

2

Séparer le fond et la forme avec HTML et CSS

HTML signifie HyperText Markup Language. Cette abréviation de quatre lettres semble être une référence inévitable sur le C.V. de tout webmestre qui se respecte. Et pour cause : il s'agit ni plus ni moins du principal langage du Web. Il n'en va pas différemment aujourd'hui des feuilles de styles CSS.

Nombre de ces webmasters, même professionnels, ne connaissent pourtant qu'une infime part de ce langage, leur pratique se limitant souvent à utiliser un logiciel d'édition web qui génère des documents HTML automatiquement. Cette limitation, vous le percevrez tout au long de cet ouvrage, est un réel frein au développement et à la conception de sites web dans un avenir où le respect de normes rigoureuses sera de plus en plus la règle.

Nous n'allons pas ici retracer tout l'historique du HTML. Sachez simplement qu'il n'est qu'une application d'un standard bien plus ancien, le SGML. Ce dernier est un méta-langage, c'est-à-dire un ensemble de règles de grammaire à respecter par tout langage qui veut relever de sa famille. Conçu pour structurer des données, SGML utilise en outre une syntaxe à balises (*markup* en anglais). Ces deux caractéristiques le distinguent de langages de programmation plus classiques tels que le C, PHP, etc.

La base du HTML : les balises

HTML repose sur l'utilisation de balises et d'éléments.

JARGON **Ne pas confondre éléments et balises**

La confusion est fréquente entre les termes « balise » et « élément ». Nous précisons donc la signification de ces deux termes.

Les balises, délimitées par des chevrons ouvrant et fermant `<` et `>`, qualifient des portions de texte. Sauf exception, elles fonctionnent par paires, et à chaque balise ouvrante correspond sa balise fermante, débutant par les caractères `</`. Ces paires délimitent la portion de texte à laquelle elles s'appliquent. Parfois, une balise est refermée immédiatement. Une telle « balise vide » ne délimite alors qu'un point, et non une zone du document.

Un élément est l'ensemble composé d'une balise ouvrante, d'un contenu (du texte et/ou d'autres balises), et de la balise fermante correspondante. Voici un exemple faisant intervenir la balise ``, qui permet de renforcer une partie de texte :

Il ne faut `jamais` oublier de fermer les balises!

Dans cet exemple, nous distinguons :

- l'élément `jamais` ;
- le contenu d'élément « jamais » ;
- la balise ouvrante `` ;
- la balise fermante correspondante ``.

Le contenu « jamais », délimité par le couple de balises `` et ``, sera renforcé par rapport au reste du texte – les navigateurs graphiques recourent pour cela à une police plus grasse.

HTML propose toute une panoplie de balises pour structurer un document : paragraphes, titres, blocs, listes, tableaux, citations, etc.

Dans la version stricte et moderne du langage, le nom des balises s'écrit tout en minuscules. `` et `` n'y sont donc plus valables.

ÉVOLUTION **Balises obsolètes**

De nombreuses balises HTML sont aujourd'hui déconseillées voire obsolètes. Il s'agit de balises de mise en forme, fonction désormais dévolue aux styles CSS. Ce cimetière des balises oubliées comporte notamment ``, `<basefont>`, `<center>`, `<u>`, `<s>`, etc.

Les attributs

Les balises acceptent parfois des caractéristiques supplémentaires permettant de les distinguer ou de les personnaliser. Ces options s'inscrivent dans leur nom et sont appelées « attributs » (ou « propriétés »).

Leur valeur est toujours entourée d'apostrophes simples (') ou doubles (").

Exemple :

```
<h1 id="monTitre">Bienvenue chez moi</h1>
```

Ce code introduit un titre de premier niveau affublé d'une propriété `id` de valeur `monTitre`. La valeur de cet attribut, identifiant la balise, doit être unique dans le document (nous y reviendrons).

Les attributs sont indispensables pour distinguer les éléments entre eux et pouvoir créer la mise en page du document à l'aide des feuilles de styles CSS.

