

Outils de présentation

HTML - CSS

Florent Devin

EISTI

Introduction

Bibliographie

- ▶ <http://www.w3.org/>
- ▶ SÉBASTIEN LAOÛT : Tutoriel HTML et CSS

Schéma

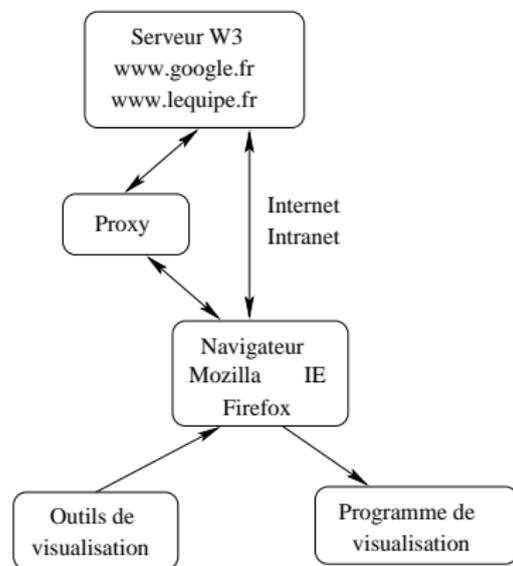


FIG.: Fonctionnement du Web

Présentation

- ▶ HyperText Transfert Protocol - TIM BERNERS LEE
- ▶ Protocole : adapté au multimédia
- ▶ Léger et rapide, à coût d'exploitation très bas.
- ▶ Dérive du protocole FTP
- ▶ Utilise TCP : port 80
- ▶ Devait être un protocole très simple

Principales requêtes

- ▶ GET : demander quelque chose
- ▶ POST : envoi de paramètres
- ▶ HEAD : demander non pas le document mais des informations le concernant
- ▶ PUT, DELETE, etc... : Utilisés par les outils d'auteurs

Différents serveur Web

- ▶ Apache (le plus répandu, Unix/NT)
- ▶ Microsoft Information Server (NT)
- ▶ Netscape Server (Unix, Windows NT)
- ▶ Jigsaw (java, portable, W3C)
- ▶ MacHTTP (Mac)
- ▶ ...

Serveur Apache

- ▶ Gratuit (license GNU)
- ▶ Modulaire
- ▶ Simple (quoique)
- ▶ Efficace (pas de *forking* furieux)
- ▶ Robuste (c'est le plus utilisé)
- ▶ Développement actif.

Rôle d'un serveur

- ▶ Transformation de l'URL en fichier ou en script
- ▶ Vérification d'identité
 - ▶ Le client est-il qui il prétend être ?
- ▶ Vérification d'accès
 - ▶ Le client est-il autorisé à effectuer cette requête ?
- ▶ Envoi de la réponse au client
- ▶ Mise à jour des fichiers de *logs*
- ▶ Traitements évolués (interprète perl, accès à des BDs, etc...)

Services rendus

- ▶ Servir une page en ajoutant des entêtes appropriés
- ▶ Servir une page telle quelle
 - ▶ Utile pour des pages à expiration rapide par exemple
- ▶ Exécuter un script et renvoyer le résultat comme une page
 - ▶ Interprétation de formulaires, ...
- ▶ Exécuter un script et renvoyer le résultat tel quel

Services rendus

- ▶ Interpréter des instructions
 - ▶ Langage de requête ASP
 - ▶ Javascript interprété au niveau serveur
 - ▶ Langage de requête pour Msql (module Apache)
 - ▶ Perl (module)

Possibilité

- ▶ Vérifier la machine d'origine (ou le dernier proxy)
 - ▶ Certaines autorisations
 - ▶ Redirection suivant la localisation géographique (foireux)
- ▶ Vérifier le mot de passe et le *login*
- ▶ Construire une page au vol avec des ordres spécifiques
 - ▶ Équivalents de `#include`

Possibilité

- ▶ Reconnaître les entêtes de requêtes et servir des pages différentes suivant le client ou ses préférences
 - ▶ Avoir des pages différentes selon le *browser*, la langue
- ▶ Lister des répertoires
- ▶ Encrypter les données et sécuriser les transactions

