

Les langages dédiés au Web (X)HTML – CSS



M1 PANIST – 2012-2013
LIONEL MÉDINI
COURS 4-5

Université Claude Bernard Lyon 1
UFR FST – département Informatique

D'après le cours de Yannick Prié

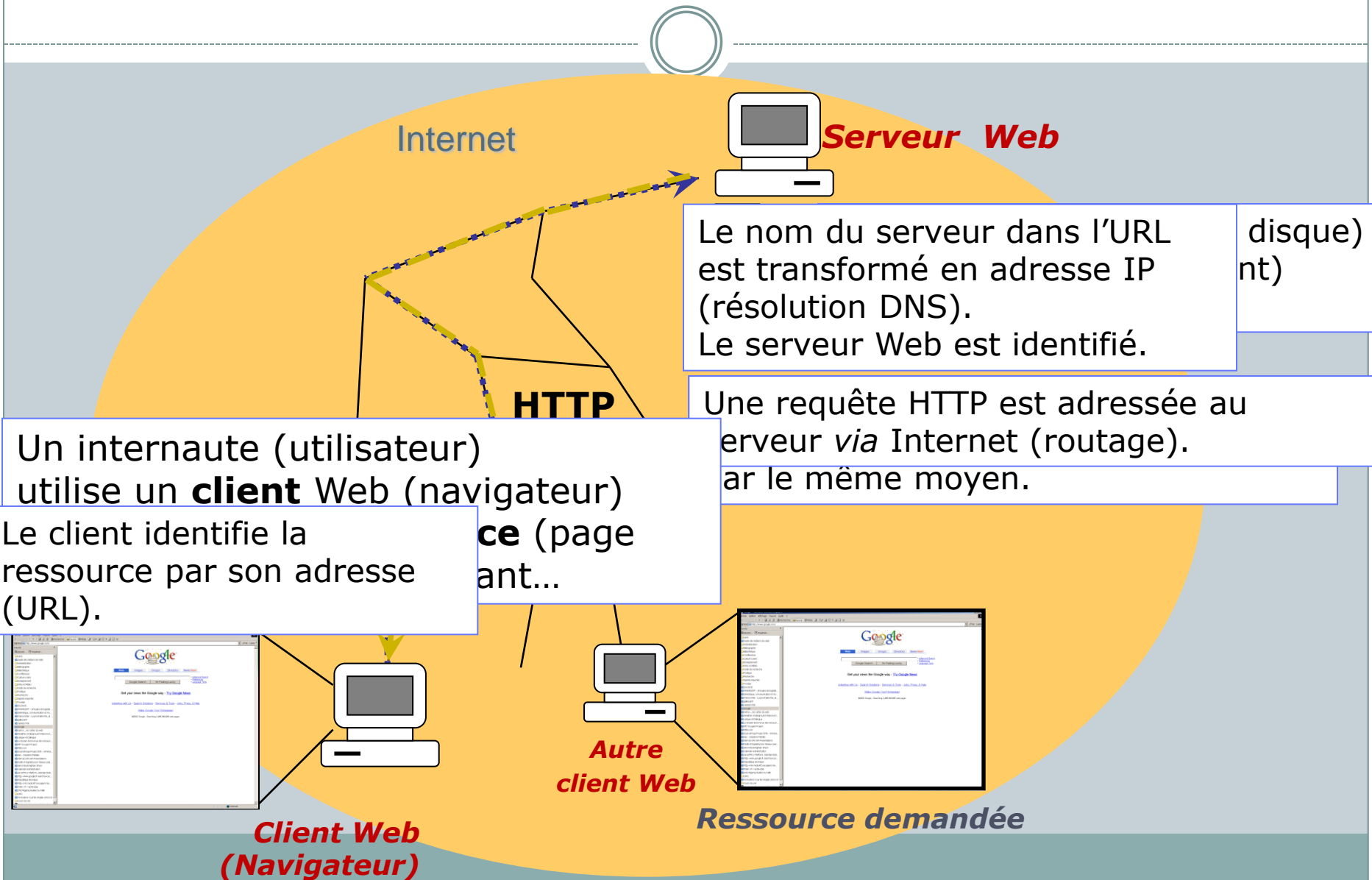
Objectifs du cours

2

- Présenter des documents XML dans un navigateur web
 - Historique rapide des langages dédiés au Web
 - Description de pages Web : HTML et XHTML
 - Mise en forme : CSS

Introduction :

Fonctionnement du Web en un schéma



Internet



Serveur Web

Le nom du serveur dans l'URL est transformé en adresse IP (résolution DNS).
Le serveur Web est identifié.

