

XHTML

Thierry Lecroq

Université de Rouen
FRANCE

Plan

- 1 Généralités sur le web
- 2 Généralités sur XHTML
- 3 Les principaux éléments
- 4 Les liens
- 5 Les listes
- 6 Les images
- 7 Les tableaux
- 8 Les méta-informations

Arpanet

- 1959–68 : Le programme Arpa naît pendant la guerre froide
La peur d'une guerre nucléaire
 - ▶ Faiblesse du système centralisé versus distribué
 - ▶ Proposition d'un maillage d'ordinateurs (1964, P. Baran)
 - ▶ 1^{re} communication téléphonique entre 2 machines en 1965
- 1969 : Arpanet
1969 : 4 nœuds, 1971 : 15 nœuds, 1972 : 37 nœuds
- 1970–82 : Ouverture sur le monde
 - ▶ Apparition du courrier électronique
 - ▶ Communications internationales (Angleterre, Norvège)
 - ▶ Apparition de TCP/IP (1974) plus puissant que NCP
- 1983 : TCP/IP adopté comme standard Arpanet ⇒ Internet

Internet / World Wide Web

- 1983–89 : Expansion du réseau : autoroutes de l'information
 - ▶ La NSF (National Science Foundation) effectue des progrès importants (réseau NFSNET)
 - ▶ Réseaux hétérogènes (NCP et TCP/IP)
 - ▶ Utilisation importante par les scientifiques
 - ▶ Fin officielle de Arpanet en 1989 (TCP/IP)
- 1990–97 : L'explosion d'internet
 - ▶ 1990, le physicien Tim Berners Lee (CERN) étend le concept de lien hypertexte à Internet
 - ▶ HyperText Markup Language et HyperText Transfer Protocol
 - ▶ 1^{er} navigateur : NCSA Mosaic
 - ▶ 1995 ouverture au grand public (Netscape et Internet Explorer)
 - ▶ 1997 des dizaines de milliers de nœuds dans plus de 42 pays

Historique

Logiciels disponibles

Clients Netscape, Mozilla, Konqueror, Opera, Lynx, emacs, Internet Explorer

Serveurs Apache, Internet Information Server (Microsoft), iPlanet (Netscape)

Serveur	août 2007	%	septembre 2008	%
Apache	65153417	50,96	68228561	50,4
Microsoft	43861854	34,31	47232300	34,94
Google	5702456	4,46	6616713	4,90
Sun	2195495	1,72	2212821	1,64

Sources : <http://www.netcraft.com>

Principes de fonctionnement

La base du Web

Nécessité d'un protocole de communication

Principes de fonctionnement

Différents type de ressources

- Côté serveur
 - ▶ Statiques : HTML, image, son, applet, Javascript
 - ▶ Dynamiques : CGI, SSI, servlets, scripts serveur (php)
- Côté client
 - ▶ Statiques : HTML (3.2), images, son
 - ▶ Dynamiques : HTML 4, applet (Java), Javascript, Plugin, ActiveX

Principes de fonctionnement

URL : Uniform Resource Locator

- Spécification d'une ressource de manière unique

URN : Uniform Resource Name

- Mécanisme de nommage des ressources
- URN : `<Namespace>:<SpecificString>`
- Namespace : identificateur de nommage
- SpecificString : chaîne de caractères spécifique désignant la ressource de manière unique

URI : Uniform Resource Identifier

- $URI = URL + URN$
- En pratique, la forme d'URI la plus utilisée est l'URL

Principes de fonctionnement

URL : Uniform Ressource Locator

Format <protocole>://<serveur>:<port>/<chemin>/<ressource>

Remarque

Certains caractères doivent être encodés par % suivi de leur valeur hexadécimale en ISO Latin ou ASCII (ex : doc#2.html) doc

Exemple

```
http://www.linux-mandrake.com:80/fr/index.html
```

```
http://monge.univ-mvl.fr/~lecroq/index.html
```

```
ftp://ftp.debian.fr.org/
```

```
file://home/cours/
```

```
mailto:thierry.lecroq@univ-rouen.fr
```

```
telnet://user:password@host:port
```

Principes de fonctionnement

Fonctionnement d'un serveur HTTP

- Serveur : application qui écoute un port de communication
- Port standard : 80 (seul root peut écouter ce port)
- Serveur « maître » : utilisateur root écoute le port 80
- Serveurs « esclaves » : créés par le maître (propriétaire différent)
- Réception d'une requête :
 - 1 le maître reçoit la connection
 - 2 le maître crée un esclave et lui transmet le canal de
 - 3 communication l'esclave traite la requête et retourne le résultat