ÉVOLUTION Attributs obsolètes

Comme pour les balises, de nombreux attributs utilisés dans les langages HTML non stricts ou anciens sont pour la plupart destinés à la mise en forme des éléments : `align`, `valign`, `color`, `bgcolor`, `border`, `face`, `width`, `height`. Il est donc dorénavant déconseillé de les utiliser. Toutefois, pour des raisons de permissivité, ils sont encore tolérés dans les doctypes transitionnel et frameset.

Nous verrons dans les chapitres consacrés à la mise en forme en CSS comment obtenir, à l'aide de styles, l'effet auparavant produit par ces attributs.

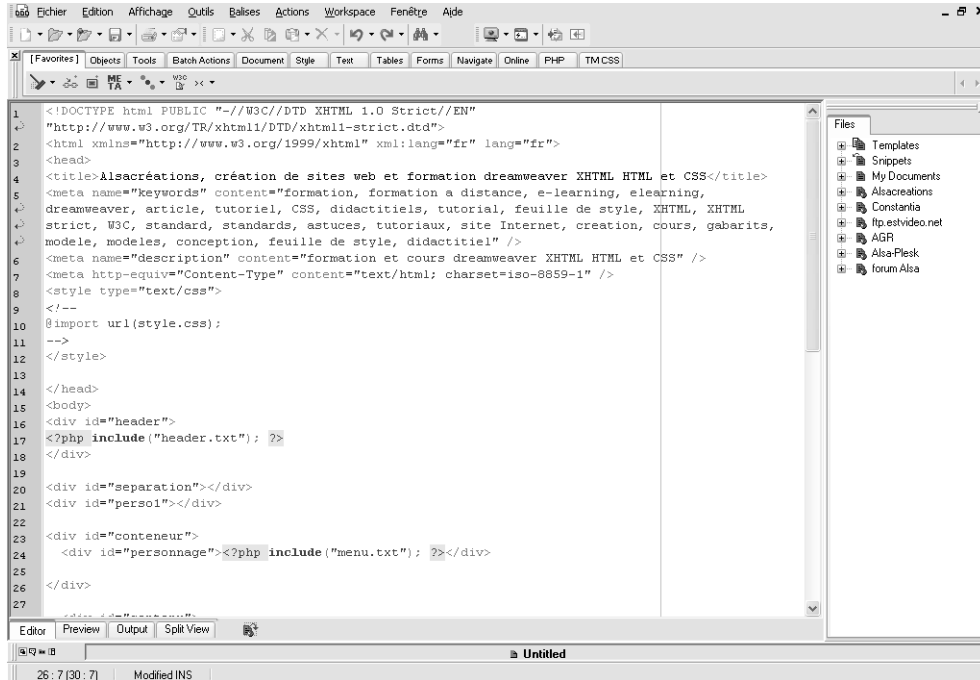
L'imbrication des éléments

Les éléments peuvent être imbriqués, à condition de former un « bon parenthésage » respectant la hiérarchie, et en évitant de faire se chevaucher des balises ouvrantes et fermantes qui ne se correspondent pas. Par exemple :

```
<h1 id="monTitre">Bienvenue chez Toto <em>et moi</em></h1>
```

Nous sommes en présence d'un titre de premier niveau plaçant en emphase la portion de texte « et moi ». Cette particularité est souvent traduite par des italiques dans les navigateurs graphiques (ou par une police droite dans un texte déjà en italique). Remarquez l'ordre des éléments : la balise `` s'ouvre et se ferme à l'intérieur de l'élément défini par la balise `<h1>`.

Nous verrons plus loin que (presque) tous les documents web sont structurés sous forme de balises à plusieurs niveaux d'imbrication. Cette structure constitue une arborescence globale qui débute par la première balise, racine du document : `<html>`.



```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
4 <head>
5 <title>Alsacrations, création de sites web et formation dreamweaver XHTML HTML et CSS</title>
6 <meta name="keywords" content="formation, formation a distance, e-learning, elearning,
7 dreamweaver, article, tutoriel, CSS, didactiels, tutorial, feuille de style, XHTML, XHTML
8 strict, W3C, standard, standards, astuces, tutoriaux, site Internet, creation, cours, gabarits,
9 modele, modeles, conception, feuille de style, didacticiel" />
10 <meta name="description" content="formation et cours dreamweaver XHTML HTML et CSS" />
11 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
12 <style type="text/css">
13 <!--
14 <import url(style.css);
15 -->
16 </style>
17 </head>
18 <body>
19 <div id="header">
20 <?php include("header.txt"); ?>
21 </div>
22 <div id="separation"></div>
23 <div id="perso1"></div>
24 <div id="conteneur">
25 <div id="personnage"><?php include("menu.txt"); ?></div>
26 </div>
27
```