HTML

Historique

- ▶ Conçu par **TIM BERNERS-LEE** au **CERN** en **1990**
- ▶ Rendu populaire en **1993**
- ▶ Suivi de nombreuses extensions diverses
- ▶ Novembre 1995 : **HTML 2.0** défini par **IETF**
- ▶ Janvier 1997 : **HTML 3.2** défini par **W3C**
- ▶ Décembre 1997 : **HTML 4.0** défini par **W3C**
- ▶ **HTML+** (1993) et **HTML 3.0** (1995) sont non standards

Présentation

- ▶ Internationalisation
- ▶ Accessibilité (meilleur support d'agents non graphiques)
- ▶ Nouveau modèle de tableaux (meilleur contrôle de la structure et du formatage)
- ▶ Documents composés (mécanisme général de composition)

Présentation

- ▶ Feuilles de style (séparation structure/présentation)
- ▶ Scripting (documents dynamiques/interactifs)
- ▶ Impression (collections de documents)

Standard Generalized Markup Language

- ▶ Marquage de documents permettant d'enrichir le contenu :
 - ▶ Structure
 - ▶ Présentation
 - ▶ Sémantique
- ▶ HTML est un exemple de marquage

Exemple de document HTML

Structure

```
<!-- Type de document -->
<!DOCTYPE HTML PUBLIC
  "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <!-- En tête du document -->
  <head>

  </head>
  <!-- Fin de l'en tête -->
  <!-- Début du document -->
  <body>

  </body>
  <!-- Fin du document -->
</html>
```

Exemple de document HTML

Exemple : `html.html`

```
<!DOCTYPE HTML PUBLIC
  "-//W3C//DTD_HTML_4.01_Transitional//EN">
<html>
  <head>
    <title>Exemple de page web</title>
  </head>
  <body>
    <h1>Exemple de page web</h1>
    <hr />
    <address>
      <a href="mailto:florent.devin@eisti.fr">
        Devin Florent
      </a>
    </address>
  </body>
</html>
```

Exemple de document HTML



Présentation

- ▶ Façon de hiérarchiser l'information
- ▶ Possibilité de les imbriquer
- ▶ Concerne le fond du document
- ▶ Forme \Rightarrow CSS
- ▶ Deux types de balises :
 - ▶ Blocs (Vertical) symbolisé par \downarrow
 - ▶ Saut de ligne avant et après le bloc
 - ▶ Possibilité de mixer des balises blocs ou lignes
 - ▶ En ligne (Horizontal) symbolisé par \rightarrow
 - ▶ Pas de saut de ligne
 - ▶ Donner une signification particulière à un (groupe de) mot(s)

Exemple

- ▶ Un titre : utilisation de la balise `h1`
- ▶ Texte contenu dans la balise `h1`
- ▶ `<h1>Titre</h1>`

DOCTYPE

- ▶ Commence toujours par indiquer la version
- ▶ Puis balise `html`
- ▶ Deux parties
 - ▶ Balise `head` : entête. Contient des méta-données.
 - ▶ Balise `body` : corps. Contient le texte
- ▶ `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">`

Balises

- ▶ Tout texte doit être contenu dans une balise
- ▶ Trois types de paragraphe
 - ▶ ↓ **p** : paragraphe normal
 - ▶ ↓ **pre** : un paragraphe pré-formaté
 - ▶ → **br** : un saut de ligne

Exemple : paragraphe.html

`<p>` un paragraphe normal avec des espaces

un peu particulier pour `
` voir l'action des
différentes balises de paragraphes`</p>`

`<pre>`Il y a aussi un paragraphe pré`
` formaté
`</pre>`

Résultat



Balises

- ▶ Séparer les idées
- ▶ Deux types
 - ▶ ↓ Balise **h1** : Titre de page
 - ▶ ↓ Balise **h2**, **h3**, **h4**, **h5**, **h6** : section, sous-section,
...