Principes de fonctionnement

Systèmes de fichiers

<code>http://<serveur>/</code>	racine du serveur
<code>http://<serveur>/fic.html</code>	fic.html à la racine
<code>http://<serveur>/cgi-bin</code>	répertoire des scripts CGI
<code>http://<serveur>/~arthur</code>	la page d'accueil de arthur
<code>http://<serveur>/~arthur/cgi-bin</code>	les scripts CGI de arthur
<code>http://<serveur>/cgi-bin/arthur</code>	les scripts CGI de arthur

Remarque

Interdiction d'accéder aux fichiers de configuration du serveur extérieurs au serveur WEB (système)

Affichage d'une page web

Un client (le navigateur Web) va chercher chez un serveur (sur le réseau, à l'adresse donnée par une URI) un programme (le texte HTML de la page concernée) qu'il exécute ensuite « en local »

Plan

- 1 Généralités sur le web
- 2 Généralités sur XHTML**
- 3 Les principaux éléments
- 4 Les liens
- 5 Les listes
- 6 Les images
- 7 Les tableaux
- 8 Les méta-informations

Généralités sur l'XHTML

eXtensible HyperText Markup Language

- SGML (Standard Generalized Markup Language) en 1986
- HTML (HyperText Markup Language) en 1992
- CSS (Cascading Style Sheets) en 1996
- XML (eXtensible Markup Language) en 1998
- HTML + XML = XHTML

Historique

- dans les années 1990 : HTML est créé par Tim Berner-Lee au Centre Européen de Recherche Nucléaire (CERN)
- 1995 : HTML 2.0 normalisation par l'IETF (Internet Engineering Task Force)
- 1996 : HTML 3.2 ajout des tables, des applets (Java), etc.
- 1998 : HTML 4.01 ajout des feuilles de styles, des frames, d'objets, etc.
- 2000 : XHTML 1.0 reformulation de HTML 4 en XML 1.0
- 2002 : XHTML 2.0 en cours de spécification

Normalisation par le W3C (World Wide Web Consortium

<http://www.w3c.org>) depuis 1996.

Généralités sur l'XHTML

eXtensible HyperText Markup Language

- Langage descriptif
- Fichier texte
- Pas de séquences de contrôle
- Description de la sémantique du document
- Balises
 - ▶ Balise ouvrante : `<tag>`
 - ▶ Balise fermante : `</tag>`
 - ▶ Les deux en une : `<tag/>`
- Standardisé
- W3C : <http://www.w3c.org>
- Dernière version : XHTML 2.0 en cours
- Validation automatique : <http://validator.w3.org>
- Balise `?xml` et `DOCTYPE`, sur les 2 premières lignes du fichier

Généralités sur l'XHTML

eXtensible HyperText Markup Language

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
```

Strict : inclue tous les éléments et attributs qui n'ont pas été déclarés caduques (deprecated) dans les versions précédentes

Transitional : inclue les éléments de la DTD strict et les éléments/attributs caduques

Frameset : inclue tous les éléments de la DTD transitional plus les éléments et attributs liés aux frames

XHTML 1.0 n'est qu'une reformulation de HTML 4 en tant qu'application XML 1.0

Principales différences

- les noms d'éléments et d'attributs doivent être en minuscule
- les valeurs d'attributs sont entre double quote
- un attribut a toujours une valeur
- toute balise ouverte doit être fermée
- tous les éléments doivent être imbriqués correctement : pas de chevauchement de balises

Éléments, balises et attributs

```
<nomElement attribut1="valeur1" attribut2="valeur2">
```

Contenu de l'élément

```
</nomElement>
```

- minuscule
- double quote

Généralités sur l'HTML

- Encapsulation de type « pile »
- (Mal)formation
 - ▶ correct : `<a> `
 - ▶ incorrect : `<a> `
- Structure arborescente
- Arbre minimal

