

Chapitre 2

Réalisation d'une page simple

2.1 Un dynamisme croissant

En 1990, Tim Berners-Lee, un informaticien du CERN invente le concept *de world wide web*¹. Il s'agit de créer des pages comportant un contenu et des liens entre les réseaux des différentes universités travaillant sur des projets scientifiques. Le principe est en soi dynamique puisqu'une page à laquelle je peux accéder par un lien peut varier d'un jour à l'autre et m'offrir des liens vers d'autres pages, nouvelles ou non, que je connaissais pas encore.

Ce dynamisme reste relativement modéré puisque la page elle-même est éminemment statique. Toute modification impose l'édition du code HTML qui la sous-tend, une opération fastidieuse. Quoi qu'on dise, il reste un grand nombre de site bâti sur ce principe, notamment des sites publicitaires pour des commerçants qui n'envisagent pas de réaliser des affaires en ligne². Evidemment, la plupart des gens voudront un site attractif et on mettra un maximum de flash et de CSS pour que le site soit joli.

Les sites originels se sont vus peu à peu enrichis de différentes techniques :

- des illustrations pour agrémenter l'aspect et la convivialité du site, ainsi que des images à zones réactives pouvant servir de liens
- les feuilles de style pour une bonne séparation du contenu et de l'aspect
- DHTML (c'est-à-dire un mélange de javascript et d'HTML) pour animer les menus et améliorer la gestion des formulaires
- des programmes en Flash ou en Java pour provoquer une animation. Notons qu'ici, à partir du moment où le programme échange des données avec le serveur, nous quittons le domaine du statique.

Devant la lourdeur de gestion des listes d'inventaire et d'articles, ou dans le cas de site devant sans cesse être modifié, on a dû innover. Des programmes se sont vus assigner la tâche de réaliser les pages de manière dynamique, en fonction des demandes de l'utilisateur et en fonction d'informations venues d'ailleurs, mises à jour en temps réel. On a rapidement privilégié l'utilisation des langages de scripts, Perl, puis PHP. D'autres techniques, issues du monde Microsoft ou de la galaxie Java, sont également intervenues (ASP, les servlets...). Dernière en date, la technologie Ajax autorise la mise à jour de portions de page en interaction avec l'utilisateur et/ou le serveur. Ces techniques atteignent leur paroxysme avec la technique du *cloud* : le site devient le réceptacle des données et des programmes utilisés par un utilisateur nomade. En plein essor, cette technique est maintenant pleinement opérationnelle (voir par exemple, les nombreux services « offerts » par Google).

1. Voir <http://public.web.cern.ch/public/fr/About/Web-fr.html>

2. Un exemple proposé par une firme de création de site en Belgique : <http://www.pro-painting.be>. On remarquera que la modicité du contenu est ici compensée par des effets visuels et de la musique.

2.2 Structure de base

On peut considérer une page web comme une arborescence structurée et entourée des balises racines `<html>` et `</html>`. Dans la racine, on trouve deux éléments : `head` et `body` qui comportent respectivement la définition de la page et son contenu. Dans l'en-tête, on trouve quelques éléments, dont le seul à être directement lisible est `title`, et qui permet de spécifier le texte qui s'affichera comme titre de la fenêtre (généralement à côté du nom du navigateur). Les autres éléments sont facultatifs : mentionnons `meta` permettant de renseigner les navigateurs et les robots d'indexation sur le comportement à adopter à l'égard de la page, et `link` pour indiquer le nom des feuilles de style.

```

<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML_1.0_Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content=
    "text/html; charset=ISO-8859-15" />
  <meta name="author" content="Jacques_THOORENS" />
  <meta name="robots" content="index, follow" />
  <title>Java et mod&eacute;lisation objet</title>
  <meta name="description"
    content="Java_et_mod&eacute;lisation_objet" />
  <link rel="stylesheet" type="text/css"
    href="css/styledef.css"
    title="defaut" media="screen" />
</head>
<body>
... Contenu de la page
</body>

```

Les premières lignes appartiennent aux conventions XML et spécifient le numéro de version XML et le type de langage employé.