Exemple : titre.html

Exemple

```
<h1>Les balises de titres</h1>  
<h2>Première section</h2>  
<h3>Sous section de la première section</h3>  
un peu de bla bla  
<h3>Seconde sous section de la première section</h3>  
<h4>Première sous sous section de la  
seconde sous section</h4>
```

Balises

- ▶ Trois types
 - ▶ ↓ Balise `ul` : liste non ordonnée
 - ▶ ↓ Balise `ol` : liste énumérée
 - ▶ ↓ Balise `dl` : liste de définitions
- ▶ Utilisation de la balise `li` pour les listes
↳ “classiques”
- ▶ ↓ Utilisation de la balise `dt` pour définir un terme
- ▶ ↓ Utilisation de la balise `dd` pour décrire le terme

Exemple : liste.html

Exemple

```
<ul>  
  <li>Premier niveau</li>  
  <li>un objet</li>  
  <ul>  
    <li>Second niveau</li>  
  </ul>  
  <li>un autre objet</li>  
</ul>
```

Balise

- ▶ Deux types de lien
 - ▶ Lien externe : ``
 - ▶ Lien interne : ``; Utilisation de #

Exemple : lien.html

```
<h2><a name="Premier"></a>Premier</h2>
bla bla bla
<h2><a name="Second"></a>Second</h2>
  Lien vers <a href="#premier">premier</a>
<br />
  Lien vers <a href="http://www.eisti.fr/">EISTI</a>
```

Images

Balise

- ▶ `img`
- ▶ Pas de balise fermante

Exemple : image.html

```
<p>Une image  
  
</p>  
<p>Une image avec un lien  
  <a href="http://www.eisti.fr">  
      
  </a>  
</p>
```

Remarques

- ▶ À utiliser au lieu d'utiliser des balises de mise en forme
- ▶ Donne un sens au texte

Balises

- ▶ → **sup** : Exposant
- ▶ → **sub** : Indice
- ▶ → **em** : Mise en évidence
- ▶ → **strong** : Mise en évidence forte

Balises les plus courantes

- ▶ → **abbr** : Signification d'une abbréviation
`<abbr title="baccalauréat">bac</abbr>`
- ▶ → **acronym** : Signification d'un acronyme
`<acronym title="École Internationale des Sciences du Traitement de l'Information">EISTI</acronym>`
- ▶ → **dfn** : Définition d'un terme
Une `<dfn title="Arbre de représentation...">ontologie</dfn>`

Balises les plus courantes

- ▶ → **cite** : Citation d'un ouvrage
Ce `<cite title="Super cours">cours</cite>...`
- ▶ → **q** : Citation d'un auteur
`<q title="Florent Devin">Il y a des jours, vaudrait mieux que ce soit des nuits</q>`

Balises les plus courantes

- ▶ → **code** : Exemple de code
- ▶ → **samp** : Exemple de sortie de programme
- ▶ → **kbd** : Signale le texte que doit saisir l'utilisateur
- ▶ → **var** : Signale une variable

Balises les plus courantes

- ▶ ↓ `hr` : Trait de séparation horizontal
- ▶ ↓ `blockquote` : Bloc de citation
- ▶ ↓ `address` : Signale une adresse

Balises

- ▶ ↓ **div** : Regroupe des objets afin de leur attribuer une mise en forme commune
- ▶ → **span** : Particularise un sous-élément pour y appliquer une mise en forme spéciale

Introduction

- ▶ À utiliser avec parcimonie
- ▶ Ne sert pas réaliser une présentation
- ▶ Utile si on a plusieurs données à présenter ensemble

Balises

- ▶ **table** : Déclaration d'un tableau
- ▶ **thead** : Déclaration de l'en-tête du tableau
- ▶ **tfoot** : Déclaration du pied du tableau
- ▶ **tbody** : Déclaration du corps du tableau
- ▶ **tr** : Déclaration d'une nouvelle ligne
- ▶ **td** : Déclaration d'une nouvelle cellule dans le corps du tableau
- ▶ **th** : Déclaration d'une nouvelle cellule ailleurs

Pourquoi ?