(disque)
(nt)

Une requête HTTP est adressée au serveur *via* Internet (routage).
Par le même moyen.

Un internaute (utilisateur) utilise un **client** Web (navigateur).
Le client identifie la ressource par son adresse (URL).
Le client identifie la ressource par son adresse (URL).

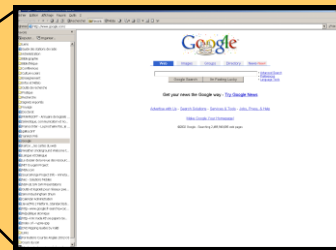
ressource (page)
demandée...



Client Web (Navigateur)



Autre client Web



Ressource demandée

Les langages dédiés au Web



- **Historique 1/3**
 - 1960-1986 : SGML (norme ISO)
 - 1989 : ODA (norme ISO, concurrent de SGML)
 - Fin des années 80 : apparition/essor du web
 - 1992-1997 : HTML (versions 1.0 -> 4.01)
 - Octobre 1994 : création du World Wide Web Consortium (W3C) : <http://www.w3.org>
 - 1996-1999 : CSS Level 1 (fonctionnalités de base)
 - Février 1998 : XML version 1.0
 - 1998 : CSS Level 2 (fonctionnalités supplémentaires)
 - Octobre 1998 DOM Level 1 (supporte XML et HTML)

Les langages dédiés au Web



- **Historique 2/3**
 - Décembre 1999 : XHTML 1.0
 - 1999-2004 : RDF et RDF-Schema 1.0
 - Novembre 2000 : DOM Level 2 (supporte CSS et espaces de noms XML)
 - Mai 2001 : schémas XML 1.0
 - Juin 2001 : XLink 1.0
 - Juillet 2001 : SVG 1.0
 - Novembre 2001 : XSL 1.0
 - Janvier 2003 : SVG 1.1

Les langages dédiés au Web



- **Historique 3/3**

- Février 2004 et août 2006 : XML 1.1
- Février 2004 : OWL 1.0
- Avril 2004 : DOM Level 3 Core
- Octobre 2004 : XML Schema (2^e édition)
- Octobre 2004 : XQuery 1.0
- Octobre 2004 : XPath 2.0 (Working Draft)
- Novembre 2004 : XSLT 2.0 (Working Draft)
- Décembre 2004 : WSDL 2.0
- ...
- Restent en développement : CSS L3, DOM L3...
- 17 décembre 2012 : HTML5 → Candidate Recommendation

HTML

7

- DTD de SGML
- Ensemble d'éléments et d'attributs permettant de définir des documents hypertextes
 - structure, liens, images, tableaux, formulaires...
- Quatre versions
 - HTML 1.0 → HTML 4.01
- Problèmes principaux
 - Mélange structure physique et logique
 - ✦ Ex. : un élément pour mettre en gras
 - Syntaxe peu stricte
 - ✦ Ex : pas forcément obligatoire de fermer un élément
- Tout cela a une origine historique (guerre des navigateurs), et a paradoxalement contribué au développement massif du web (simplicité, tolérance)

CSS

8

- Séparation de la structure logique et de la présentation documents HTML
 - Structure logique = HTML = ensemble d'éléments de contenu
 - Présentation suivant une feuille de style (*style sheet*) qui traite les éléments de contenu en éléments de présentation
- Cascading Style Sheets
 - Feuilles de styles associées à HTML (à XML)
 - En cascade
 - ✦ on peut utiliser des feuilles de styles multiples
 - ✦ il y a un degré d'importance pour chaque feuille de style