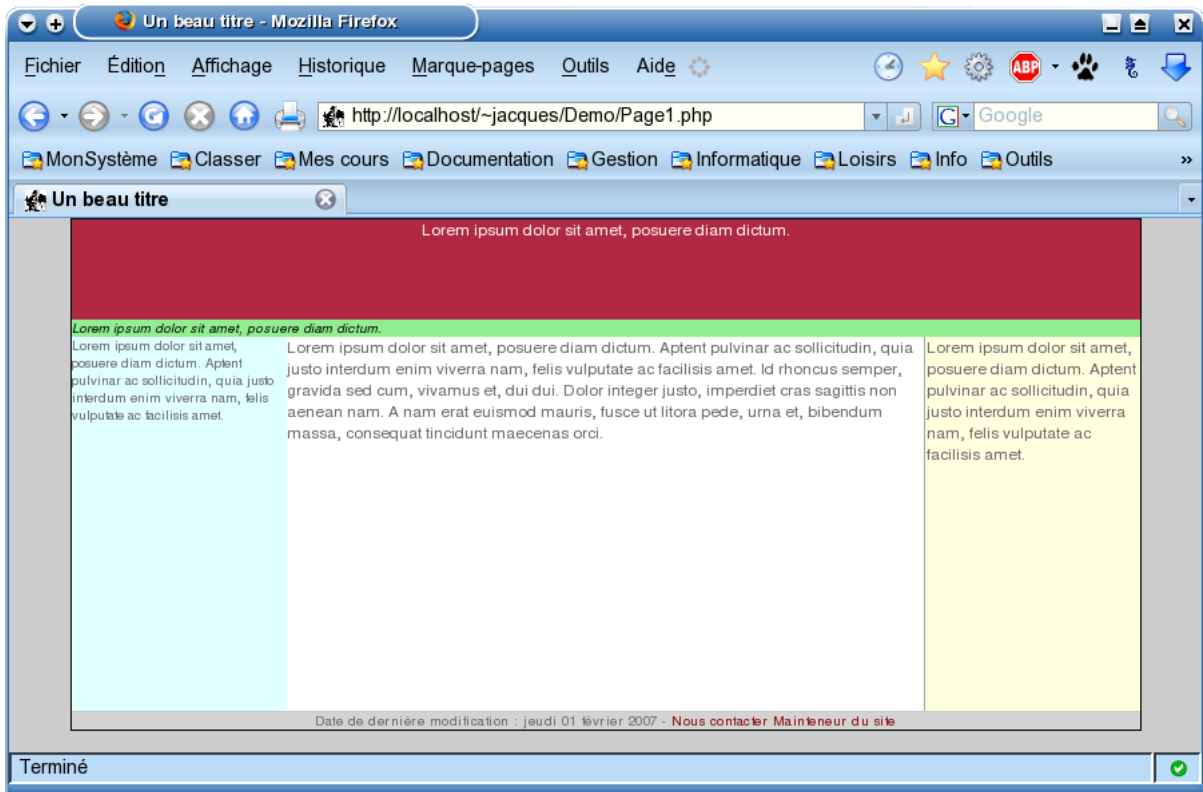
2.2.1 Une page bien structurée

Plus personne n'écrit des pages ne comportant qu'une seule zone. En général, on découpe logiquement la page en plusieurs parties :

- l'**en-tête**, souvent orné de graphiques et comportant le nom du site et un menu d'accès aux différentes parties
- trois zones avec du contenu : une partie **gauche**, une partie **centrale** et une partie **droite**, l'un ou l'autre pouvant manquer. En général la partie centrale contient le contenu proprement dit, tandis que les zones latérales servent à placer des liens annexes, des menus, des petites informations ou souvent des publicités.
- un **pied** de page contient généralement quelques informations techniques, des liens vers les responsables du site ou des mentions légales.
- un **historique** , souvent placé sous l'en-tête, permet à l'internaute de trouver le chemin qui l'a mené là, en retrouvant des liens sur les étapes intermédiaires.

