	Entête 2 sur 2 colonnes	
Donnée 1-1	Donnée 1-2	Donnée 1-3
Donnée 2-1	Donnée 2-2/3	

Remarque 3.32 Pour des tableaux importants, on peut également utiliser les balises `<thead>`, `<tbody>` et `<tfoot>` qui permettent une meilleure structuration. On consultera internet pour plus de détails.

3.5 Balises de formulaires

Voir le document **Introduction aux formulaires HTML5**.

3.6 Balises sectionnantes

- ★ `<header>...</header>` : représente un groupe de contenu introductif ou de contenu aidant à la navigation. Il peut contenir des éléments de titre, mais aussi d'autres éléments tels qu'un logo, un formulaire de recherche, etc... (type « bloc »).

Exemple 3.33 Un exemple d'utilisation de `header` :

```
<header>
  <h1>Titre introductif</h1>
</header>

<h1>Premier titre du document</h1>
<p>Premier paragraphe</p>
```

- ★ `<nav>...</nav>` : représente une section d'une page ayant des liens vers d'autres pages ou des fragments de cette page. Autrement dit, c'est une section destinée à la navigation dans un document (avec des menus, des tables des matières, des index, etc...) (type « bloc »).

Exemple 3.34 Un exemple d'utilisation de `nav` :

```
<nav>
  <ol>
    <li><a href="#">Lien 1</a></li>
    <li><a href="#">Lien 2</a></li>
    <li><a href="#">Lien 3</a></li>
  </ol>
</nav>
```

- ★ `<section>...</section>` : représente une section générique d'un document, par exemple un groupe de contenu thématique. Une section commence généralement avec un titre (type « bloc »).

Exemple 3.35 Un exemple d'utilisation de `section` :

```
<section>
  <h1>Introduction</h1>
  <p>Ceci est un paragraphe.</p>
</section>
<section>
  <h1>Première partie</h1>
  <p>Un premier paragraphe</p>
  <p>Un second paragraphe</p>
</section>
```

- ★ `<article>...</article>` : représente du contenu autonome dans un document, une page, une application ou un site (type « bloc »). Ce contenu est prévu pour être distribué ou réutilisé indépendamment.

Exemple 3.36 Un exemple d'utilisation de `article` :

```
<article>
  <h1>Titre</h1>
  <p>Paragraphe</p>
</article>
```

- ★ `<aside>...</aside>` : représente une partie d'un document dont le contenu n'a qu'un rapport indirect avec le contenu principal du document (type « bloc »).

Exemple 3.37 Un exemple d'utilisation de `aside` :

```
<p>Un paragraphe de description</p>
<aside>Complément sur ce paragraphe</aside>
<p>Un autre paragraphe</p>
```

- ★ `<footer>...</footer>` : représente le pied de page de la section ou de la racine de sectionnement la plus proche. Un pied de page ou de section contient habituellement des informations sur l’auteur de la section, les données relatives au droit d’auteur (copyright) ou les liens vers d’autres documents en relation (type « bloc »).

Exemple 3.38 Un exemple d’utilisation de `footer` :

```
<article>
  <h1>Titre</h1>
  <p>Paragraphe</p>
  <footer>
    <p>copyright</p>
  </footer>
</article>
```

Remarque 3.39 Les aspects de toutes les balises sectionnantes peuvent être modifiés à partir de fichiers CSS.

3.7 Balises génériques

Les balises génériques sont des balises qui n’ont pas de sens sémantique comme les autres balises HTML. Ces balises sont utilisées en général pour construire le design de la page et lorsqu’aucune autre balise ne convient. Il en existe deux :

- ★ `<div>...</div>` : balise générique de type « bloc » ; utilisée pour mettre en valeur tout un contenu particulier de la page.

Exemple 3.40 Un exemple d’utilisation de `div` :

```
<div>
  <h1>Titre</h1>
  <p>Un premier paragraphe</p>
  <hr/>
  <p>Un second paragraphe</p>
</div>
```

- ★ `...` : balise générique de type « en ligne » ; utilisée pour mettre en valeur un passage particulier d’un texte présent dans la page.

Exemple 3.41 Un exemple d’utilisation de `span` :

Dans le langage HTML, les ``balises`` délimitent un contenu donc sont constituées d'une ``partie ouvrante`` et d'une ``partie fermante``.

Remarque 3.42 Les aspects de ces deux balises génériques peuvent être modifiés à partir de fichiers CSS.

4 Validation d’une page HTML

Afin de vérifier la syntaxe d’un document HTML et, en cas d’erreur, d’en être informé, le W3C fournit un validateur de fichier HTML à l’adresse suivante : <http://validator.w3.org/>