Figure 2-1
Exemple de code HTML produit avec le logiciel HTML-Kit

La structure des éléments

La bonne compréhension de la structure des éléments est indispensable. Elle aura de nombreuses applications par la suite, mais c'est paradoxalement un sujet rarement connu des webmasters.

Les éléments HTML ont chacun une structure particulière. Il en existe deux familles : les éléments de type bloc et les éléments au fil du texte, dits en ligne (ou *inline*). Ce type dictera tous leurs comportements : positionnement, affichage, etc.

La distinction fondamentale est assez claire. Les blocs distinguent des parties entières comme des titres, des paragraphes, des listes, des citations, etc. Les éléments en ligne sont prévus pour enrichir localement des portions de texte (lien hypertexte, emphase, renforcement, etc.).

Il en découle des spécificités d'affichage que nous détaillerons dans le chapitre 7, consacré au positionnement :

- Même si dans le code HTML ils sont écrits côte à côte, les éléments de type bloc sont par défaut placés l'un sous l'autre par le navigateur. Exemples : une suite de paragraphes (balise `<p>`) ou les éléments d'une liste (balise ``).
- Les éléments en ligne s'appliquent à des portions de texte, comme des groupes de mots dans une phrase. Par exemple : le renforcement d'une partie de texte à l'aide de la balise ``, comme vu précédemment.

Quelques exemples pratiques

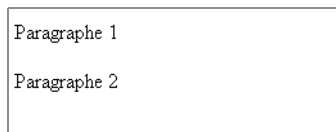
Les échantillons de code suivants permettront de mieux cerner cette différence fondamentale de structure.

```
<p>Paragraphe 1</p><p>Paragraphe 2</p>
```

Ces deux paragraphes occuperont chacun une ligne distincte, car la balise `<p>` est de type bloc.

```
<strong>Toto</strong> et <em>moi</em>
```

Figure 2-2
Affichage de deux
éléments de type bloc

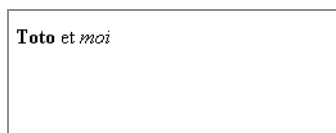
Une boîte rectangulaire à double bordure contenant deux lignes de texte. La première ligne est "Paragraphe 1" et la deuxième ligne est "Paragraphe 2".

Paragraphe 1
Paragraphe 2

Ce texte s'affichera à la suite (sans retour à la ligne) car les balises qui le définissent sont en ligne.

```
<p><strong>Toto</strong> et <em>moi</em></p>
```

Figure 2-3
Affichage de deux
éléments en ligne


Une boîte rectangulaire à double bordure contenant une seule ligne de texte. Le premier mot "Toto" est en gras et le mot "moi" est en italique.

Toto et *moi*

Ce code ne prendra (probablement) qu'une ligne à l'écran car il ne comporte qu'une seule balise de type bloc et son contenu reste au fil du texte.

```
<h1><strong>Toto</strong> et <em>moi</em></h1>  
<p>Je connais <strong>Toto</strong> depuis mon enfance.</p>
```

Figure 2-4
Affichage mixte d'un bloc
avec éléments en ligne



Toto et moi

Cette portion du document apparaîtra en deux parties : la première comprendra le titre représenté par l'élément bloc `<h1>` (contenant à son tour deux éléments en ligne). Le bloc de paragraphe `<p>` constituera la seconde. Il contient lui aussi un élément en ligne.

