

# L'accès aux bases de données sur internet

## 1. HTML - CSS

© Yann Viémont – 2009

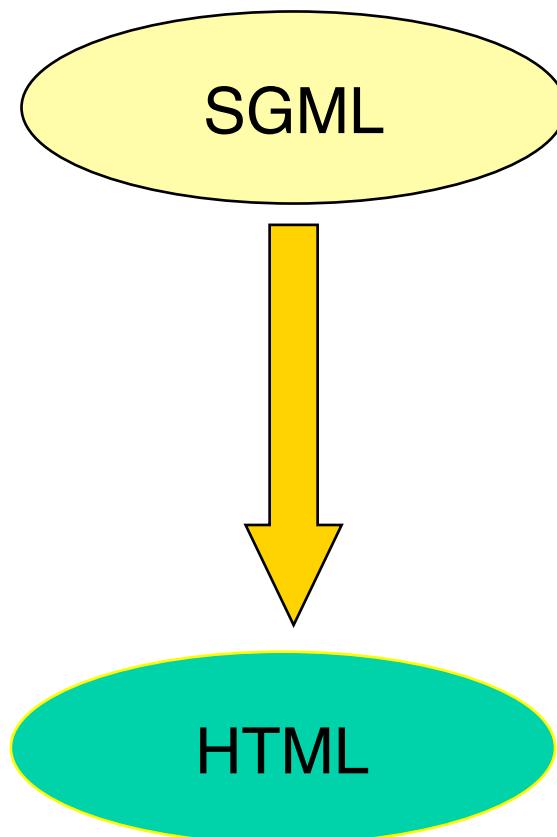
Université de Versailles St-Quentin

# Plan

1. [Introduction](#)
2. [Principales balises \(HTML 5\)](#)
3. [Hyperdocuments](#)
4. [Formulaires](#)
5. [Feuilles de style](#)
6. [Conclusion](#)

# Introduction

# Langages de balises



**Standard Generalized Markup Language**  
standard ISO 8879 1986

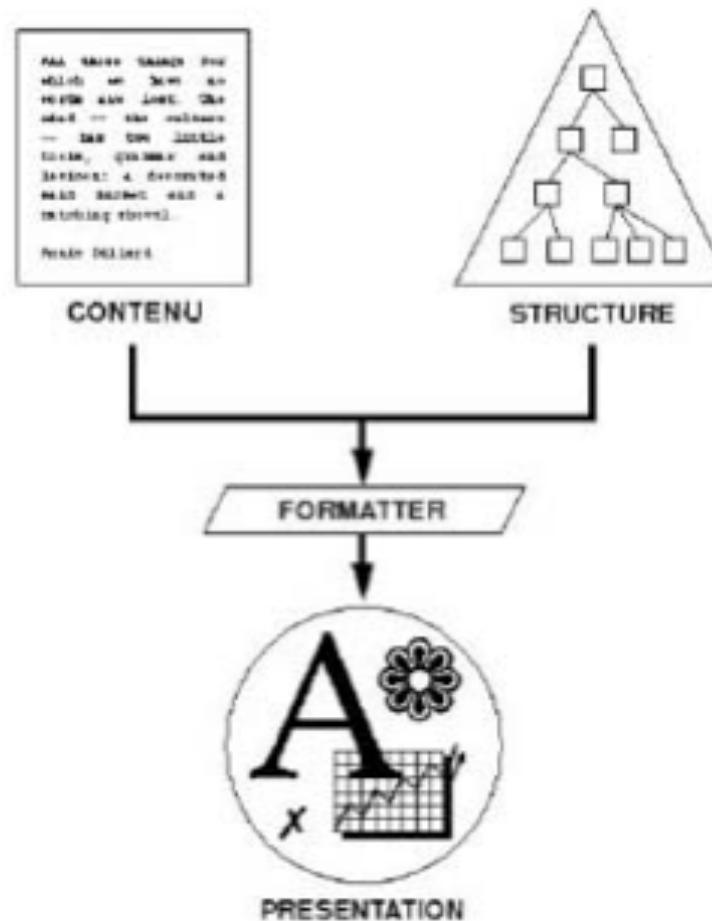
**HyperText Markup Language**  
Standard W3C - HTML 5 2009

# Composition d'un document SGML

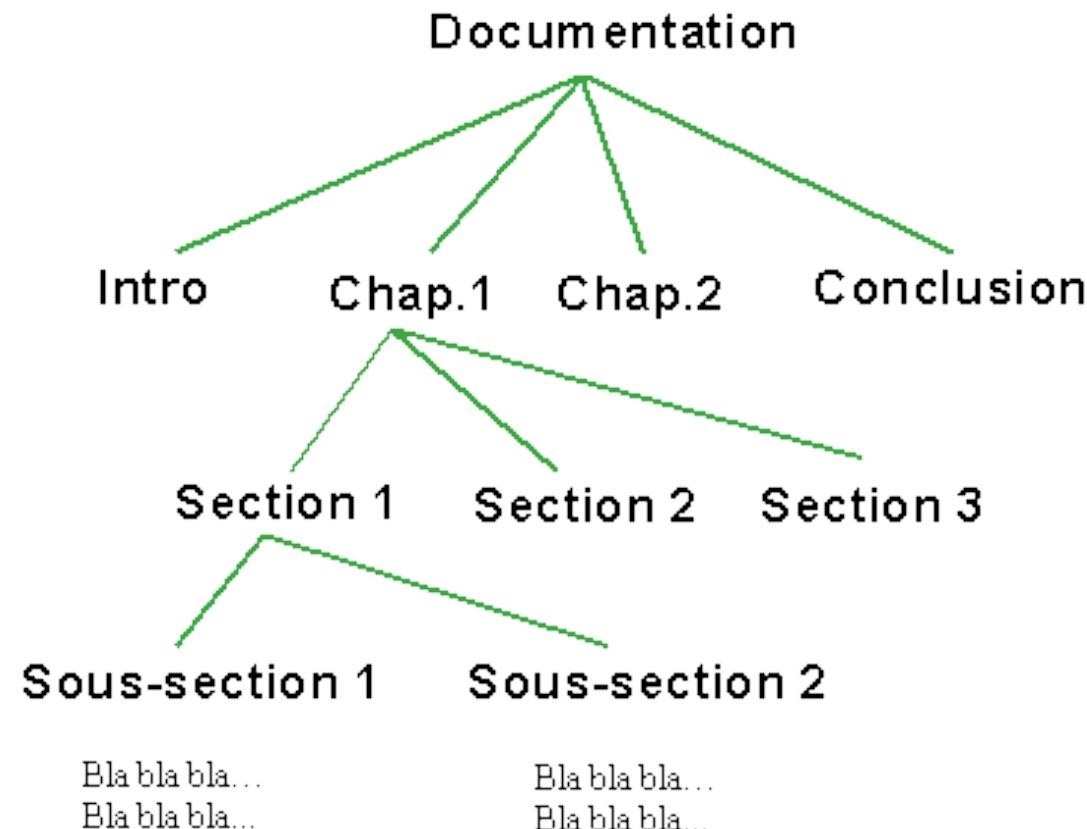
Trois parties :

- Le type - La **Définition de Type de Document (DTD)** décrit un modèle des documents
- Le contenu - Le document lui-même structure l'information selon sa nature
- La présentation - La feuille de style précise les options de mise en forme du document

# Séparer contenu / mise en forme



# Structuration du document



# Exemple

```
<anthologie>
  <poème>
    <titre>Villanelle</titre>
    <auteur>Alain Debroise</auteur>
    <strophe>
      <vers>Une feuille d'or,</vers >
      <vers>Une feuille rousse,</vers>
      <vers>Un frisson de mousse</vers>
      <vers>Sous le vent du nord.</vers>
    </strophe>
    <strophe>
      <vers>Quatres feuilles rousses,</vers>
      <vers>Quatre feuilles d'or,</vers>
      <vers>Le soleil s'endort</vers>
      <vers>Dans la brume douce.</vers>
    </strophe>
    <!-- suite des poèmes -->

  </anthologie>
```

# Principe des balises

Deux sortes de balises

## 1. Encadrement

<bal1> on place le contenu ici ...

autant qu'il en faut ...</bal1>

Une balise ouvrante + une balise fermante

## 2. Délimitation

<bal2> début du contenu ...

