

# Programmation Web *HTML/CSS*

Rémy Malgouyres  
LIMOS UMR 6158, IUT, département info  
Université Clermont 1  
B.P. 86  
63172 AUBIERE cedex  
<http://malgouyres.org/>

Tous mes cours sur le *Web* sont sur le *Web* :

Cours de programmation *WEB* sur les documents hypertexte *HTML/CSS* :

<http://malgouyres.org/programmation-html-css>

Tutoriel sur le *CMS Drupal* :

<http://malgouyres.org/tutoriel-drupal>

Cours de programmation *WEB* côté serveur en *PHP* :

<http://malgouyres.org/programmation-php>

Cours de programmation *WEB* côté client en *JavaScript* :

<http://malgouyres.org/programmation-javascript>

Cours sur l'administration de serveurs (Serveurs *WEB* avec *apache*, *SSL*, *LDAP*...) :

<http://malgouyres.org/administration-reseau>

# Table des matières

Table des matières	1
<b>1 Pages web statiques <i>HTML5</i></b>	<b>2</b>
1.1 <i>HTML</i> , la norme et son évolution . . . . .	2
1.2 Validation W3C et tests de portabilité . . . . .	2
1.3 Structure d'un document <i>HTML</i> ou <i>XHTML</i> . . . . .	4
1.4 Premier document <i>HTML5</i> . . . . .	5
1.5 Structure du texte en <i>HTML</i> . . . . .	6
1.6 Mise en forme du texte <i>HTML</i> : styles <i>CSS</i> . . . . .	7
1.7 Formes d'inclusion de styles <i>CSS</i> . . . . .	12
1.8 Liens . . . . .	14
1.9 Tableaux . . . . .	17
1.10 Insertion d'images . . . . .	22
1.11 Figures et sous-figures avec légende . . . . .	24
1.12 Caractères spéciaux . . . . .	26
<b>2 Styles <i>CSS</i> et mise en page</b>	<b>27</b>
2.1 Éléments <i>HTML</i> de type <i>block</i> et <i>inline</i> . . . . .	27
2.2 Distinguer des parties dans un document : balise <code>div</code> . . . . .	31
2.3 Imbrication des balises et <i>CSS</i> . . . . .	32
2.4 Arborescence de balises et <i>CSS</i> . . . . .	34
2.5 Classes <i>CSS</i> . . . . .	36
2.6 Sélecteurs de style <i>CSS</i> par ID . . . . .	42
2.7 Marges et bordures . . . . .	45
2.8 Positionnement absolu . . . . .	46
2.9 Positionnement relatif . . . . .	48
2.10 Structuration d'une page en <i>HTML5</i> . . . . .	50
2.11 Exemples de mise en page . . . . .	52
2.12 <i>CSS</i> adaptatifs : <i>Media Queries</i> . . . . .	58

# Chapitre 1

## Pages web statiques *HTML5*

### 1.1 *HTML*, la norme et son évolution

Le langage *HTML*, ou *Hyper Text Markup Language*, permet de décrire et de mettre en forme des documents variés, depuis du simple texte jusqu'à des documents multimédia riches avec la définition d'*HTML 5*.

Une page *HTML* est en général destinée à être publiée sur le *World Wide Web*, où toutes sortes de gens utilisent toutes sortes de navigateurs qui fonctionnent sous différentes plateformes (Mac OS, iPad, Linux, Android, MS Windows, etc. pour citer les plus courants). Pour que cela fonctionne, il a été nécessaire de définir un standard pour le langage *HTML*. C'est le standard du *World Wide Web Consortium*, ou *W3C*. À noter le rôle déterminant qu'a joué la libération du code source du navigateur *Netscape* par la société *Netscape Communications Corporation* pour que la pluralité des acteurs n'évitent que les standards du web soient de fait propriétaire.

### 1.2 Validation W3C et tests de portabilité

#### 1.2.1 Valideur W3C

Lorsqu'on écrit une page web en *HTML*, pour être sûr que celle-ci soit correctement interprétée et affichée par tous les navigateurs qui supportent correctement la norme, il faut valider cette page pour s'assurer qu'elle est conforme à la norme. Le processus est similaire à l'analyse de la syntaxe d'un programme par un compilateur et peut se faire en ligne en *uploadant* le fichier ou en donnant son URL publique. Certains éditeurs proposent une validation interne ou même à la volée.

Les scripts en *PHP* et autre ne peuvent pas être directement validés car ce sont en fait des programmes. Par contre, leur sortie (ce qu'ils affichent) doit être du *HTML* conforme à la norme.