Figure 2-5
Affichage mixte d'éléments
bloc et en ligne



Toto et moi
Je connais **Toto** depuis mon enfance

Contenus autorisés selon les types de balises

Ces deux familles de balises diffèrent également par les contenus qu'elles acceptent, en dehors du texte à proprement parler :

- Un élément bloc peut contenir un (ou plusieurs) éléments bloc et/ou en ligne, à l'exception des éléments de paragraphe `<p>` et des titres `<h1>`, `<h2>`..., limités aux contenus en ligne.
- Un élément en ligne ne peut renfermer que d'autres éléments en ligne.

Il existe aussi deux sortes d'éléments en ligne : « remplacés » et « non remplacés ».

- Seuls les éléments remplacés acceptent des attributs de dimensions (`height`, `width`). Il s'agit des éléments ``, `<input>`, `<textarea>`, `<select>` et `<object>`.
- Les autres n'ont pas de dimension à proprement parler, et n'occupent que la place nécessaire à leur contenu. C'est le cas des éléments ``, ``, `<a>`, ``, etc.

Une mise en page se fera donc préférentiellement à l'aide de balises de type bloc. La plus indiquée pour cet usage est `<div>` (« division ») : c'est une balise générique servant de conteneur neutre.

Voici une illustration de notre propos :

```
<div><p>Paragraphe 1</p><p>Paragraphe 2</p></div>
```

La balise de bloc `<div>` englobe les deux blocs de paragraphe, ce qui est autorisé. En revanche, ce qui suit est invalide :

```
<span><p>Paragraphe 1</p><p>Paragraphe 2</p></span>
```

En effet, la balise en ligne `` ne peut contenir des blocs tel le paragraphe (`<p>`).

Les éléments en ligne

Le contenu des éléments en ligne est affiché au fil du texte. Par exemple : des portions de phrases mises en exergue dans un paragraphe, des liens hypertexte, etc. On les appelle encore « éléments internes » car ils visent à donner du sens à des portions du document tout en restant au fil du texte, c'est-à-dire sans créer de nouveaux blocs.

C'est par exemple le cas des éléments de renforcement (représentés en gras par les navigateurs graphiques), et d'emphase (italique).

Ces éléments, conçus pour demeurer au sein du texte afin de l'enrichir et de lui apporter du sens, ne sont pas vraiment prévus pour un positionnement précis dans le document (même si cela est possible). À l'exception des éléments remplacés, déjà mentionnés, leurs dimensions seront déterminées par leur contexte.

Par défaut, et contrairement aux éléments de type bloc, les éléments en ligne ont des marges internes et externes d'épaisseur nulle, donc inexistantes.

Les éléments de type bloc

Il s'agit de ceux dont le rendu visuel forme un bloc. C'est par exemple le cas des paragraphes.

Cette structure leur permet de prévoir des dimensions (hauteur, largeur, profondeur), de contenir d'autres éléments dimensionnés, et de posséder des marges internes (`padding`). Leur propriété la plus importante est la possibilité de les placer (ou positionner) en les sortant du flux du document, contribuant ainsi à sa mise en page.

Ces caractéristiques leur sont pour la plupart réservées. Ainsi, une balise en ligne comme `` ne pourra posséder de dimensions propres, celles-ci dépendant du contenu (texte ou image) de son élément.

Il est facile de passer d'une structure bloc à une structure en ligne (et inversement) grâce à la propriété CSS `display`. Nous verrons également comment positionner les différents éléments selon leur type.

PERTURBANT Marges par défaut

À l'exception de la balise neutre `<div>`, tous les éléments de type bloc possèdent par défaut des marges internes (`padding`) et externes (`margin`) non nulles. Vous constaterez toute leur importance en observant que les différents navigateurs leur donnent des valeurs différentes. C'est pourquoi il faut parfois les annuler ou les expliciter pour éviter de grosses différences de rendu visuel.

Les principales balises HTML standards

Depuis la naissance du HTML, le W3C a créé de nombreuses balises.

Au fur et à mesure des versions successives de HTML (et des conflits d'intérêt entre les navigateurs), beaucoup d'éléments ont été déclarés « dépréciés » (traduction fréquente mais incorrecte du faux-ami *deprecated*, qui indique plutôt une désapprobation), ou obsolètes.