```
<!DOCTYPE>  
<html>  
  <head>  
    <title/>  
  </head>  
  <body/>  
</html>
```

Généralités sur l'HTML

- Commentaires entre `<!--` et `-->`
- Balises toujours fermées
- Transformation en d'autres formats
- HTML donne un découpage selon la sémantique du document
- Mise en forme visuelle
- en HTML : moins lisible, plus de code
- en CSS : plus lisible, regroupement et généralisation

Plan

- 1 Généralités sur le web
- 2 Généralités sur XHTML
- 3 Les principaux éléments**
- 4 Les liens
- 5 Les listes
- 6 Les images
- 7 Les tableaux
- 8 Les méta-informations

Les attributs courants

`id` : identifiant unique (pour style ou Javascript, remplace `name`)
`class` : classe CSS
`title` : texte dans une bulle

Les attributs d'internationalisation

`xml:lang` : langue utilisée (code sur 2 caractères, ex : fr ou en)
`dir` : direction ltr ou rtl

Les attributs de gestion d'évènements

`onclick` : clic sur le contenu de l'élément

`ondblclick` : double clic sur le contenu de l'élément

`onkeydown` : maintien d'une touche enfoncée

`onkeypress` : frappe sur une touche

`onkeyup` : relâchement d'une touche enfoncée

`onmousedown` : enfoncement d'un bouton de la souris

`onmousemove` : le curseur de la souris bouge dans la zone de l'élément

`onmouseout` : le curseur de la souris quitte la zone de l'élément

`onmouseover` : le curseur de la souris est au dessus de la zone de l'élément

`onmouseup` : relâchement d'un bouton de la souris au dessus de la zone de l'élément

L'élément <body>

Les enfants

address, blockquote, del, div, dl, fieldset, form, h1, h2, h3, h4, h5, h6, hr, ins, ol, noscript, p, pre, script, table, ul

Pas de texte !!

Espaces

Tous les espaces blancs (y compris `\t` et `\n`) sont ignorés
Cas particulier : l'espace seul est reproduit tel quel

Exemple

```
Mon t e x t e a v e c d e s e s p a c e s  
e t p l u s i e u r s l i g n e s  
n ' a p p a r a i t p a s c o m m e c e c i
```

Résultat

```
Mon t e x t e a v e c d e s e s p a c e s e t p l u s i e u r s l i g n e s
```

⇒ Il faut utiliser des balises

Les accents et caractères spéciaux

- `á` : á
- `à` : à
- `â` : â
- `ä` : ä
- `ç` : ç
- `ã` : ã
- `æ` : æ
- `"` : ”
- `&` : &
- `<` : <
- `>` : >
- `€` : euro
- ` ` : insérer un espace

Mise en forme au niveau paragraphe

- `
` : aller à la ligne
- `<p>` et `</p>` : paragraphe
- `<div>` et `</div>` : division
- `<blockquote>` et `</blockquote>` : citation
- `<pre>` et `</pre>` : texte préformaté

Paragraphes

- l'élément de bloc `<p>` permet de construire des paragraphes et par un attribut `align` permet de spécifier le type de justification.
- l'élément `
` permet de contrôler les sauts de lignes.

Exemple

```
<p align="justify"> paragraphe paragraphe paragraphe  
paragraphe paragraphe paragraphe paragraphe paragraphe <br/>  
paragraphe paragraphe paragraphe paragraphe paragraphe  
paragraphe</p>
```

Mise en forme au niveau phrase

- `<abbr>` : abréviations
- `<acronym>` : acronyme
- `<cite>` : citation
- `<code>` : code
- `<dfn>` : définition
- `<kbd>` : keyboard
- `<q>` : citation
- `<samp>` : exemple
- `<var>` : variable

Mise en forme au niveau phrase

- `` : gras
- `<it>` : italique
- `` : mise en valeur (italique)
- `` : mise en valeur forte (gras)
- `<sup>` : exposant
- `<sub>` : indice
- `<big>` : plus gros
- `<small>` : plus petit

Plan

- 1 Généralités sur le web
- 2 Généralités sur XHTML
- 3 Les principaux éléments
- 4 Les liens**
- 5 Les listes
- 6 Les images
- 7 Les tableaux
- 8 Les méta-informations

Liens hypertexte

- Entre `<a>` et ``
- Choisir le(s) bon(s) mot(s) pour servir de lien
- Liens internes (``) ou externes (``)
- Possibilité de lien sur une image
- Lier les pages de manière cohérente

Liens hypertextes

Liens internes

- ``
- ``

Liens externes

- ``
- ``
- ``

Liens ciblés

- ``

Envoi de courrier

- ``
- ``

Plan

- 1 Généralités sur le web
- 2 Généralités sur XHTML
- 3 Les principaux éléments
- 4 Les liens
- 5 Les listes**
- 6 Les images
- 7 Les tableaux
- 8 Les méta-informations

Listes non-ordonnées

- `` marque le début de la liste et `` sa fin
- `` permet de commencer un nouvel item dans la liste et `` sa fin

Exemple