<bal2> suite du contenu ...

<bal2> et encore plus ...

Une seule balise qui ferme et ré-ouvre

# HTML

**Langage de balise simple dérivé de SGML :**

- **Pour les documents hypertexte internet**
  - **Pour les documents multi-média**
  - **Ouvre une porte sur les applications internet**
- 
- **HTML1 mélange contenu et présentation**
  - **XHTML basé sur XML en réaction**
  - **Convergence HTML 5 et XHTML 1**
  - **La présentation est confiée aux CSS**

# **Principales balises (HTML 5)**

# Premier exemple

```
<h1>Un titre de niveau 1</h1>
<p>Le premier paragraphe du document.<br>
Sur plusieurs lignes si on le souhaite.</p>
<h2>Un titre de niveau 2</h2>
<p>Un second paragraphe très court.</p>
<p> Jusqu'ici HTML sépare <em>correctement</em>
le contenu de la présentation</p>
```

# Résultat

## Un titre de niveau 1

Le premier paragraphe du document.  
Sur plusieurs lignes si on le souhaite.

## Un titre de niveau 2

Un second paragraphe très court.

Jusqu'ici HTML sépare *correctement* le contenu de la présentation

# Forme générale des balises

En général les balises HTML s'écrivent sous la forme :

`<balise attribut1="valeur1" attribut2="valeur2" ... >`

Objet décrit par la balise

`</balise>`

Il est possible d'imbriquer les balises :

`<balise1> <balise2> <balise3>`

objet décrit par les balises 1, 2 et 3

`</balise3> </balise2> </balise1>`

# Structure générale du document

```
<!DOCTYPE html PUBLIC  
"-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">  
<html>  
<head>  
    <title> Titre de mon document </title>  
</head>  
<body>
```

C'est ici que l'on met tout le contenu ...

```
</body>  
</html>
```

# Organisation en bloc

Le contenu d'une page est organisé en blocs :

- Il y a de nombreux blocs prédéfinis : **section**, **nav**, ...
- Un bloc peut être important
- Il peut regrouper de multiples contenus associés (par exemple des titres et des sections)

La balise générale **div** sert délimiter une division :

- Un bloc de contenu indéterminé
- Avec des propriétés communes

# Exemple de division

```
<div lang="de-DE"> <blockquote>  
Ich weiß nicht, was soll es bedeuten,<br>  
Daß ich so traurig bin, <br>  
Ein Märchen aus uralten Zeiten, <br>  
Das kommt mir nicht aus dem Sinn. <br>  
</blockquote> ... </div>
```

## Rendus possibles

Ich weiß nicht, was soll es bedeuten,  
Daß ich so traurig bin,  
Ein Märchen aus uralten Zeiten,  
Das kommt mir nicht aus dem Sinn.

...

Ich weiß nicht, was soll es bedeuten,  
Daß ich so traurig bin,  
Ein Märchen aus uralten Zeiten,  
Das kommt mir nicht aus dem Sinn.

... |

# Structuration logique du contenu

Balises servant à structurer le contenu d'une page :

**section** - Division principale d'un document

**nav** - Section contenant surtout des références

**article** - Document ayant une existence  
indépendante dans une page

**aside** - Document séparé ne faisant pas partie du  
document central

**header** - Contenu placé en tête d'une division

**footer** - Contenu placé en queue d'une division

**adresse** - Adresse de contact pour la page

# Les titres

On peut utiliser jusqu'à 6 niveaux de titres :

<h1>Un titre de niveau 1</h1>

<h2>Un titre de niveau 2</h2>

<h3>Un titre de niveau 3</h3>

...

**Un titre de niveau 1**

**Un titre de niveau 2**

**Un titre de niveau 3**

**Un titre de niveau 4**

**Un titre de niveau 5**

**Un titre de niveau 6**

# Paragraphes

Les paragraphes sont encadrés par : <p> ... </p>

Un simple retour à ligne est obtenu avec : <br>

Une séparation plus marquée avec : <hr>

Un paragraphe en retrait avec : <blockquote>

# Contenus particuliers - phrases

Texte mis en exergue : <**em**> ...</**em**>

Texte mis en évidence : <**strong**> ...</**strong**>

Confere : <**cite**> ... </**cite**>

Citation : <**q**> ... </**q**>

Définition : <**dfn**> ... </**dfn**>

Note dans le texte : <**small**> ... </**small**>

Marqué : <**mark**> ... </**mark**>

Code source : <**code**> ... </**code**>

Texte produit: <**samp**> ... </**samp**>

Texte entré : <**kbd**> ... </**kbd**>

Texte stylé : <**span**> ... </**span**>

# Exemple de phrase structurée

```
<p><strong>cryptage n. m.</strong><br>
<dfn>
Opération par laquelle est substitué, à un texte en clair,
un texte inintelligible, inexploitable pour quiconque ne possède
pas la clé permettant de le ramener à sa forme initiale.
</dfn></p>
```

# Rendu

## **cryptage n. m.**

*Opération par laquelle est substitué, à un texte en clair, un texte inintelligible, inexploitable pour quiconque ne possède pas la clé permettant de le ramener à sa forme initiale.*

# Contenus particuliers - mots

Acronyme : <**abbrabbr**>

Date et heure : <**time**> ... </**time**>

Variable (Maths) : <**var**> ... </**var**>

Exposant : <**sup**> ... </**sup**>

Indice : <**sub**> ... </**sub**>

Mesure : <**meter**> ... </**meter**>

# Exemples de mots

Acronyme :

l'`<acronym title="Organisation des Nations Unies">ONU</acronym>`  
comprend 192 états membres.`<br>`

Abréviation :

c'est un `<abbr title="fanatique">fan</abbr>` du chocolat. `<br>`

Date et heure :

il est `<time datatime="16:10">16h10</time>.<br>`

Variable (Maths) :

la variable `<var>n</var>` prend ses valeurs dans `<var>N</var>`,`<br>`

Exposant :

et pas dans `<var>R<sup>n</sup></var>.<br>`

Indice :

la suite `<var>X<sub>i</sub></var>` converge.`<br>`

Mesure :

il a sauté `<meter min=0 max=20 value=18.54>18.54 m</meter>`  
d'un seul bond.`<br>`

# Rendu

Acronyme : l'ONU comprend 192 états membres.

Abréviation : c'est un fan du chocolat.

Date et heure : il est 16h10.

Variable (Maths) : la variable  $n$  prend ses valeurs dans  $N$ ,

Exposant : et pas dans  $R^n$ .

Indice : la suite  $X_i$  converge.

Mesure : il a sauté 18.54 m d'un seul bond.

# Caractères spéciaux

- Signes : &copy;
- Pièges : &lt; &gt; &&
- Symboles : &not;
- Math : &frac34;
- Espace insécable : &nbs;
- Accents : &agrave;
- Diacritiques : &ampaelig



Signes : ©

Pièges : < > &

Symboles : ¬

Math : ¾

Espace insécable :

Accents : à

Diacritiques : æ

# Exemple avec paragraphes, phrases, mots, caractères

```
<p>Voici un exemple de définition tiré du  
<cite>Grand dictionnaire terminologique</cite> de l'Office  
québécois de la langue française&nbsp;:</p>  
  
<p><strong>cryptage n. m.</strong><br>  
<dfn>  
Opération par laquelle est substitué, à un texte en clair,  
un texte inintelligible, inexploitable pour quiconque ne possède  
pas la clé permettant de le ramener à sa forme initiale.  
</dfn></p>  
  
<p>La suite du texte continue normalement après la définition.  
Remarquer la balise &lt;strong&gt; pour mettre le nom à définir  
en évidence.</p>
```

# Rendu

Voici un exemple de définition tiré du *Grand dictionnaire terminologique* de l'Office québécois de la langue française :

**cryptage n. m.**

*Opération par laquelle est substitué, à un texte en clair, un texte inintelligible, inexploitable pour quiconque ne possède pas la clé permettant de le ramener à sa forme initiale.*

La suite du texte continue normalement après la définition.  
Remarquer la balise `<strong>` pour mettre le nom à définir en évidence.