On réalise ces différentes zones à l'aide d'éléments `div` auxquels on attribue des attributs `id` afin de pouvoir spécifier des styles qui leurs sont affectés. Ces styles précisent l'aspect, mais

surtout la position des différentes parties. Voici un exemple de page basée sur ce principe, dans laquelle, j'ai supprimé le contenu pour ne laisser apparente que la structure.



```

<body>
<div id="main">
<div id="tete">
    contenu en-tête
</div><!-- end tete-->
<div id="historique">
    historique
</div><!-- end historique-->
<div id="gauche">
    partie gauche
</div><!-- end gauche-->
<div id="droite">
    partie droite
</div><!-- end droite-->
<div id="centre">
    contenu principal
</div><!-- end centre-->
<div id="pied">
</div><!-- end pied-->
</div><!-- end main-->
</body>
</html>

```

Pour obtenir l'aspect de la page illustrée plus haut, il suffit de lier une feuille de style qui prend en compte ces différentes parties. Une simple ligne à placer dans l'élément head :

```
<link media="screen" title="base" type="text/css"
href="css/base.css" rel="stylesheet"/>
```

Voici le contenu du fichier base.css :

```
* {
margin:0;
padding:0;
}
#main {
border : 1px solid black;
margin-left: auto;
margin-right: auto;
width : 800px;
}
#tete {
text-align: center;
}
#gauche {
float:left;
min-height : 280px;
width:20%;
padding-right:1px;
}
#centre {
min-height : 280px;
padding-left : 1px;
text-align:left;
}
#droite {
float:right;
width:20%;
min-height : 280px;
padding-left : 1px;
}
#pied {
clear:both;
font-size:9px;
text-align: center;
float:none;
letter-spacing: 0.5px;
}
}
```

2.2.2 Conception assistée...

Complicé? Certes, mais ce style peut resservir pour toutes les pages conçues sur ce schéma. De plus de nombreux outils facilitent la réalisation de telles page. À cet égard, je ne peux que conseiller la consultation du site <http://builder.yaml.de/>. Il offre les services suivants :

- un framework complet pour la réalisation assistée de pages comprenant deux ou trois colonnes, en-tête, pied, navigation par onglets ou boutons et menus hiérarchiques.

- une compatibilité garantie pour tous les navigateurs actuels (y compris en mode texte).
- un manuel (en anglais) exposant la problématique, les solutions et leur mise en œuvre.
- un générateur en ligne (yaml-builder) permettant de réaliser un squelette de page en fonction de ses besoins propres.

Une remarque néanmoins : l'utilisation gratuite de YAML nécessite la pose d'un lien vers le site concerné.

2.3 Les éléments de mise en page

2.3.1 Titres

HTML dispose de six niveaux de titres de 1 à 6. Les balises permettant de réaliser ces titres vont également par paires :

<code><h1>...</h1></code>	<code><h2>...</h2></code>	<code><h3>...</h3></code>
<code><h4>...</h4></code>	<code><h5>...</h5></code>	<code><h6>...</h6></code>

2.3.2 Mises en évidence

On peut changer le type des caractères. Nous nous contenterons de quatre effets :

Balises	Effet	Equivalent CSS
<code> ...</code>	Texte mis en gras	<code>font-weight:bold;</code>
<code><i>...</i></code>	<i>Texte en italiques ou en évidence</i>	<code>font-style:italic;</code>
<code><u>...</u></code>	<u>Texte souligné</u>	<code>text-decoration: underline;</code>
<code><tt>...</tt></code>	Texte en caractères à échappement fixe	<code>font-family: monospace;</code>

On remarquera que ce sont en fait les seuls attributs employés dans ces notes. Ils suffisent généralement à assurer une présentation claire. Notons que ces balises servent à mettre en évidence un mot ou deux, mais qu'elles ne doivent pas s'employer pour caractériser un paragraphe.

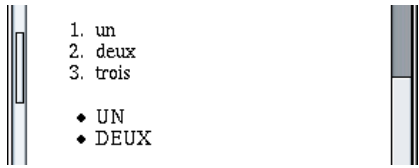
2.3.3 Listes

Beaucoup de variantes sont disponibles à ce niveau, je me contenterai de citer les deux formes les plus simples à travers une simple illustration :

```

<ol>
  <li>un</li>
  <li>deux</li>
  <li>trois</li>
</ol>
<ul>
  <li>UN</li>
  <li>DEUX</li>
</ul>

```



Les feuilles de style permettent de modifier l'aspect des listes (notamment de les placer horizontalement). Elles servent souvent de base aux menus.