```
<ul>  
<li>un</li>  
<li>deux</li>  
<li>trois</li>  
</ul>
```

Listes ordonnées

- `` marque le début de la liste et `` sa fin
- `` permet de commencer un nouvel item dans la liste et `` sa fin

Exemple

```
<ol>
<li>un</li>
<li>deux</li>
<li>trois</li>
</ol>
```

Listes de définitions

- `<d1>` marque le début de la liste et `</d1>` sa fin
- `<dt>` permet de définir un titre pour une définition et `</dt>` sa fin
- `<dd>` permet de définir le contenu d'une définition et `</dd>` sa fin

Exemple

```
<d1>
<dt>Un</dt>
<dd>1</dd>
<dt>Deux</dt>
<dd>2</dd>
<dt>Trois</dt>
<dd>3</dd>
</d1>
```

Plan

- 1 Généralités sur le web
- 2 Généralités sur XHTML
- 3 Les principaux éléments
- 4 Les liens
- 5 Les listes
- 6 Les images**
- 7 Les tableaux
- 8 Les méta-informations

Insertion d'images

L'élément ``

Les attributs spécifiques

- `src` : source
- `alt` : message alternatif (obligatoire)
- `width` et `height` : hauteur et largeur
- `usemap` : pour les images cliquables

Les images réactives

L'élément `<map>`

- possède l'attribut `id` : lien avec l'attribut `usemap` de l'élément ``
- contient l'élément `<area/>`

Les images réactives

L'élément <area/>

Les attributs

- `accesskey` : raccourci clavier
- `alt` : message alternatif
- `href` : URI du document cible
- `shape` : définition de la zone sensible
4 valeurs
 - ▶ `rect` : rectangle (valeur par défaut)
 - ▶ `circle` : cercle
 - ▶ `poly` : polygone
 - ▶ `default` : zones non définies par ailleurs

L'élément `<area/>`

- `coords` : coordonnées de la zone
 - ▶ si `shape="rect"` : `coords="x,y,z,t"` avec (x,y) les coordonnées du coin supérieur gauche et (z,t) les coordonnées du coin inférieur droit
 - ▶ si `shape="circle"` : `coords="x,y,r"` avec (x,y) les coordonnées du centre et r le rayon
 - ▶ si `shape="poly"` : `coords="x0,y0,x1,y1,...,xn-1,yn-1,x0,y0"` avec les (x_i, y_i) les coordonnées des différents points du polygone

Plan

- 1 Généralités sur le web
- 2 Généralités sur XHTML
- 3 Les principaux éléments
- 4 Les liens
- 5 Les listes
- 6 Les images
- 7 Les tableaux**
- 8 Les méta-informations

Forme du tableau

- un tableau commence par `<table>` et finit par `</table>`
- une ligne dans un tableau commence par `<tr>` et finit par `</tr>`
- une cellule dans une ligne commence par `<td>` et finit par `</td>`
- une cellule d'en-tête dans une ligne commence par `<th>` et finit par `</th>`

Forme du tableau

Exemple

2 lignes × 3 colonnes