(a) L'upload d'un fichier HTML

(b) Résultat correct sur le validateur W3C

FIGURE 1.1 : Le processus de validation d'un fichier *HTML* sur le validateur *W3C*

### 1.2.2 Tests de portabilité

Même pour un document validé, les navigateurs n'ont pas tous la même implémentation du moteur d'analyse et de rendu du document, notamment en ce qui concerne les styles par défaut. Pour cette raison, il est indispensable avant de publier un site web, de le tester sur le plus possible de plate-formes et de navigateurs. Ceci peut être simplifié par des outils comme *VirtualBox* (mais il en existe d'autres...), qui permettent en virtualisant de faire tourner simultanément plusieurs systèmes d'exploitation sur un même ordinateur. Par exemple, sur la figure 1.2, on voit un exemples où l'affichage de la page est testé simultanément avec *Internet Explorer* et *Safari* sous *Windows*, *Firefox*, *Chrome* et *Opera* sous *linux* et *Chrome* sous tablette et smartphone *Android*. Le système *Windows* tourne en virtuel dans *VirtualBox* (disponible dans la logitèque d'*Ubuntu*). et le serveur *Web* (*Apache*) est installé en local sous *Ubuntu*. Les systèmes *Windows* et *Ubuntu* sont connectés par un réseau local virtuel interne au système hôte, en l'occurrence *Ubuntu*. Notons que les navigateurs sur smartphones sont particulièrement capricieux car ils n'ont pas les ressources pour implémenter toutes les variantes, surtout sur du code non validé.

FIGURE 1.2 : Le processus de test de portabilit  avec *VirtualBox*.

### 1.3 Structure d'un document *HTML* ou *XHTML*

Un document *HTML* ou *XHTML* doit comporter :

1. Sur la premi re ligne qui ne soit pas constitu e d'un commentaire : le **doctype** qui sp cifie le type de document (*HTML*, *XHTML* et version). Ceci permet que le navigateur puisse supporter et interpr ter plusieurs formats de documents.
2. Une balise de d but de description de document `<html [+attributs]>`.
3. Un en-t te compris dans une balise `<head>...</head>`.
4. Dans le *header*, une sp cification de l'*encoding* ou **charset** : *ISO-8859-1*, *latin1*, et maintenant syst matiquement **utf-8**. Ce dernier est d'ailleurs le d faut utilis  lorsque l'*encoding* n'est pas sp cifi .
5. Dans le *header*, de mani re optionnelle, une description du style de document (couleurs, polices, tailles, etc.) au format *CSS*.
6. Un corps du document dans une balise `<body>...</body>`.

Par exemple, le document “Hello World!” en *XHTML 1.0 strict* se compose comme suit :

exemples/ChapitreHTML/ex01\_helloWorldXhtml.html

```

1  /<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
2    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
3  <!-- Déclaration du debut d'un document HIML avec la langue : -->
4  <html xml:lang="en" lang="en" xmlns="http://www.w3.org/1999/xhtml">
5    <head> <!-- début de l'en-tête HIML -->
6      <!-- Déclaration du type d'encodage -->
7      <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
8      <!-- Titre de la page dans la fenêtre ou l'onglet du navigateur -->
9      <title>My first XHTML 1.0 Strict document</title>
10     </head>
11     <body>
12       <p>Hello world!</p>
13     </body>
14   </html>

```

Le document “Hello World!” en *HTML5* se compose comme suit :

exemples/ChapitreHTML/ex02\_helloWorldHtml5.html

```

1  </!doctype html>
2  <!-- Declaration du debut d'un document HIML avec la langue : -->
3  <html lang="fr">
4    <head>
5      <meta charset="UTF-8"/>
6      <!-- Déclaration du type d'encodage -->
7      <!-- Titre de la page dans la fenêtre ou l'onglet du navigateur -->
8      <title>My first HIML 5 document</title>
9    </head>
10   <body>
11     <p>
12       Hello world!
13     </p>
14   </body>
15 </html>

```

Le *XHTML 1.0 strict* possède une syntaxe plus stricte que ses prédécesseurs, simplifiant le travail des navigateurs et des moteurs de recherche. En particulier, toutes les balises sont en minuscules et il y a obligation de fermer les balises, quitte à mettre une balise auto-fermante comme `<br/>`. La norme *HTML5* réintroduit un certain laxisme au niveau de la syntaxe mais il est préférable pour la lisibilité du code de respecter la syntaxe *XHTML 1.0 strict* dans une page *HTML5*.

## 1.4 Premier document *HTML5*

exemples/ChapitreHTML/ex03\_corps\_balises.html

```

1  </!doctype html>
2  <!-- Declaration du debut d'un document HIML -->
3  <html lang="fr">
4    <!-- Voici l'en-tete qui declare les proprietes generales de la page -->
5    <head> <!-- debut de l'en-tete HIML -->

```