2.3.4 Les tableaux

Les tableaux jouent un rôle capital dans l'affichage des données provenant de bases de données. Ils servaient autrefois à faire des pages complexes. Ce n'est plus très prisé aujourd'hui.

Un tableau est défini par une série de paires d'éléments, illustrés par l'exemple suivant.

Début	Fin	Signification
<table>	</table>	Encadrent tout le tableau (<i>table</i>)
<tr>	</tr>	Encadrent une ligne (<i>row</i>)
<td>	</td>	Encadrent une cellule de tableau
<th>	</th>	Encadrent une cellule de l'en-tête (<i>header</i>)

```
<table border="3">
  <thead>
    <tr>
      <th>Début</th>
      <th>Fin</th>
      <th>Signification</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td><tt>&lt;table></tt></td>
      <td><tt>&lt;/table></tt></td>
      <td>Encadrent tout le tableau (<em>table</em>)</td>
    </tr>
    <tr>
      <td><tt>&lt;tr></tt></td>
      <td><tt>&lt;/tr></tt></td>
      <td>Encadrent une ligne (<em>row</em>)</td>
    </tr>
    <tr>
      <td><tt>&lt;td></tt></td>
      <td><tt>&lt;/td></tt></td>
      <td>Encadrent une cellule de tableau</td>
    </tr>
    <tr>
      <td><tt>&lt;th></tt></td>
      <td><tt>&lt;/th></tt></td>
      <td>Encadrent une cellule de
        l'en-tête (<em>header</em>)</td>
    </tr>
  </tbody>
</table>
```

On peut disposer trois zones particulières au sein du tableau, marquées par `thead`, `tbody` et `tfoot`, pour marquer respectivement l'en-tête, le corps et le pied du tableau.

Quelques attributs permettent de modifier l'aspect des tableaux.

Attribut	Signification
<code><table border="3"></code>	Spécifier l'aspect des bordures (0 par défaut signifie pas de bordure).
<code><table width="100%"></code>	Définir la largeur du tableau par rapport à la fenêtre (en %) ou en nombre de pixels (à déconseiller)
<code><td rowspan="2"></code>	Prolonge la cellule courante sur autant de lignes que spécifié.
<code><td colspan="2"></code>	Prolonge la cellule courante sur autant de colonnes que spécifié.

Les attributs `border` et `width` seront avantageusement déplacés dans les feuilles de style.

2.3.5 Caractères spéciaux

Il est téméraire d'utiliser simplement les caractères accentués dans une page. En fonction des réglages du navigateur et du serveur Web, on peut arriver à des résultats curieux. Les caractères « èr » pourront donner par exemple un splendide idéogramme chinois.



La table de conversion suivante donne les principaux caractères accentués du français. La plupart des programmes d'éditeurs récents donnent un moyen d'opérer la conversion sans devoir les taper nous-mêmes.

â	ä	é	è	ê
<code>&acirc;</code>	<code>&auml;</code>	<code>&eacute;</code>	<code>&egrave;</code>	<code>&ecirc;</code>
ë	î	ï	ô	ù
<code>&euml;</code>	<code>&icirc;</code>	<code>&iuml;</code>	<code>&ocirc;</code>	<code>&ugrave;</code>
û	ü	ç	Ç	É
<code>&ucirc;</code>	<code>&uuml;</code>	<code>&ccedil;</code>	<code>&Ccedil;</code>	<code>&Eacute;</code>
<	>	espace		
<code>&lt;</code>	<code>&gt;</code>	<code>&nbsp;</code>		

S'il est relativement simple, bien que fastidieux, d'utiliser les entités HTML dans un texte fixe, lors de l'utilisation de données provenant d'une base, on devra adopter deux techniques :

- soit convertir les caractères avant leur affichage et, dans l'autre sens, avant leur stockage dans la base de données
- soit opter pour un encodage cohérent. Je conseille évidemment l'encodage UTF-8, tourné vers l'avenir. Cela suppose les points suivants :
 - définir des tables en encodage UTF8 au moment de leur création (les interfaces proposent souvent un autre encodage par défaut).
 - envoyer la commande `SET NAMES 'UTF8'` au serveur `mySQL` dès l'ouverture de toute nouvelle connexion
 - prévenir le navigateur qu'on utilise UTF8 par une instruction dans l'en-tête

```
<meta http-equiv="Content-type" content="text/html;_charset=utf-8" />
```

Moyennant ces précautions préalables, on peut oublier les problèmes d'encodage des caractères (nous verrons cependant que le problème des apostrophes et des guillemets n'est pas réglé pour autant).