HTML 4 (XHTML 1) compte actuellement une trentaine de balises de type bloc et autant de balises en ligne.

Nous n'aborderons ici que les balises principales et les plus utiles. Compte tenu des objectifs de cet ouvrage, nous passerons sous silence les balises obsolètes, créées originellement pour modifier la mise en forme visuelle du document.

Les principales balises en ligne

Parmi toutes les balises au fil du texte, six sont à connaître. Elles sont regroupées dans le tableau 2-1. Les balises incontournables du type bloc sont présentées au tableau 2-2.

Tableau 2-1 Les principales balises en ligne

Balise	Commentaires	Exemple d'utilisation
<a>	Désigne un lien hypertexte. Elle s'accompagne de l'attribut href, qui renferme la cible du lien (son contenu représentant le texte à cliquer pour activer le lien).	allez en page 2
	Met en emphase une portion de texte. Quand la police utilisée est droite, la plupart des navigateurs graphiques la traduisent comme une mise en italique.	<p>Voici un texte important.</p>
	Inclut une image dans le document. Cette balise s'accompagne des attributs alt (texte alternatif pour malvoyants ou navigateurs en mode texte) et src (qui indique le chemin vers l'image). Le format de fichier de l'image, en théorie libre, est souvent limité à ce que comprennent les navigateurs – c'est-à-dire GIF, JPEG, et PNG. Remarquons que est une balise sans contenu (« auto-fermante », ou encore « balise vide »). Ceci se traduit par son /> final.	

Tableau 2-1 Les principales balises en ligne (suite)

Balise	Commentaires	Exemple d'utilisation
<q>	Balise utilisée pour les citations courtes, en ligne. On utilisera la balise bloc <blockquote> pour des citations plus longues.	<p>Comme le dit toujours ma grand-mère : <q>il ne faut pas vendre la peau de l'ours avant de l'avoir tué </q>.</p>
	Conteneur en ligne générique, dépourvu d'un sens précis mais qui peut servir à regrouper d'autres éléments au fil du texte. Son équivalent est l'élément <div>.	un texte
	Indique un renforcement, généralement représenté en gras dans les navigateurs graphiques.	<p>Comme le dit toujours ma grand-mère : <q>il ne faut surtout pas vendre la peau de l'ours avant de l'avoir tué</q>.</p>

Tableau 2-2 Les principales balises de type bloc

Balise	Commentaires	Exemple d'utilisation
<blockquote>	Introduit des citations longues. Par défaut, certains navigateurs prévoient une marge gauche aux blocs de citation, qu'on pourra bien sûr modifier en CSS.	<p>Comme le dit toujours ma grand-mère :</p><blockquote><p>il ne faut pas vendre la peau de l'ours avant de l'avoir tué</p><p>non non, il ne faut pas !</p></blockquote>
<div>	Conteneur générique de type bloc. Cette balise n'apporte pas de sens spécifique, à l'instar de son équivalent en ligne . Elle sert à regrouper d'autres balises bloc ou en ligne.	<div><p>Voici un texte important.</p><p>Voici un autre texte</p>.</div>
<dl>	Liste de définitions, utile pour structurer les éléments associant des définitions ou contenus à des termes ou à des titres. Ces listes distinguent les titres (<dt>) des éléments de définition (<dd>).	<dl><dt>Titre de l'élément</dt><dd>description de l'élément</dd><dd>Suite de la description</dd></dl>

Tableau 2-2 Les principales balises de type bloc (suite)