- ▶ Guider les moteurs de recherche
- ▶ "Influencer" les navigateurs
- ▶ "Commander" les robots
- ▶ Situées entre les balises `<head>` et `</head>`

Différents types d'informations

- ▶ Sur vous
- ▶ Sur vos pages
- ▶ Pour orienter les robots
- ▶ Pour rafraîchir
- ▶ Et d'autres ...

Auteur

- ▶ Permet de vous référencer sur un moteur de recherche
- ▶ `<meta name="author" lang="fr" content="Prénom NOM">`
- ▶ `name="author"` : Permet de préciser le champ Author
- ▶ `lang="fr"` : langue utilisée
- ▶ `content="Prénom NOM"` : auteur de la page

Publisher

- ▶ Permet de spécifier qui est le publisher, vous ou l'entreprise
- ▶ `<meta name="publisher" content="Prénom NOM">`

Exemple

```
<meta name="publisher" content="Florent  
DEVIN">
```

Adresse électronique

- ▶ Précise votre adresse électronique
- ▶ `<meta name="reply-to" content="Insérez ici votre adresse électronique et entre parenthèses vos prénom et nom.">`

Exemple

```
<meta name="reply-to"  
content="florent.devin@eisti.fr (Florent  
Devin)>
```

Description du site

- ▶ Une courte description du site
- ▶ `<meta name="description" content="Ici se trouvera votre courte description.">`

Exemple

```
<meta name="description" content="Site de  
présentation de la norme w3c">
```

Mots clefs

- ▶ Précise les mots clefs du site
- ▶ `<meta name="keywords" content="liste des mots clefs"`

Exemple

```
<meta name="keywords"  
content="HTML,CSS,Vulgarisation,tutoriel"
```

Adresse du site

- ▶ Spécifie l'adresse du site
- ▶ `<meta name="Indentifier-URL" CONTENT="Ici se trouvera l'adresse complète du site">`

Exemple

```
<meta name="Indentifier-URL"  
CONTENT="http ://test1.etude.pau.eisti.fr  
/eisti/">
```

Copyright

- ▶ Indique un copyright
- ▶ `<meta name="copyright" content="un copyright">`

Exemple

```
<meta name="copyright" content="©2005  
Florent Devin">
```

Logiciels utilisés

- ▶ Permet de préciser quels logiciels ont été utilisés
- ▶ `<meta name="generator" content="Listez ici à la façon des mots clés les différents logiciels utilisés.">`

Exemple

```
<meta name="generator"  
content="gimp,xemacs,beamer,xfig">
```

Date de création

- ▶ `<meta name="date-creation-yyyymmdd" content="La date de création">`

Exemple

```
<meta name="date-creation-yyyymmdd"
content="20050923">
```

Date de mise à jour

- ▶ `<meta name="date-revision-yyyymmdd" content="La date de mise à jour">`

Exemple

```
<meta name="date-revision-yyyymmdd"
content="20050924">
```

Jeu de caractères

- ▶ `<meta http-equiv="content-type"
content="text/html ; charset=ISO-8859-1">`

Orienter les robots

- ▶ Permet d'indiquer ce que les robots doivent indexer
- ▶ *All* : Indexe toutes les pages du site
- ▶ *None* : Indexe aucune page
- ▶ *Index* : Indexe la page
- ▶ *NoIndex* : N'indexe pas la page
- ▶ *Follow* : Indexe les liens
- ▶ *NoFollow* : N'indexe pas les liens

Orienter les robots

- ▶ `<meta name="robots" content="La
combinaison souhaitée">`

Exemple

```
<meta name="robots" content="index,nofollow">
```

Revisiter la page

- ▶ Demande aux robots de réindexer la page tous les n jours
- ▶ `<meta name="revisit-after" content="n">`

Exemple

```
<meta name="revisit-after" content="30">
```

Rafraîchissement de la page

- ▶ Demande aux navigateurs de rafraîchir la page
- ▶ `<meta http-equiv="refresh" content="n">`