2.3.6 Les images

Il existe plusieurs manières de placer des images dans une page. Je parlerai ici de l'image en tant qu'information. Les logos, les décorations doivent s'envisager comme des éléments de style et seront en conséquence placés dans les feuilles de style. Ces éléments sont en général présents dans toutes les pages.

HTML gère trois formats principaux :

- **jpeg** (extension `jpg`) : il s'agit d'un format compressé avec perte de qualité en fonction du coefficient de compression
- **gif** : limité à 256 couleurs, avec un taux de compression moindre que jpeg, mais sans perte de qualité. Gère la transparence. Longtemps rendu suspect par des problèmes de brevets, il est aujourd'hui tombé dans le domaine public.
- **png** : meilleur que gif (plus de couleur, mieux compressé), mais sa transparence est mal rendue par les anciennes version d'Internet Explorer, suite à un bug.

L'utilisation d'une image pose plusieurs problèmes : le temps de chargement reste un problème pour certains internautes qui ne disposent pas de DSL et les mal-voyants peuvent ne pas pouvoir les utiliser. En conséquence, on conseille :

- de prévoir des dimensions explicite afin que le moteur de rendu puisse réserver l'espace nécessaire au chargement de la page (attribut `width` et `height`)
- mettre un texte alternatif pour les mal-voyants et les navigateurs en mode texte, cette remarque est critique si l'image sert de lien (c'est l'attribut `alt` que certains navigateurs, IE par exemple, ont tendance à afficher comme bulle d'aide)
- si l'image sert de lien ou de bouton, on peut ajouter une bulle d'aide pour expliquer plus clairement à quoi elle sert (attribut `title`).

Voici un exemple complet :

```
<img source="soleil.png" alt="Image du soleil" title="Une photo du soleil"/>
```

Rappelons enfin que toute image a été créée et qu'il faut donc respecter les droits de son auteur. L'utilisation d'une image trouvée sur un autre site (même si on se contente de faire un lien sur elle sans la recopier) peut donner lieu à des poursuites judiciaires. On peut facilement trouver des images libres de droits (ou alors les créer soi-même).

2.3.7 Les liens hypertexte

Il suffit d'utiliser les balises `<a>`. Le texte à afficher figure entre les deux balises. L'URL est la valeur de l'attribut `href`.

```
<a href="html.html">Retour sur la premi&egrave;re page</a>
```

Le lien peut se faire à l'intérieur de la page. Il faut alors prévoir une *ancree* pour pouvoir y faire référence.

- créer une ancre nommée *Suite* dans une page :

```
<a name="suite"/>
```

- afficher un lien d'aspect Voir le tableau qui envoie le curseur à l'ancre *Suite* quand on clique dessus :

```
<a href="#Suite">Voir le tableau</a>
```

Insertion d'une image servant de lien

Il suffit de combiner un lien et une image.

```
<a href="html.html"></a>
```

Exemple complet

Le code suivant réalise un lien vers le site français de Firefox, en affichant une image placée dans un sous-répertoire du site, avec un message alternatif et une bulle d'aide.



```
<a href="http://mozilla-europe.org/fr/products/firefox/">
</a>
```

2.3.8 La mise en page du texte

Il est inutile de placer des espaces et des retours à la ligne dans une page Web, ils seront invariablement remplacés par une unique espace. On a longtemps utilisé des tableaux pour réaliser des mises en page sophistiquées. De même, certains nostalgiques continuent à utiliser des cadres (*frames*)³. Trois dispositifs permettent de réaliser un découpage harmonieux du texte.

Les balises <div> </div>

Il a été vu plus haut qu'elles autorisent une structuration hiérarchisée de la page (un <div> peut en contenir d'autres). On peut leur attribuer des styles dont vont hériter les différents éléments placés dans leur contexte. Ce type de balise provoque toujours une rupture et un retour à la ligne (la position exacte de la suite dépend des feuilles de styles). On accompagne généralement cette balise d'un attribut unique `id="xxxxyy"` ou d'un attribut `class="cYYYY"` pour lui donner des caractéristiques formelles.