# Listes

Il y a trois sortes de listes :

<ul>

<li>Les listes à point &lt;ul&gt; </li>

<li>Les énumérations, i.e. les listes numérotées &lt;ol&gt; </li>

<li>Les définitions, sur deux colonnes &lt;dl&gt; </li>

</ul>



Il y a trois sortes de listes :

- Les listes à point <ul>
- Les énumérations, i.e. les listes numérotées <ol>
- Les définitions, sur deux colonnes <dl>

# Définitions

Il y a trois sortes de listes :

<dl>

<dt>&lt;ul&gt;</dt> <dd> Les listes à point </dd>

<dt>&lt;ol&gt;</dt> <dd> Les énumérations, i.e.  
les listes numérotées</dd>

<dt>&lt;dl&gt;</dt> <dd>Les définitions sur deux  
colonnes</dd>

</dl>



Il y a trois sortes de listes :

<ul>

Les listes à point

<ol>

Les énumérations, i.e. les listes numérotées

<dl>

Les définitions sur deux colonnes

# Tables

```
<table>  
<tr><th>Année</th><th>Tués sur les routes</th></tr>  
<tr><td>2006</td><td align="right">4.709</td></tr>  
<tr><td>2007</td><td align="right">4.615</td></tr>  
<tr><td>2008</td><td align="right">4.443</td></tr>  
</table>
```



Année	Tués sur les routes
2006	4.709
2007	4.615
2008	4.443

# Texte pré-formaté

La balise **pre** permet de respecter le format du contenu :

```
<pre>
```

```
SELECT TITRE
```

```
FROM FILM
```

```
WHERE REALISATEUR = 'Danny Boyle'
```

```
</pre>
```



```
SELECT TITRE
```

```
FROM FILM
```

```
WHERE REALISATEUR = 'Danny Boyle'
```

# Code source

La balise **code** permet de présenter un programme :

```
<pre><code>
// Initialisation du tableau a -1
for (i = 0 ; i < 8 ; i++)
    for (j = 0 ; j < 8 ; j++)
        e [i][j] = -1 ;
</code></pre>
```



```
// Initialisation du tableau a -1
for (i = 0 ; i < 8 ; i++)
    for (j = 0 ; j < 8 ; j++)
        e [i][j] = -1 ;
```

# Figures

**Une figure comprend un contenu et une légende**

**Une figure peut être référencée**

**<p>La <a href="#6">figure</a> suivante montre un exemple de requête :</p>**

**<figure id="6">  
<dd><pre><code>**

**Select age from équipage where grade= 'Capitaine'**

**</code></pre></dd>**

**<dt>Exemple de requête SQL</dt>**

**</figure>**

# Rendu

La figure suivante montre un exemple de requête :

```
Select age from equipage where grade= 'Capitaine'
```

Exemple de requête SQL

# Hyperdocuments

# Navigation - hyperdocument

Définition : **structure hypertexte** (Bush 45)

- Organisation de l'information comme un réseau
- Les nœuds sont des fragments d'information
- Les arcs, appelés "liens" représentent les associations entre fragments

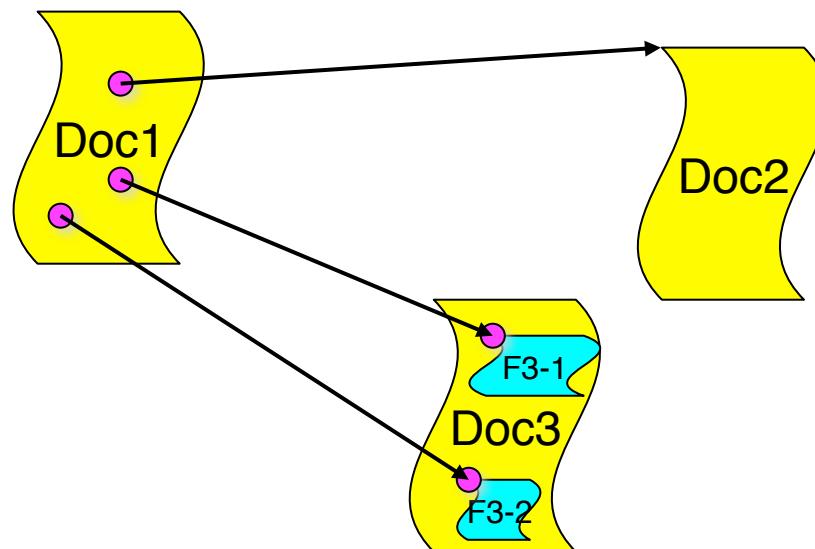
Ajout du multimédia → hyperdocument.

# Ancres et liens

Un lien relie deux **fragments** de documents

Chaque lien possède 2 ancrées :

- Ancre de départ du lien - la référence
- Ancre de destination du lien



# Ancre de départ (référence)

Une ancre de départ est insérée avec la balise :

```
<a href="ref-document">...</a>
```

L'attribut **href** désigne :

- Tout le document par son URI  
<http://www.w3.org/MarkUp/Guide/>
- Un fragment de documents par son URI et une ancre interne (ou signet)  
<http://fr.wikipedia.org/wiki/Alphabet#Histoire>

Signet

# Ancre de destination

Une ancre de destination (signet) est insérée :

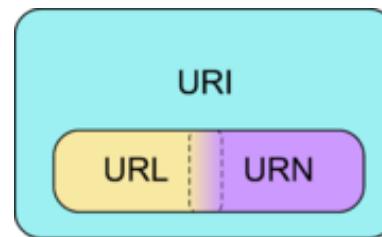
- Explicitement avec la balise :

```
<a name="nom-de-l-ancre">...</a>
```

- Implicitement avec l'attribut identifiant de n'importe quelle balise

```
<h2 id="nom-de-l-ancre">...</h2>
```

# URI : Uniform Resource Identifier



Un identifiant de ressource - URI - peut être soit :

- Un URL (Uniform Resource Locator) comme  
[http://fr.wikipedia.org/wiki/Uniform\\_Resource\\_Identifier](http://fr.wikipedia.org/wiki/Uniform_Resource_Identifier)
- Un URN (Uniform Resource Name) comme  
urn:mpeg:mpeg7:cs:VideoDomainCS:2001

# Exemple : table des matières

```
<div class="subtoc">
<p><strong>Contents</strong></p>
<ol>
<li><a href="#h-12.1">
    Introduction to links and anchors</a>
<li><a href="#h-12.2">
    The A element</a>
<li><a href="#h-12.3">
    Document relationships: the LINK element</a>
<li><a href="#h-12.4">
    Path information: the BASE element</a>
</ol>
</div>

<h2><a name="h-12.1">12.1 </a>
Introduction to links and anchors</h2>
<p>HTML offers many of the conventional publishing
idioms for rich text and structured documents, but
what separates it from most other markup languages is ...
</p>
```

# Rendu

## Contents

1. [Introduction to links and anchors](#)
2. [The A element](#)
3. [Document relationships: the LINK element](#)
4. [Path information: the BASE element](#)

## 12.1 Introduction to links and anchors

HTML offers many of the conventional publishing idioms for rich text and structured documents, but what separates it from most other markup languages is ...