Balise	Commentaires	Exemple d'utilisation
<form>	Balise délimitant un formulaire interactif. Elle contient généralement des éléments d'interface (champs de texte, boutons de validation, cases à cocher, etc.) Cet élément doit renfermer immédiatement d'autres éléments de type bloc.	<pre><form action="pagesuivante.php" method="get"> <p> <input type="text" name="recherche" /> <input type="submit" value="ok" /> </p> </form></pre>
<h1>, <h2>, ... <h6>	HTML prévoit six niveaux de titres, hiérarchiquement placés sous le titre principal (<h1>). Rappel : ces éléments constituent une exception à la règle des blocs ; ils ne peuvent pas contenir d'autres blocs.	<pre><h1>Un titre principal</h1></pre>
, 	Ces deux balises désignent des listes ordonnées () ou à puces simples (). Elles comportent exclusivement les éléments d'objet de liste ().	<pre> premier objet de la liste second objet de la liste </pre>
<p>	Balise désignant un paragraphe de texte. Cet élément constitue une exception à la règle des blocs car il ne peut en contenir d'autres.	<pre><p>Un paragraphe de texte</p></pre>
<table>	Tableau contenant des données. Les cellules du tableau sont d'abord rassemblées sous forme de rangées ou lignes (<tr>).	<pre><table> <tr> <td>cellule</td> <td>autre cellule</td> </tr> </table></pre>

Ces tableaux ne reprennent qu'une partie des éléments proposés par le HTML. Bien d'autres vous permettront de structurer vos pages (<dfn>, <address>, <acronym>, <abbr>, <kbd>, <samp>, etc.). Le lecteur soucieux de créer des structures de documents claires et logiques se penchera de plus près sur cette longue liste.

La plupart des logiciels générant du code automatique boudent les balises les plus appropriées au profit d'un petit échantillon de balises génériques dépourvues de sens réel. Cette accumulation de <div> et de conduit à une pauvreté sémantique préjudiciable au document.

Séparer le contenu de la mise en forme

La bonne connaissance de la signification et surtout de la sémantique de chacun des éléments aide à concevoir des documents structurés, faciles à interpréter par tous les agents ou clients existants, même non graphiques.

La représentation et mise en forme de ces éléments ne passe qu'après leur organisation logique. Un document web est avant tout une source d'informations qui doit rester accessible à tous. Il est donc nécessaire de s'appliquer à produire une structure claire et porteuse de sens avant de s'attacher à l'aspect purement graphique du document.

La structuration logique des données

XHTML étant devenu un système de balisage sémantique du contenu, les structurations qu'il opère désormais sont purement logiques.

Comme nous l'avons vu, les balises seront choisies en vertu de leur sens, non en fonction de leur représentation par défaut sur tel ou tel navigateur ni l'aspect qu'on veut leur donner.

XHTML ne se préoccupe ainsi plus que de la structure logique du document, la mise en forme (caractères, couleurs, marges, etc.) relevant des feuilles de styles. Ce manuel vise principalement à vous convaincre des avantages de cette nouvelle approche et à mettre celle-ci en œuvre correctement. C'est la nouvelle ligne de conduite du HTML : séparer le contenu de la mise en forme.

En XHTML, il est recommandé d'abandonner les balises de mise en forme graphique comme `<i>` (désignant l'italique) au profit de balises de structuration plus logique comme `` (mise en emphase). De même, l'élément de renforcement `` remplacera désormais la balise de gras ``.

Adoptez vous aussi ces nouvelles habitudes, et laissez de côté les balises conçues pour la mise en forme visuelle du document.

Le principe des feuilles de styles

Dans cet ouvrage, notre démarche pourra se résumer à l'abandon de toute balise de mise en forme au profit des seules balises porteuses de sens (et des conteneurs génériques `` et `<div>`). Il est temps de jeter aux orties les ``, `<center>` et autres balises de mise en page !

La mise en forme des documents se fera dès lors par des feuilles de styles détaillant la représentation des balises et attributs retenus.

Par exemple :

```
<p style="font-family:verdana,arial; font-weight:bold; color:red;">
```

qu'on peut encore améliorer en écrivant `<p class="erreur">` après avoir défini les propriétés de la classe erreur.

ATTENTION Ne pas confondre <div> et « calques »

Les balises neutres `<div>` servant de conteneurs de blocs, elles désignent une boîte rectangulaire invisible. Généralement, cet élément est assimilé au concept de « calque », surtout sur les éditeurs HTML comme Dreamweaver ou Golive. Un « calque » est alors un `<div>` positionné avec l'une des propriétés `position: absolute`, `position: relative`, ou `position: fixed`.

Rien n'oblige pourtant à imposer cette propriété de placement : on peut souvent s'en passer en plaçant les `div` les uns par rapport aux autres grâce à la propriété `margin`. Le chapitre 7 sera entièrement consacré au positionnement des éléments via CSS.