Les balises <p> </p>

Elles délimitent un paragraphe, qu'on peut également doter de styles. Lorsqu'on publie un texte d'une certaine longueur, la mise en paragraphe se fera à l'aide de ces balises, plutôt que par l'emploi de la balise
, à réserver à des situations où le retour à la ligne doit se faire brutalement.

3. Leurs inconvénients sont multiples :

- ils morcellent la page en plusieurs entités différentes, dont le contenu sera mal analysé par les moteurs de recherches
- lorsqu'un cadre contient un lien vers un site externe, l'affichage de ce site se fait dans le contexte du cadre (une pratique illégale) et le titre de la page et l'URL ne sont pas modifiés.
- ils ne permettent pas de s'adapter à la taille de la fenêtre du navigateur.

Les balises `` ``

Comme `<div>`, `` délimite un contexte, mais cette fois, sans provoquer de retour à la ligne. On peut donc l'utiliser pour donner un aspect spécial à une suite de caractères. Associé à l'attribut `class`, elles permettent de réaliser un marquage qui dépasse la simple mise en page. Par exemple, un texte parlant de cinéma pourra utiliser cette balise pour distinguer les titres et les réalisateurs.

La sortie de `Jurassic Park`, réalisé par `Steven Spielberg` a marqué les esprits...

Tant qu'on n'utilise pas les styles, le texte marqué ainsi ne se distingue pas du reste.

2.4 Exercice pratique



2.4.1 Récupération des éléments

Vous pouvez vous servir des outils présents sur le site <http://yaml.de/en>. Je conseille de télécharger l'archive contenant « YAML "Simple Project" Template » version 1.0.9. On peut également utiliser Yaml Builder, mais il est plus difficile à mettre en œuvre. La photo des joueurs de football se trouve sur mon site à l'adresse suivante : <http://football.thoorens.net/images/football.jpg>.

2.4.2 Modification de la page

Suppression des éléments inutiles :

La page proposée par le site Yaml utilise trois colonnes, des instructions de login/contact/impressions. Je conseille d'utiliser le mécanisme des commentaires pour supprimer les éléments non désirés, afin de pouvoir les récupérer plus tard si nécessaire. Pour supprimer une section, il suffit de remplacer `<div>` par `<!--div>` et `</div>` par `</div-->`.

- supprimer les trois liens en haut de l'écran (ne pas confondre avec les deux liens vers le contenu destinés au navigateur en mode texte et au mal voyants).

- supprimer la colonne 2 (remarque, si on visualise la page, son espace reste présent à cause des feuilles de style).

Correction de la mise en page

Le travail doit être fait dans le fichier `css/screen/basemode.css`. Les styles liées à la colonne 2 peuvent être maintenus, ils seront ignorés. Corriger les points suivants :

```
#col1 { width:35%; }
#col1_content { padding:10px 10px 10px 20px; }

#col2 { width:0%; }
#col2_content { padding:10px 20px 10px 10px; }

#col3 { margin:0 0% 0 35%; }
#col3_content { padding:10px; }
```

- si on veut garder l'image des joueurs à sa taille normale, il faut élargir la colonne de gauche à 35% minimum et rétrécir d'autant le colonne du contenu (ex colonne 3).

Mise à jour du contenu

Apporter les modifications suivantes :

1. modifier le titre de la page dans la partie `<head>`
2. modifier le titre également dans l'en-tête. On notera qu'il est identique. La version finale en PHP réglera cette redondance en utilisant une variable.
3. modifier les étiquettes du menu



4. ajouter le texte principal (les trois liens peuvent renvoyer vers les fichiers `clubs.php`, `joueurs.php`, `matchs.php`).

Présentation

Ce site présente un championnat de Football imaginaire destiné à illustrer quelques techniques d'interface d'un site Web avec une base de données.

Le menu principal permet d'accéder aux informations par

- **club** : liste des clubs et liens vers la liste de leurs matchs.
- **joueur** : liste des joueurs et liens vers la liste de leurs performances (uniquement les buts marqués).
- **match** : liste des matchs, classés par date, avec liens vers quelques détails. de leurs matchs.