# Multimédia

**HTML permet d'inclure des documents multimedia dans une page :**

- **Images**
- **Audios**
- **Vidéos**
- **Graphique (vecteur)**
- **Contenus actifs divers (e.g. flash)**

# Les images

```
<p>

Conçue en 1983, Set and Reset marque l'aboutissement du cycle
« Structures moléculaires instables » où le mouvement est
pensé comme un flux continu.</p>
<p>Sur la musique « technologique » de Laurie Anderson
(la compagne de Lou Reed pour la petite histoire), les sept
danseurs suivent leur propre trajectoire, se croisent ou
se rencontrent un instant seulement, trouvant dans la
percussion avec l'autre l'impulsion d'une nouvelle
direction à prendre.</p><br clear="all">
<p>Chez la chorégraphe américaine,
la danse néglige l'impact au profit de l'impulsion -
cette énergie à l'origine du mouvement.
Le premier ne semble exister que pour servir la seconde,
point d'appui au corps qui s'élanse, en conformité avec
les principes du <cite>Contact improvisation</cite> de
Steve Paxton, l'autre membre fondateur du Judson Dance
Theater dont la pratique irrigue en continue celle de
Trisha Brown.</p>
```

## Rendu



Conçue en 1983, Set and Reset marque l'aboutissement du cycle « Structures moléculaires instables » où le mouvement est pensé comme un flux continu.

Sur la musique « technologique » de Laurie Anderson (la compagne de Lou Reed pour la petite histoire), les sept danseurs suivent leur propre trajectoire, se croisent ou se rencontrent un instant seulement, trouvant dans la percussion avec l'autre l'impulsion d'une nouvelle direction à prendre.

Chez la chorégraphe américaine, la danse néglige l'impact au profit de l'impulsion - cette énergie à l'origine du mouvement. Le premier ne semble exister que pour servir la seconde, point d'appui au corps qui

# **Formulaires**

# Les formulaires HTML (FORMS)

**Un formulaire permet un dialogue avec le serveur :**

- Permet d'entrer des informations, d'exprimer ses choix, de saisir du texte
- Traité par un script tels que : JavaScript, PHP, ...
- Spécifie le protocole d'interaction
- Plusieurs formulaires possibles dans une même page web
- Mécanisme simple de création de site dynamique

# La balise <form>

La balise <**form**> ... </**form**> :

- Déclare un formulaire sur la page web
- Possède des attributs
- Peut inclure du contenu normal, y compris d'autres balises
- Encadre des contrôles spéciaux (composants) :
  - Zones de texte
  - Boutons
  - Listes de choix
  - Cases à cocher
  - Images actives
- Associe des étiquettes aux contrôles

# Spécification du protocole

La forme (presque) complète de la balise est :

```
<form id="..." method="..." action="..." enctype="...">
```

- **Identifiant (id)** : identifiant du formulaire.
- **Action (action)** : Spécifie la page qui se charge du traitement du formulaire.
- **Méthode (method)** :
  - GET - envoi des valeurs des champs avec l' URL
  - POST - envoi des valeurs dans un document encodé
- **Encodage (enctype)** : codage des données transmises.

# Les contrôles de la balise input

La balise `<input type="xxx" ...>` permet de créer les contrôles suivants:

- **text** : une zone de saisie de texte
- **password** : idem, mais sans écho
- **button** : un bouton (rustique) activant un script
- **checkbox** : une case à cocher
- **radio** : une sélection dans un choix
- **image** : une image associée à une correspondance
- **file** : une sélection de fichier lié
- **hidden** : un moyen de conserver du contexte
- **submit, reset** : la validation ou l'abandon

# Le contrôle texte

Syntaxe: <input type="text" ... >

Attributs:

- **name : nom du champ texte**
- **value : valeur par défaut**
- **size : taille en caractère de la zone de saisie**
- **maxlength : taille maximum du texte saisi**
- **tabindex : ordre de la navigation par tabulation**

Exemple :

```
<input type="text" name="loisir-prefere" size="20"  
tabindex="1">
```

# Le contrôle password

Syntaxe: <input type="password" ... >

Identique au contrôle texte mais avec un écho masquant

N'offre aucune sécurité - pas de cryptage

# Le contrôle bouton

Syntaxe: <input type="button" ... >

Attributs:

- **name** : nom du bouton
- **value** : valeur rendue et étiquette du bouton
- **tabindex** : ordre de la navigation par tabulation
- **Onclick** :

Exemple :

```
<input type="button" name="commentaire"  
      value="commentaire" onclick="entree-com()"  
      tabindex="9">
```

# Le contrôle checkbox

Permet de sélectionner (de cocher) une option

- Seules les options retenues sont transmises au serveur

Syntaxe: <**input type="checkbox"** ... >

Attributs:

- **name** : nom de l'option
- **selected** : état initial de la case à cocher
- **value** : valeur rendue
- **tabindex** : ordre de la navigation par tabulation

Exemple :

```
<input type="checkbox" name="fan-de" selected  
value="1" tabindex="8">
```

# Le contrôle radio

Permet de sélectionner (de cocher) une option unique parmi plusieurs (ou exclusif)

- Seule l'option retenue est transmise au serveur

Syntaxe: <**input type="radio"** ... >

Attributs:

- **name** : nom de l'option (partagé)
- **selected** : état initial de la case à cocher
- **value** : valeur rendue et étiquette de l'option
- **tabindex** : ordre de la navigation par tabulation

Exemple :

```
<input type="radio" name="niveau" selected value="
debutant" tabindex="1">
```

# Exemple : formulaire chanson

```
<h1>Entrez les informations sur la chanson.</h1>
<form><p>
Titre :
<input name="titre" size="40" tabindex="1" type="text"><br>
Interprete :
<input name="interprete" size="20" tabindex="2" type="text"><br>
Auteur :
<input name="auteur" size="20" tabindex="3" type="text"><br>
Compositeur :
<input name="compositeur" size="20" tabindex="4" type="text"><br>
Format :
<input name="format" value="MPEG" type="radio">MPEG
<input name="format" value="AAC" type="radio">AAC
<input name="format" value="AIFF" type="radio">AIFF
<input name="format" value="WAW" type="radio">WAW<br>
<input type="checkbox" name="fan" value="oui" selected tabindex="5">
Je suis fan.
</p></form>
```

# Rendu simple

**Entrez les informations sur la chanson.**

Titre : Ma philosophie

Interprete : Amel Bent

Auteur : Amel Bent

Compositeur : Blair Mackichan

Format :  MPEG  AAC  AIFF  WAW

Je suis fan.

## Exemple : format avec tableau

```
<h1>Entrez les informations sur la chanson.</h1>
<form><p><table cellpadding="6" border="0"><tbody>
<tr><td align="right">Titre : </td>
<td><input name="titre" size="40" tabindex="1" type="text"></td></tr>
<tr><td align="right">Interprete : </td>
<td><input name="interprete" size="20" tabindex="2" type="text"></td></tr>
<tr><td align="right">Auteur : </td>
<td><input name="auteur" size="20" tabindex="3" type="text"></td></tr>
<tr><td align="right">Compositeur : </td>
<td><input name="compositeur" size="20" tabindex="4" type="text"></td></tr>
<tr><td align="right">Format : </td>
<td><input name="format" value="MPEG" type="radio">MPEG
<input name="format" value="AAC" type="radio">AAC
<input name="format" value="AIFF" type="radio">AIFF
<input name="format" value="WAW" type="radio">WAW</td></tr>
<tr><td>&nbsp;</td><td>
<input type="checkbox" name="fan" value="oui" selected tabindex="5">
Je suis fan. </td></tr>
</tbody></table></p></form>
```

# Rendu avec tableau

**Entrez les informations sur la chanson.**

Titre :	<input type="text" value="Bad"/>
Interprete :	<input type="text" value="Michael Jackson"/>
Auteur :	<input type="text" value="Michael Jackson"/>
Compositeur :	<input type="text" value="Michael Jackson"/>
Format :	<input checked="" type="radio"/> MPEG <input type="radio"/> AAC <input type="radio"/> AIFF <input type="radio"/> WAW
<input checked="" type="checkbox"/> Je suis fan.	