Mise en situation

Débutons un petit projet qui nous servira de fil conducteur tout au long des chapitres suivants.

Il s'agit de concevoir un site web de A à Z, de sa structure à sa mise en page complète, en passant par le placement des différents éléments et par quelques réflexions sur ses menus et la manière d'y naviguer.

Endossez l'identité d'un Alsacien amoureux (à juste titre) de sa région. Votre projet consiste à concevoir un site web de tourisme en Alsace comprenant des textes de présentation, des liens et illustrations, ainsi que des photographies de cette belle région.

Exercice numéro 1

Votre page d'accueil comprend un titre principal (« Bienvenue en Alsace »), puis deux titres secondaires (« Une belle région française » et « Un patrimoine considérable »), associés chacun à un paragraphe de texte explicatif. Pour finir, un pied de page contiendra une présentation de votre société et un lien vers diverses précisions juridiques.

Comment structurer cette page d'accueil en HTML ?

Les balises de titres sont les balises de bloc `<h1>` à `<h6>`. Le titre principal sera donc défini à l'aide de la balise `<h1>` et les titres secondaires avec `<h2>`. Quant aux paragraphes, ils seront délimités par la balise de bloc `<p>`.

Pour le pied de page, on préférera encore une balise de paragraphe à la balise générique `<div>`.

Voici un exemple de code final pour cette page d'accueil :

```
<h1>Bienvenue en Alsace</h1>
<h2>Une belle région française</h2>
<p>[Paragraphe associé au sous-titre de niveau 2.]</p>
<h2>Un patrimoine considérable</h2>
<p>[Paragraphe associé à cet autre sous-titre de niveau 2.]</p>
<p>Pied de page et <a href="#">Mentions légales</a></p>
```

Rappelons le cas particulier des balises de titres et de paragraphes : contrairement aux autres éléments de type bloc, elles ne peuvent pas renfermer des blocs. Il n'est donc pas possible d'imbriquer un paragraphe dans une balise de titre.

Exercice numéro 2

Le titre principal « Bienvenue en Alsace » doit être un lien hypertexte menant à une page contenant diverses photographies : `photos.htm`.

Vous associerez à ce lien une infobulle déclenchée par le passage du pointeur de la souris et expliquant que ce lien mène à vos photos de présentation.

Quel choix allez-vous faire parmi ces trois propositions ?

- 1 `
<h1>Bienvenue en Alsace</h1>`
- 2 `<h1>
Bienvenue en Alsace</h1>`
- 3 `<h1>
Bienvenue en Alsace</h1>`

Prenez garde à la structure des balises. La balise de lien `<a>`, étant un élément de type en ligne, ne peut pas contenir d'éléments de type bloc comme `<h1>`. Ceci exclut donc la première proposition.

Il s'agit donc de choisir entre les attributs `alt` et `title`.

Le premier signifie « texte alternatif ». On l'associe aux images pour préciser un texte qui sera par exemple affiché sur les navigateurs non graphiques, comme les plages en braille utilisées par les aveugles. Pour information, l'attribut `alt` est propre à l'élément image `img`.

L'attribut `title` correspond quant à lui à l'infobulle, c'est-à-dire au comportement recherché ici. C'est donc la dernière solution qui est la bonne.

La confusion entre `alt` et `title` est fréquente car Internet Explorer déclenche aussi une infobulle sur l'attribut `alt`, comportement non prévu par la norme.

Exercice numéro 3

L'un des deux paragraphes de texte contiendra une image cliquable, servant de lien vers la page de photographies régionales.

Comment procéder ?

L'image `` devant se comporter comme un lien, il suffit de l'inclure dans une balise de lien (`<a>`). Le contenu de cette dernière (c'est-à-dire l'image) deviendra alors un lien hypertexte.

L'exercice précédent rappelait que la balise en ligne `<a>` ne pouvait pas contenir d'éléments de type bloc tels que des titres. Il est en revanche parfaitement autorisé d'y inclure une image, puisque la balise `` est de type inline.