```
<table>
<tr>
<td>Cellule 1,1</td>
<td>Cellule 1,2</td>
<td>Cellule 1,3</td>
</tr>
<tr>
<td>Cellule 2,1</td>
<td>Cellule 2,2</td>
<td>Cellule 2,3</td>
</tr>
</table>
```

Forme du tableau

- On peut donner un « titre » au tableau entre `<caption>` et `</caption>`, en dehors des définitions de lignes ou de cellules
- On peut ranger les lignes entre `<thead>` et `</thead>` pour indiquer qu'elles font partie de l'en-tête
- On peut ranger les lignes entre `<tfoot>` et `</tfoot>` pour indiquer qu'elles font partie du pied du tableau
- On peut ranger les lignes entre `<tbody>` et `</tbody>` pour indiquer qu'elles font partie des données

Attention

`<thead>` et `<tfoot>` doivent apparaître avant `<tbody>`

Il peut y avoir plusieurs `<tbody>`

Forme du tableau

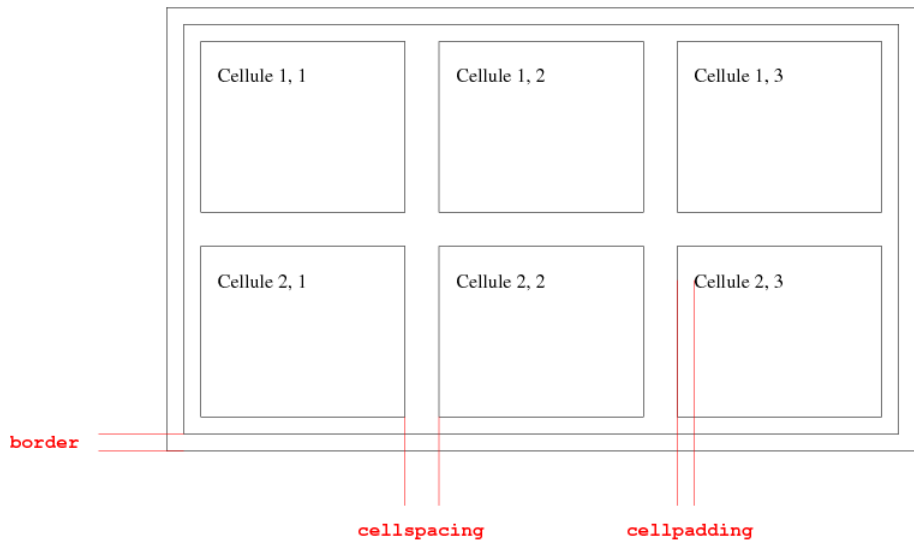
- On peut donner des indications de taille sur les colonnes entre `<colgroup>` et `</colgroup>`
- Entre ces balises, on indique avec `<col width=...>` la taille d'une colonne On peut aussi utiliser `<colgroup span=XX width=YY>` pour spécifier XX colonnes de taille YY
- Les balises `<td>` et `<th>` ont les attributs :
 - ▶ `rowspan` pour indiquer sur combien de lignes s'étant la cellule
 - ▶ `colspan` pour indiquer sur combien de colonnes s'étant la cellule
 - ▶ `align` pour indiquer l'alignement horizontal dans la cellule (`left`, `right` ou `center`)
 - ▶ `valign` pour indiquer l'alignement vertical dans la cellule (`top`, `middle`, `bottom`)

Bordures du tableau

`<table>` a les attributs :

- `border` : l'épaisseur des bordures
- `width` : largeur du tableau
- `cellpadding` : largeur de l'espace entre le contenu d'une cellule et sa bordure
- `cellspacing` : espace entre les bordures de chaque cellule (border doit avoir une valeur non nulle, border par défaut)
- `frame` : bordure externe du tableau
 - ▶ `void` : supprime toutes les bordures
 - ▶ `above` : ne laisse que la bordure supérieure
 - ▶ `below` : ne laisse que la bordure inférieure
 - ▶ `lhs` : ne laisse que la bordure gauche
 - ▶ `rhs` : ne laisse que la bordure droite
 - ▶ `hsides` : ne laisse que les bordures horizontales
 - ▶ `vsides` : ne laisse que les bordures verticales
 - ▶ `box` ou `border` : ne laisse que les 4 bordures externes

Bordures du tableau



Bordures du tableau

- `rules` : type de bordure entre les cellules
 - ▶ `none` : pas de bordure
 - ▶ `groups` : entre les groupes de lignes (`<thead>`, `<tfoot>`, `<tbody>`) et les groupes de colonnes (`<colgroup>`, `<col>`)
 - ▶ `rows` : entre les lignes uniquement
 - ▶ `cols` : entre les colonnes uniquement
 - ▶ `all` : toutes les cellules

Plan

- 1 Généralités sur le web
- 2 Généralités sur XHTML
- 3 Les principaux éléments
- 4 Les liens
- 5 Les listes
- 6 Les images
- 7 Les tableaux
- 8 Les méta-informations**

Les méta-informations

L'élément `<meta/>` (élément enfant de `<head>`) contient des informations qui ne sont pas visibles dans la page et qui sont destinées aux serveurs web, aux navigateurs et aux moteurs de recherche.

Syntaxe

```
<meta name="nom" content="valeur"/>
```

ou

```
<http-equiv="nom" content="valeur"/>
```

Différentes valeurs possibles de l'attribut name

`author` : auteur(s) de la page

`keywords` : mots-clés

`description` : description de la page

`robots` : différentes valeurs de content : none ou noindex, index, follow, nofollow

`revisit-after` : content="N days"