# Les contrôles submit, reset

Ce sont des boutons à la sémantique prédéfinie

**submit :**

- Construit la liste des contrôles réussis
- Encode la liste
- Envoie la liste (get ou post)

**reset :**

- Réinitialise le formulaires avec les valeurs initiales

# Exemple de validation

```
<h1>Entrez les informations sur la chanson.</h1>
<form><p>
Titre :
<input name="titre" size="40" tabindex="1" type="text"><br>
Interprete :
<input name="interprete" size="20" tabindex="2" type="text"><br>
Auteur :
<input name="auteur" size="20" tabindex="3" type="text"><br>
Compositeur :
<input name="compositeur" size="20" tabindex="4" type="text"><br>
Format :
<input name="format" value="MPEG" type="radio">MPEG
<input name="format" value="AAC" type="radio">AAC
<input name="format" value="AIFF" type="radio">AIFF
<input name="format" value="WAW" type="radio">WAW<br>
<input type="checkbox" name="fan" value="oui" selected tabindex="5">
Je suis fan.<br>
<input value="Abandon" type="reset">
<input value="Validation" type="submit">
</p></form>
```

# Rendu

**Entrez les informations sur la chanson.**

Titre : Bridges

Interprete : Tracy Chapman

Auteur : Tracy Chapman

Compositeur : Tracy Chapman

Format :  MPEG  AAC  AIFF  WAW

Je suis fan.

Abandon

Validation

# Etiquettes

On peut juxtaposer du texte à côté des contrôles

On peut aussi utiliser une balise **label** :

- Pour associer une étiquette à un contrôle
- Pour contrôler le focus sur l'étiquette
- Pour rendre l'étiquette active (bouton)
- Pour spécifier la présentation des étiquette

Exemple :

```
<label for="titre">Titre : </label>  
<input id="titre" name="titre-chanson" size="40"  
      tabindex="1"><br>
```

# Exemple : source avec étiquettes

```
<h1>Entrez les informations sur la chanson.</h1>
<form id="saisie-chanson" action="http://nexistepas.org/musique.php" method="GET"><p>
<label for="titre">Titre : </label>
<input id="titre" name="titre" size="40" tabindex="1" type="text"><br>
<label for="interprete">Interprete : </label>
<input id="interprete" name="interprete" size="20" tabindex="2" type="text"><br>
<label for="auteur">Auteur : </label>
<input id="auteur" name="auteur" size="20" tabindex="3" type="text"><br>
<label for="compositeur">Compositeur : </label>
<input id="compositeur" name="compositeur" size="20" tabindex="4" type="text"><br>
Format :
<input name="format" value="MPEG" type="radio">MPEG
<input name="format" value="AAC" type="radio">AAC
<input name="format" value="AIFF" type="radio">AIFF
<input name="format" value="WAW" type="radio">WAW<br>
<input id="fan" type="checkbox" name="fan" value="oui" selected tabindex="5">
<label for="fan"> Je suis fan.<br>
<input value="Abandon" type="reset">
<input value="Validation" type="submit">
</p></form>
```

# Select / Option

La balise **select** permet de créer un menu déroulant :

- Il se comporte comme une sélection dans une liste
- Chaque option est introduite par la balise **option**
- Une option peut être pré-sélectionnée
- La sélection peut être multiple

Exemple :

```
<select id="genre" name="genre" size="1"  
        tabindex="5">  
  
<option value="1">Alternatif
```

## Exemple : genre des chansons

```
<input id="compositeur" name="compositeur" size="20" tabindex="1">
<label for="genre"> Genre :
<select id="genre" name="genre" size="1" tabindex="5">
  <option value="1">Alternatif
  <option value="2">Classique
  <option value="3">Disco
  <option value="4">Electro
  <option value="5">Folk
  <option value="6">Grunge
  <option value="7">Hard rock
  <option value="8">Latino
  <option value="9" selected>Pop
  <option value="10">Rap
  <option value="11">Rithm'n Blues
  <option value="12">Rock
  <option value="13">Ska
  <option value="14">World
  <option value="15">Zouk
</select><br>
Format :
<input id="mpeg" name="format" value="MPEG" type="radio">
<label for="mpeg"> MPEG </label>
<input id="aac" name="format" value="AAC" type="radio">
<label for="aac"> AAC </label>
```

# Rendu : menu déroulé

**Entrez les informations sur la chanson.**

Titre : Bridges

Interprete : Tracy Chapman

Auteur : Tracy Chapman

Compositeur : Tracy Chapman

Genre : Folk

Format : Alternatif

Classique

Disco

Electro

Folk

Grunge

Hard rock

Latino

Pop

Rap

Rithm'n Blues

Rock

Ska

World

Zouk

Je suis

Abandon

AAC  AIFF  WAW

# Soumission par la méthode GET

Avec la balise **form** suivante :

```
<form id="saisie-chanson" action="http://  
nexistepas.org/musique.php" method="GET">
```

Le navigateur construit l'URI suivante

<http://nexistepas.org/musique.php?titre=Bridges&interprete=Tracy+Chapman&auteur=Tracy+Chapman&compositeur=Tracy+Chapman&genre=5&format=MPEG&fan=oui>

## Reste à voir

- **button**
- **textarea**
- **fieldset legend**
- **focus**

# Feuilles de styles

# HTML en mode laxiste

**HTML est très laxiste sur les documents autorisés :**

- Il hérite de son passé - les tout débuts du www
- Il doit rester compatible avec les anciennes versions des navigateurs encore en usage
- Il doit rester compatible avec les outils d'écriture de pages qui génèrent du HTML *exotique*
- Il doit permettre à des débutants de faire des pages personnelles à la main

# Mélange contenu / présentation

```
<body topmargin=20 leftmargin=30 rightmargin=20>
<h1><font face="verdana" color="blue" size="5">
Un titre de niveau 1 en bleu</font></h1>
<font face="arial" size="4">
<p>Le premier paragraphe du document commence avec le
style de base sans modification. Une autre phrase avec
<i><font color="grey">presque</font></i> le même
style.</p>
<p align="center">Un second paragraphe <b>vraiment</b>
très court et centré.</p>
<p style="border: solid; border-color: rgb(125, 125, 125);
background-color: rgb(245, 245, 245); padding: 0.5em;">
On peut dire que HTML <b>n'oblige pas</b> à séparer le
contenu de la présentation&nbsp;; le texte est mis en
forme <i><font color="grey">au flot</font></i>. </p>
</font></body>
```

# Rendu

## Un titre de niveau 1 en bleu

Le premier paragraphe du document commence avec le style de base sans modification. Une autre phrase avec *presque* le même style.

Un second paragraphe **vraiment** très court et centré.

On peut dire que HTML **n'oblige pas** à séparer le contenu de la présentation : le texte est mis en forme *au flot*.