Exemple

```
<meta http-equiv="refresh" content="30">
```

Interdire les caches

- ▶ `<meta http-equiv="pragma"
content="no-cache">`

Date d'expiration

- ▶ `<meta http-equiv="expires" content="Mon, 26 sep 2005 14 :00 :00">`

CSS

Présentation

- ▶ Sert à mettre en forme du code HTML
- ▶ Permet de spécifier précisément le style à appliquer
- ▶ Insertion de la balise `<link>`
- ▶ `<link rel="stylesheet" type="text/css" href="feuille de style">`
- ▶ Permet de différencier fond et forme
- ▶ Nécessité d'utiliser les bonnes balises
- ▶ Caractérisées par `id` et `class`

Différence

- ▶ Deux sélecteurs de classe : `id` et `class`
- ▶ `id` :
 - ▶ Doit être unique dans le document
 - ▶ Permet de remplacer les ancres nommées
- ▶ `class` :
 - ▶ Ce que l'on ne peut pas faire avec `id`

Bloc et en ligne

- ▶ Deux groupes de balises
- ▶ Bloc
 - ▶ `p`, `ul`, `div`, `h1`, ...
 - ▶ Positionnement : au dessus des uns des autres
- ▶ En ligne
 - ▶ `a`, `strong`, `em`, ...
 - ▶ Positionnement : côte à côte

Imbrication

- ▶ Bloc
 - ▶ Contient : bloc ou en ligne
 - ▶ Possibilité de dimensionner
- ▶ En ligne
 - ▶ Contient : ligne
 - ▶ Pas de dimension

Notion de bloc

- ▶ Permet de positionner des blocs les uns par rapport aux autres
- ▶ Boite invisible
- ▶ Configurable

Marges

- ▶ Par défaut : en haut, à gauche de son conteneur
- ▶ Prend toute la largeur du bloc
- ▶ Positionnement facile grâce aux marges

Exemple

```
p {  
  width :100px ;  
  height :100px ;  
  background-color :yellow ;  
  margin-left :15px ;  
  margin-top : 200px ;  
}
```

FLOAT

- ▶ Permet de positionner un bloc
- ▶ Sort du flux courant : pas de calcul de hauteur
- ▶ Souvent utilisé pour aligner une image
- ▶ Permet de positionner des blocs côte à côte

Positionnement flottant

Exemple

```
p{  
  float : left ;  
}
```

Présentation

- ▶ Permet difficilement le centrage
- ▶ Pas ou peu d'adaptation au résolution
- ▶ Seul moyen de superposer des éléments (*z-index*)

Utilisation

- ▶ Position relative ou absolu : bloc "positionné"
- ▶ Retiré du flux : position identique
- ▶ Relative : position dépendante du père
- ▶ Moyen de placement déroutant

Quelques conseils

- ▶ Éviter trop d'imbrication de `div`
- ▶ Ne pas utiliser les `table` pour la mise en page
- ▶ Utilisez les balises adéquates
- ▶ Utilisez les listes pour les menus

Attention à l'utilisation de `div`

```
<div class="gauche">
  <div class="menu">
    <ul>
      <li><a href="">Menu 1</a></li>
      <li><a href="">Menu 2</a></li>
    </ul>
  </div>
</div>
```

INUTILE

L'imbrication de `div` est inutile !

Il vaut mieux

```
<ul class="menu gauche">  
  <li><a href="">Menu 1</a></li>  
  <li><a href="">Menu 2</a></li>  
</ul>
```

Introduction

- ▶ Permet de préciser le style
- ▶ Plusieurs types de sélecteurs
- ▶ Du plus simple au complexe
- ▶ Toutes les conditions respectées \Rightarrow style appliqué
- ▶ Utilisation du sélecteur le plus précis

Sélecteurs universels

- ▶ * : Tout élément (peut être omis)

Exemple

- ▶ * { font-family : sans-serif ; }
- ▶ *.warning et .warning sont équivalents
- ▶ *#myid et #myid sont équivalents.