On écrira donc un code proche de :

```
<p>[Paragraphe associé au sous-titre de niveau 2.]  
<a href="photos.htm" title="lien vers des photos d'Alsace">  
</a>  
</p>
```

Exercice numéro 4

Vous avez décidé d'inclure un menu vertical de navigation comprenant les liens suivants :

- Retour à l'accueil ;
- Présentation de la région ;
- Historique de l'Alsace ;
- Gastronomie locale ;
- Hôtels et gîtes ;
- Photographies.

Quelle structure parmi les trois suivantes vous paraît-elle la plus appropriée ? Pourquoi ?

- 1 Inclure chaque élément (<a>) du menu au sein d'un élément de bloc comme les balises <div> ou <p>, de sorte que ces liens s'affichent les uns sous les autres.
- 2 Séparer ces liens par des balises de retour à la ligne
.
- 3 Utiliser les éléments de liste (et) pour structurer ce menu et ses liens.

Exemple:

```
<ul>  
<li><a href="...">Retour à l'accueil</a></li>  
<li><a href="...">Présentation de la région</a></li>  
...  
</ul>
```

Le succès d'un site web dépendant directement de son ergonomie et de son intuitivité, l'élaboration des méthodes de navigation permettant d'y circuler fera l'objet d'une réflexion poussée et d'une grande attention.

Ces trois propositions auront des aspects sensiblement équivalents : chacune affichera une liste de liens placés les uns sous les autres.

Rappelons que l'aspect visuel présent ne doit pas compter (il sera toujours temps de le corriger dans la feuille de styles, et les chapitres suivants vous feront la

démonstration de la grande puissance de cette technique). Il s'agit uniquement de construire le menu le plus logiquement possible. Quelle est donc la structure la plus sémantique parmi les trois proposées ?

Séparer les liens par une balise de retour à la ligne `
` est une erreur : il s'agit justement du piège à éviter, puisque l'on ne se préoccupe alors que du rendu visuel de la navigation. Placer chacun des liens dans une balise de type bloc est intéressant : cela permet de les séparer d'une manière structurelle et non graphique. Cependant, le choix des balises `<div>` ou `<p>` est discutable. En effet, la première est une balise générique n'apportant aucun sens au menu de navigation. Quant à la seconde, qui désigne un paragraphe de texte, elle serait utilisée ici à contre-emploi.

Le plus approprié est ici d'opter pour les balises de listes non ordonnées. Il s'agit bel et bien d'une liste (éléments de navigation) ; ce choix convient donc tout à fait et il est porteur de sens. Nous verrons plus loin, dans le chapitre consacré à la conception de menus, comment supprimer les puces par défaut et donner aux listes l'habillage graphique souhaité.

Cette première mise en situation débouche ainsi sur un code HTML proche de :

```
<body>
<ul>
<li><a href="#">Retour à l'accueil</a></li>
<li><a href="#">Présentation de la région</a></li>
<li><a href="#">Historique de l'Alsace</a></li>
<li><a href="#">Gastronomie locale</a></li>
<li><a href="#">Hôtels et gîtes</a></li>
<li><a href="#">Photographies</a></li>
</ul>
<h1><a href="photos.htm" title="photos d'Alsace">Bienvenue en Alsace
</a></h1>
<h2>Une belle région française</h2>
<p>[Paragraphe associé au sous-titre de niveau 2.]
<a href="photos.htm" title="lien vers des photos d'Alsace">
</a></p>
<h2>Un patrimoine considérable</h2>
<p>[Paragraphe associé à cet autre sous-titre de niveau 2.]</p>
<p>Pied de page et <a href="#">Mentions légales</a></p>
</body>
```

Le menu est volontairement placé au début du code HTML. Ceci s'explique par des raisons structurelles : la navigation représente la boussole d'une page web, indiquant la position actuelle et proposant d'autres destinations.

C'est en général le premier point de repère dans la page. Placer ainsi la navigation en tête la situera en bonne position dans les navigateurs graphiques ainsi que dans les navigateurs en mode texte ou en braille. Cela facilitera la manipulation du site par les utilisateurs de ces programmes.