# HTML en mode bonne pratique

**HTML 5 + "bonne pratique" -->**

- **Contenu du document séparé de sa présentation**
- **Styles à part dans l'en-tête ou dans un fichier séparé**
- **Syntaxe claire et puissante pour spécifier la présentation**

# Séparation des styles

```
<title>Séparation des styles</title>
<style type="text/css">
    body
    {
        font-family: Arial, sans-serif;
        font-size: large;
        margin: 20px 50px 0 40px;
    }
    h1
    {
        font-family: Verdana, sans-serif;
        font-size: x-large;
        color: blue;
    }
    em { color: grey; }
    .important { text-align: center; }
    #conclusion
    {
        border: solid;
        border-color: rgb(125, 125, 125);
        background-color: rgb(245, 245, 245);
        padding: 0.5em;
    }
</style>
</head>
```

# Nettoyage du contenu

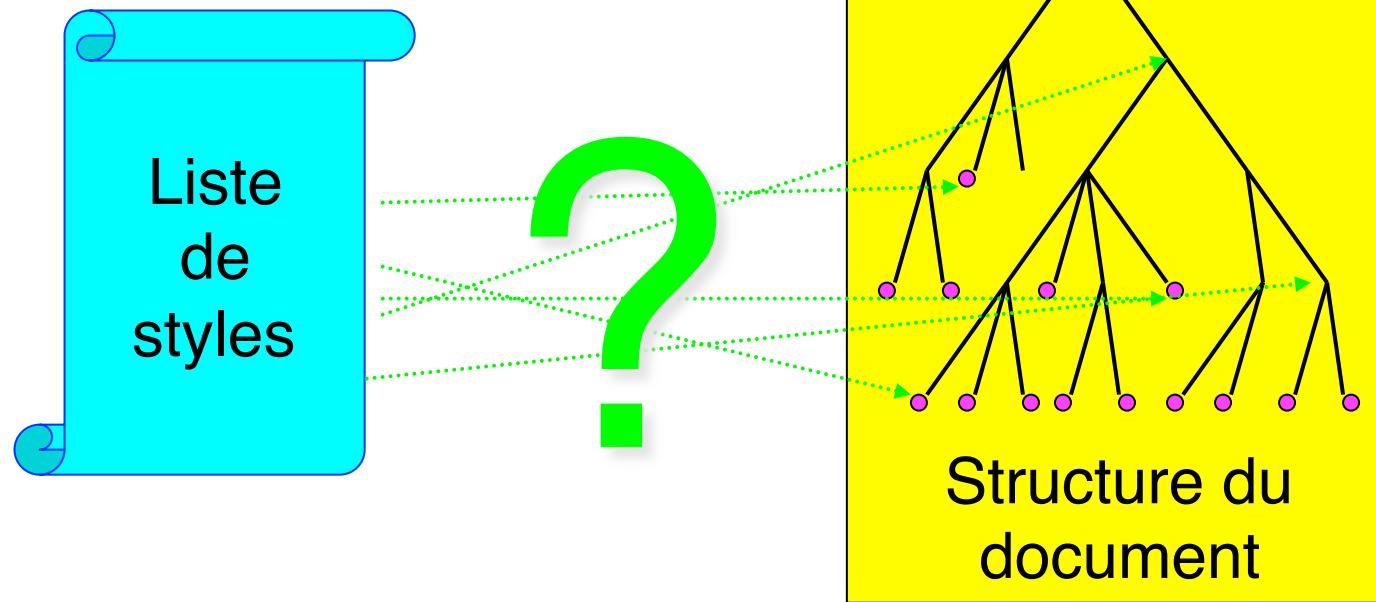
```
<body>
  <h1>Un titre de niveau 1 en bleu</h1>
  <p>Le premier paragraphe du document commence avec le
  style de base sans modification. Une autre phrase avec
  <em>presque</em> le même style.</p>
  <p class="important">Un second paragraphe
  <strong>vraiment</strong> très court et centré.</p>
  <p id="conclusion">On peut dire que HTML
  <strong>n'oblige pas</strong> à séparer le contenu de
  la présentation : le texte est mis en forme
  <em>au flot</em>.</p>
</body>
```

# Etablir la correspondance

**Une page web efficace doit être bien présentée**

- Comment associer la présentation au contenu de façon efficace
- Les balises HTML ne couvrent pas tous les besoins de structuration dans le moindre détail
- Il faut pouvoir traiter cas particuliers et exceptions
- Besoin d'un modèle suffisamment riche

# Correspondance



Comment établir la correspondance style / composant ?

# A quoi associer un style ?

Solutions possibles pour les éléments ciblés :

- Un type de balise
- Une occurrence de balise (un objet particulier dans l'arbre)
- Une annotation dans l'arbre
- Une règle liée à la structure d'un sous-arbre
- Une autre règle
- ...

# S1 - Un type de balise

```
body
{
    font-family: Arial, sans-serif;
    font-size: large;
    margin: 20px 30px 0 20px;
}

h1
{
    font-family: Verdana, sans-serif;
    font-size: x-large;
    color: blue;
}

em { color: grey; }
```

## S1 - Bilan

### Avantages :

- Très simple
- Séparation complète contenu/présentation

### Inconvénients :

- Nombre de balises limité -  
HTML ne peut pas prévoir tous les cas
- Le contexte structurel (la position dans le document) n'est pas pris en compte
- Tous les paragraphes <p> ... </p>  
formatés à l'identique

## S2 - Une occurrence de balise

```
#conclusion
{
    border: solid;
    border-color: rgb(125, 125, 125);
    background-color: rgb(245, 245, 245);
    padding: 0.5em;
}
```

## S2 - Bilan

### Avantages :

- Très complet et surtout très détaillé
- Bonne séparation

### Inconvénients :

- Oblige à nommer tous les objets de l'arbre
- Trop détaillé - aucune factorisation
- La description de la correspondance change avec la structure du document

## S3 - Une annotation dans l'arbre

```
.important { text-align: center; }
```

## S3 - Bilan

### Avantages :

- Contrôle total
- Choix du niveau de détail variable

### Inconvénients :

- Séparation faible contenu / présentation
- Grande redondance

## S4 - Une règle liée à la structure

```
h1 > p:first-child
{
    color : blue;
    font-style : italic;
}
```

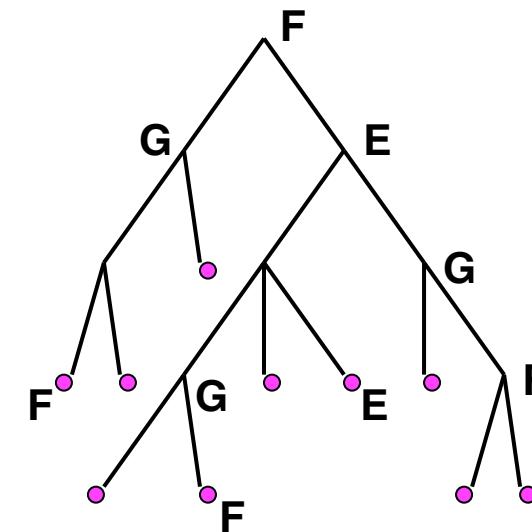
# Règles contextuelles

Si E, F, G sont des types de balise

- E F → F descendant de E
- E > G → G fils de E
- G + E → E frère suivant de G
- G:first-child → G premier fils