Sélecteurs de type

- ▶ A : Tout élément de type A

Exemple

- ▶ `h1 { font-family : sans-serif ; }`

Sélecteurs descendants

- ▶ "A B" : un élément B qui est un descendant arbitraire d'un élément A

Exemple

- ▶ `h1 em { color : blue ; }`

Sélecteur d'enfant

- ▶ $A > B$: Tous les éléments B enfant de A

Exemple

- ▶ `body > p { line-height : 1.3; }`
- ▶ `div ol>li p`

Sélecteurs d'enfants adjacents

- ▶ `A + B` : A et B ont le même parent et quand A précède immédiatement B.

Exemple

- ▶ `h1 + h2 { margin-top : -5mm ; }`
- ▶ `h1.opener + h2 { margin-top : -5mm }`

Syntaxe

- ▶ `[att]` : un élément a un attribut "att", quelle que soit sa valeur
- ▶ `[att=val]` : un élément a un attribut "att" dont la valeur est exactement "val"
- ▶ `[att~=val]` : un élément avec un attribut "att" qui contient au moins le mot val
- ▶ `[att|=val]` : un élément avec un attribut "att" qui commence par val

Exemple

- ▶ `h1[title] { color: blue; }` : les éléments `h1` qui possède un *titre*
- ▶ `span[class=exemple] { color: blue; }` : les éléments `span` dont l'attribut *class* vaut *exemple*
- ▶ `a[name~="titre"]` : les éléments `a` dont l'attribut *name* contient au moins *titre*
- ▶ `a[href|= "https"]` : les éléments `a` dont l'attribut *name* commence par *https*

Syntaxe

- ▶ `.truc` : Sélectionne les éléments qui ont l'attribut *class* égal à `truc`

Exemple

- ▶ `.pastoral { color: green; }` : tous les éléments avec `class="pastoral"`
- ▶ `h1.pastoral { color: green; }` : les éléments `h1` avec `class="pastoral"`

Syntaxe

- ▶ `#truc` : Sélectionne les éléments qui ont l'attribut *id* égal à `truc`

Exemple

- ▶ `#pastoral { color: green; }` : tous les éléments avec `id="pastoral"`
- ▶ `h1#pastoral { color: green; }` : les éléments `h1` avec `id="pastoral"`

Propriétés et pseudo-classes

- ▶ Voir la feuille distribuée

Attention

- ▶ Respecter une norme ne suffit pas :
 - ▶ Balises : valeur par défaut différentes selon les navigateurs
- ▶ Valider la syntaxe
 - ▶ Vérifie la syntaxe
 - ▶ Ne vérifie pas la sémantique

Index des balises

a, 32, 67

abbr, 38

acronym, 38

address, 41

author, 47

block

address, 41

blockquote, 41

body, 24

dd, 30

div, 42, 67, 69,
76–78

dl, 30

dt, 30

h1, 23, 28, 67

h2, 28

h3, 28

h4, 28

h5, 28

h6, 28

head, 24

hr, 41

html, 24

ol, 30

p, 25, 67

pre, 25

ul, 30, 67

blockquote, 41

body, 24

br, 25 

cite, 39
code, 40
dfn, 38
em, 37, 67
img, 34
kbd, 40
q, 39
samp, 40
span, 42
strong, 37, 67
sub, 37
sup, 37
var, 40

kbd, 40
keywords, 51

li, 30
link, 65

meta

author, 47
content-type, 57
copyright, 53
date-creation-
 yyyymmdd,
 55
date-revision-
 yyyymmdd,
 56
description, 50
expires, 63
generator, 54
Identifïer-URL, 52
keywords, 51
link, 65
pragma, 62
publisher, 48

refresh, 61
reply-to, 49
revisit-after, 60
robots, 59

ol, 30

p, 25, 67
pragma, 62
pre, 25
publisher, 48

q, 39

refresh, 61
reply-to, 49
revisit-after, 60
robots, 59

samp, 40

span, 42
strong, 37, 67
sub, 37
sup, 37

table, 44, 76
tbody, 44
td, 44
tfoot, 44
th, 44
thead, 44
tr, 44

ul, 30, 67

var, 40