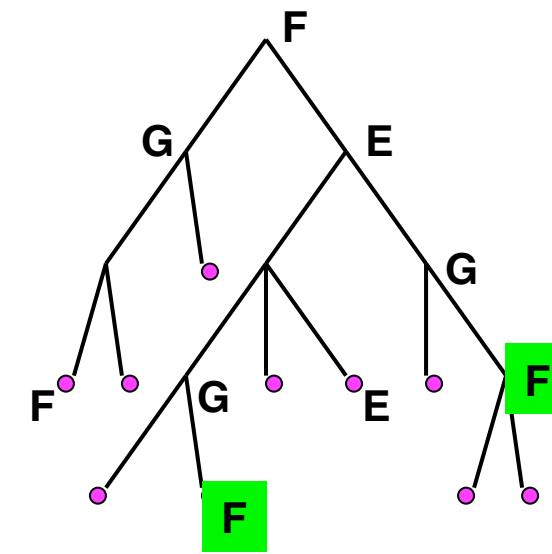
Et de nombreux autres avec \*

- E > \* > G → G petit-fils de E



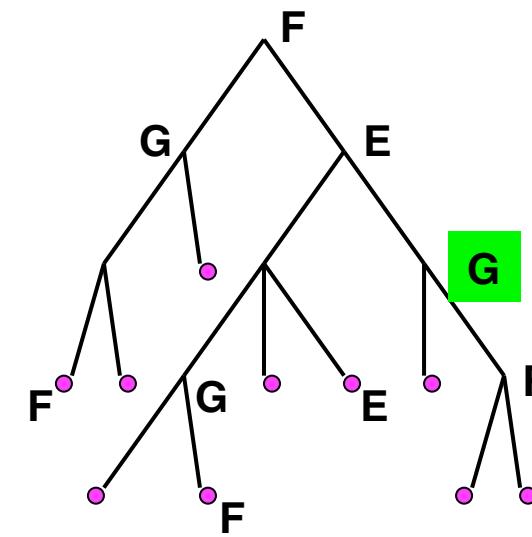
# Descendant de

E F



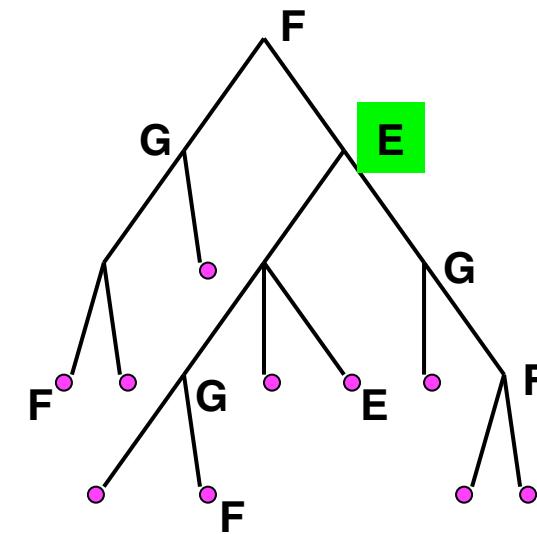
# Fils de

E > G



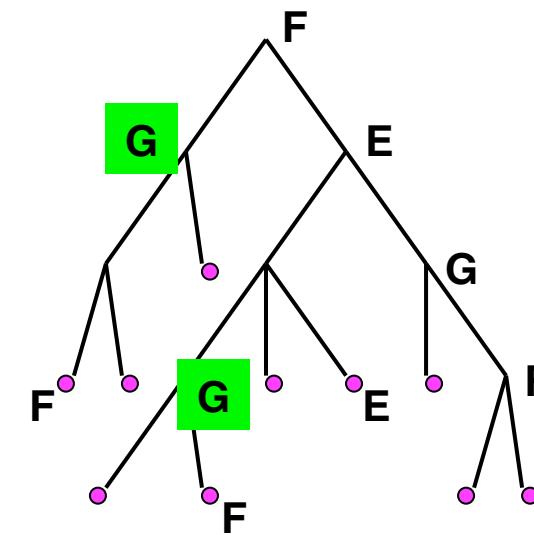
# Frère suivant de

G + E



# Premier fils

G:first-child



## S4 - Bilan

### Avantages :

- Contrôle lié au contexte d'une balise
- Séparation complète contenu/présentation
- Choix du niveau de détail variable

### Inconvénients :

- Très (trop ?) compliqué pour un utilisateur
- Niveau métadonnées (grammaire)

# Solution CSS

**Les CSS permettent de mettre en œuvre les 4 solutions simultanément (selon les besoins)**

**S1 - Un type de balise**

- Règles de base - valeurs par défaut

**S2 - Une occurrence de balise**

- Traitement des exceptions

**S3 - Une annotation dans l'arbre**

- Règles de base - valeurs par défaut

**S4 - Une règle liée à la structure**

- Règles liées au contexte

## Exemple : style des tables

```
table { empty-cells: show; }

table, tbody, td, th
{
    border: solid;
    border-color: rgb(125, 125, 125);
    border-collapse: collapse;
    background-color: rgb(245, 245, 245);
    padding: 0.5em;
}

table, tbody { border-width: 2px; }

td, th { border-width: 1px; }

th { border-bottom-width: 2px; }

td.numerique { text-align: right; }

caption { caption-side: bottom; }
```

## Exemple : classe de contenu

```
.sql
{
    font-family: monaco, monospace;
    font-size: smaller;
    color: blue;
}
```

# Exemple : contrôle des flottants

```
.flottant, .flottant-gauche
{
    float: left;
    margin: 0.2em 1em 0.3em 0.3em;
}

.flottant-droit
{
    float: right;
    margin: 0.2em 0.3em 0.3em 1em;
}

.rupture { clear: both; }
```

# Exemple : liste de définition sur une seule ligne

```
.meme-ligne { display: inline; }

dd.meme-ligne { margin-left: 1.5em; }

dd.meme-ligne:after
{
    content: "\A";
    white-space: pre-line;
}
```

# Exemple : création d'une balise

```
bal {color: blue; }  
bal:before { content: "<"; }  
bal:after { content: ">"; }
```

## La création de nouvelles balises :

- Apporte de nombreuses possibilités
- Donne une plus grande souplesse
- Est un mécanisme simple

## Mais :

- Cache la sémantique aux outils
- N'est pas supportée par tous les navigateurs

## Reste à voir

- Détail des propriétés de style
  - Fonte, couleur, fond, bordure
  - Marge, taille, espacement, indentation
  - Positionnement horizontal et vertical
  - Tabulation, flottant
- Numérotation automatique
- Cascade
- Héritage des propriétés
- Modèle de boite - Visualisation
- Paramétrage selon le média de présentation

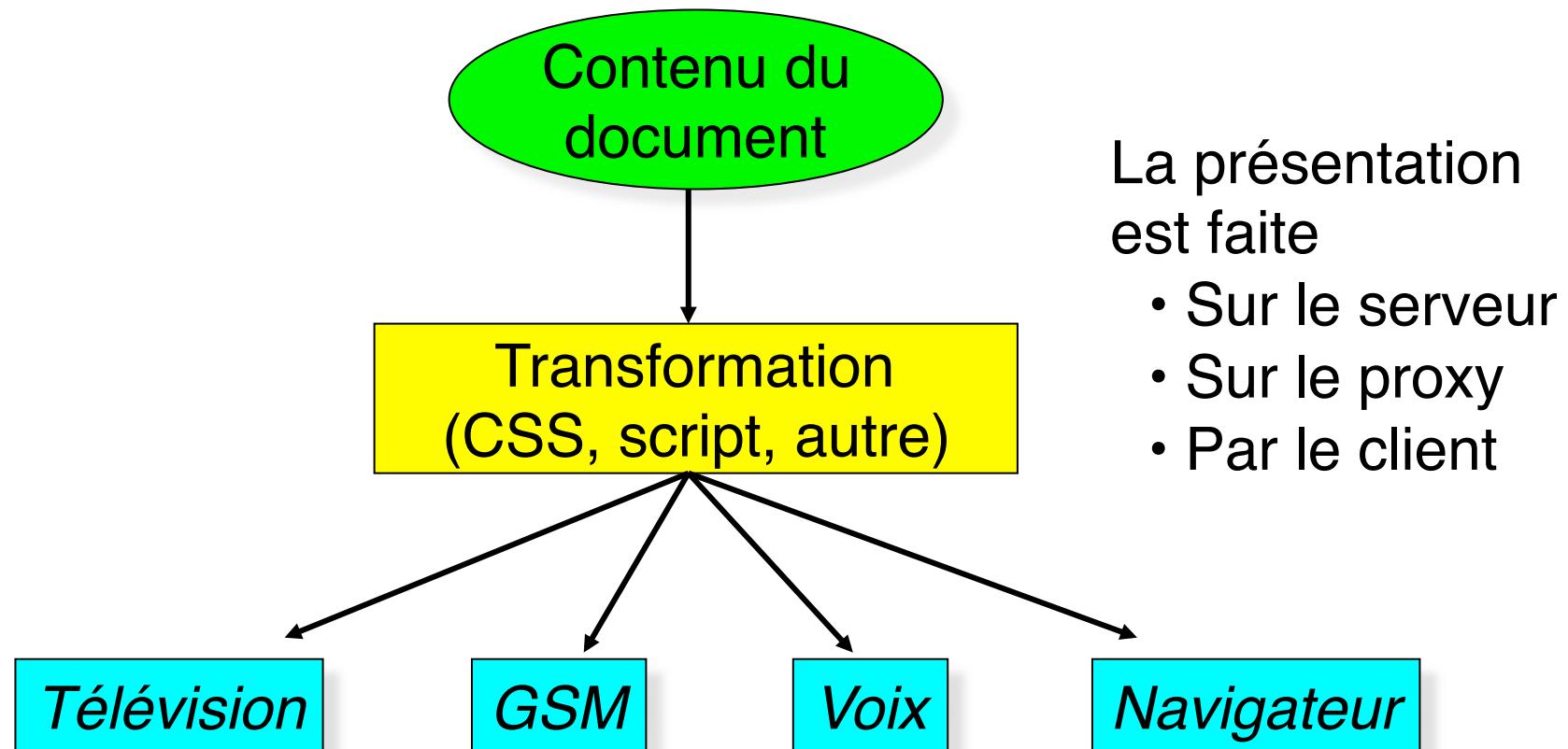
# Cours de Arnaud Le Hors

Excellent cours de Arnaud Le Hors qui a travaillé pour le w3c.

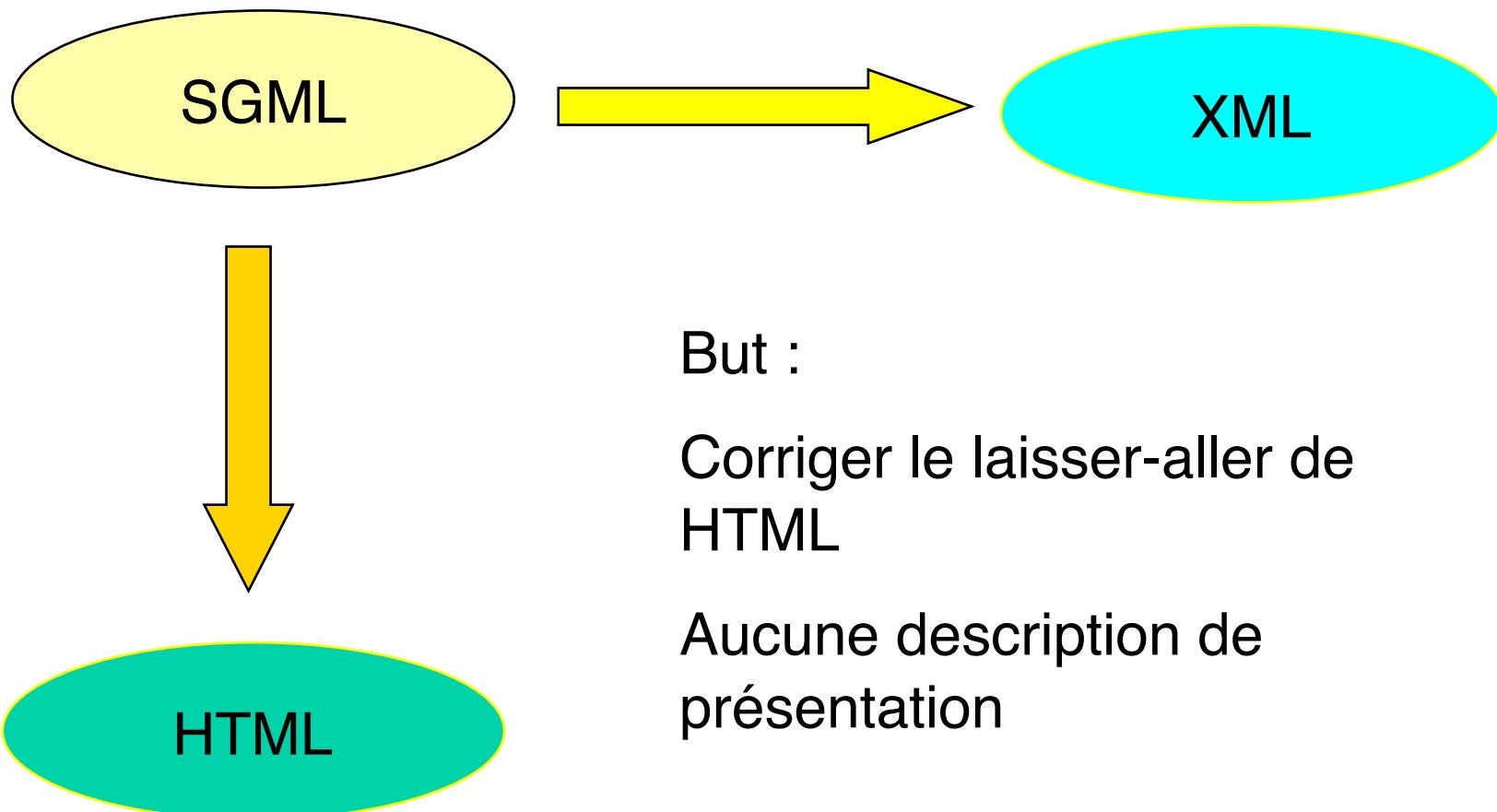
[http://users.polytech.unice.fr/~buffa/cours/internet/  
POLYS/makeslide/HTMLnCSS/slide27.htm](http://users.polytech.unice.fr/~buffa/cours/internet/POLYS/makeslide/HTMLnCSS/slide27.htm)

# **Conclusion**

# 1 x contenu - n x présentations



# XML



## Exemple XML

```
<CARNET>
    <ADRESSE NOM= " DURAND " >
        <DOMICILE>
            <RUE>57 rue du dessous des berges</RUE>
            <VILLE>Paris</VILLE>
            <CODEPOSTAL>75013</CODEPOSTAL>
            <PAYS>France</PAYS>
            <EMAIL>durand@hotmail.com</EMAIL>
        </DOMICILE>
        <BUREAU>
            <RUE>14 rue Voltaire</RUE>
            <VILLE>Le Kremlin Bicêtre</VILLE>
            <CODEPOSTAL>94270</CODEPOSTAL>
            <PAYS>France</PAYS>
        </BUREAU>
    </ADRESSE>
</CARNET>
```

## La DTD de l'exemple

```
<!ELEMENT CARNET (ADRESSE)+>
<!ELEMENT ADRESSE (BUREAU I DOMICILE)+>
<!ATTLIST ADRESSE NOM CDATA #REQUIRED>
<!ELEMENT BUREAU (RUE, VILLE, CODEPOSTAL,
    PAYS,EMAIL, PCDATA ?)>
<!ELEMENT DOMICILE (RUE, VILLE, CODEPOSTAL,
    PAYS, EMAIL, PCDATA ?)>
<!ELEMENT RUE (PCDATA ?)>
<!ELEMENT VILLE (PCDATA ?)>
<!ELEMENT CODEPOSTAL (PCDATA ?)>
<!ELEMENT PAYS (PCDATA ?)>
<!ELEMENT EMAIL (PCDATA ?)>
```

# Conclusion

**HTML est un standard "de facto" indépendamment de la volonté de tout organisme de standardisation**

**HTML basique : mélange contenu / présentation**

**CSS : Cascading Style Sheets**

**HTML + auto-discipline + CSS**

- **Bonne séparation**
- **Pouvoir d'expression convenable**

# Références

W3C

<http://www.w3.org/standards/webdesign/htmlcss>

**HTML 4.01 Specification**

<http://www.w3.org/TR/html401/>

**HTML 5**

<http://www.w3.org/TR/html5/>

**Cascading Style Sheets, level 1**

<http://www.w3.org/TR/1999/REC-CSS1-19990111>

**Cascading Style Sheets Level 2 Revision 1 Specification**

<http://www.w3.org/TR/CSS/>