XHTML

9

- **XHTML 1.0 (1999)**
 - reformulation XML de HTML4
 - définit trois DTD pour exprimer des documents HTML
 - ✦ XHTML-1.0-Strict
 - ✦ XHTML-1.0-Transitional
 - ✦ XHTML-1.0-Frameset
 - sémantique des balises
 - ✦ Définie dans HTML4
- **XHTML 1.1**
 - Modularisation
 - ✦ faciliter le mixage de fragments XML dans XHTML
- **XHTML 2**
 - Modulaire
 - Nouvelles fonctionnalités
 - En cours de discussion
- **HTML, le retour...**
 - HTML 5 : W3C Candidate Recommendation (17 décembre 2012)

HTML → XHTML : à savoir

10

- De la rigueur...
 - Nom d'éléments en minuscules
 - ✦ `<P>` → `<p>`
 - Fermetures correctes : toute balise ouverte doit être fermée
 - ✦ `<p> ... </p>`
 - Éléments vides fermés :
 - ✦ `<hr />`
 - Un attribut associé à une valeur, entre guillemets
 - ✦ `` → ``
 - Attribut id pour identifier un élément (vs name avant)
 - ✦ `<h1 id="partie1" name="partie1"> ... </h1>`

Plan

11

- **XHTML**
 - Structure générale d'un document XHTML
 - XHTML Strict et Transitionnel
 - Éléments les plus utiles
 - HTML5
- **CSS**
 - Généralités
 - Règles et sélecteurs
 - Propriétés utiles
 - Placement des styles
 - Héritage et cascade

Structure générale d'un document XHTML

12

- Prologue
 - Déclaration XML
 - Déclaration de DTD
- Racine
 - Élément `html`
- En-tête
 - Élément `head`
- Corps
 - Élément `body`
- Commentaires n'importe où
 - `<!-- ... -->`

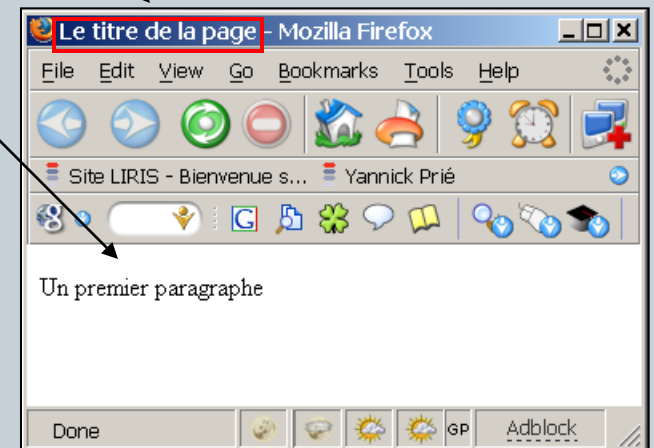
Premier exemple

13

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
  <head>
    <title>Le titre de la page</title>
  </head>
  <body>
    <!-- un commentaire -->
    <p>Un premier
    paragraphe</p>
  </body>
</html>
```

Remarque :

ici le navigateur utilise une feuille de style par défaut pour afficher du XHTML strict non lié à une feuille de style CSS.



XHTML Strict et Transitionnel

14

- **Transitionnel**

- Les éléments de présentation de HTML sont encore autorisés
 - ✦ b, center, font, ...

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

- **Strict**

- Séparation stricte du document XHTML de sa présentation → utilisation de style CSS obligatoire

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

- **Déclaration d'espace de nom dans la balise ouvrante html**

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
```

- Pour des débutants : **XHTML strict + CSS**

En-tête : contenu de l'élément **head**

15

- Titre du document
 - `<title></title>`
- Autre informations non affichées à l'écran, utilisées par le navigateur, les moteurs, etc.
 - `<meta name="..." content="..." />`
 - ✦ `<meta http-equiv="Refresh" content="4" ; URL=http://www.google.com" />`
 - ✦ `<meta name="author" content="" />`
 - ✦ `<meta name="Keywords" content="motcle1, motcle2, motcle3"/>`
 - ✦ `<meta name="language" content="fr"/>`
 - `<base href="URL-de-base" />`
 - ✦ URL de base pour les URL relatives
- Styles
 - `<style />` → inclure une feuille de style CSS dans la page
 - `<link />` → lier le document à une ressource externe (typiquement, feuille de style)
- Scripts
 - `<script />` → ajouter un script à la page

Corps

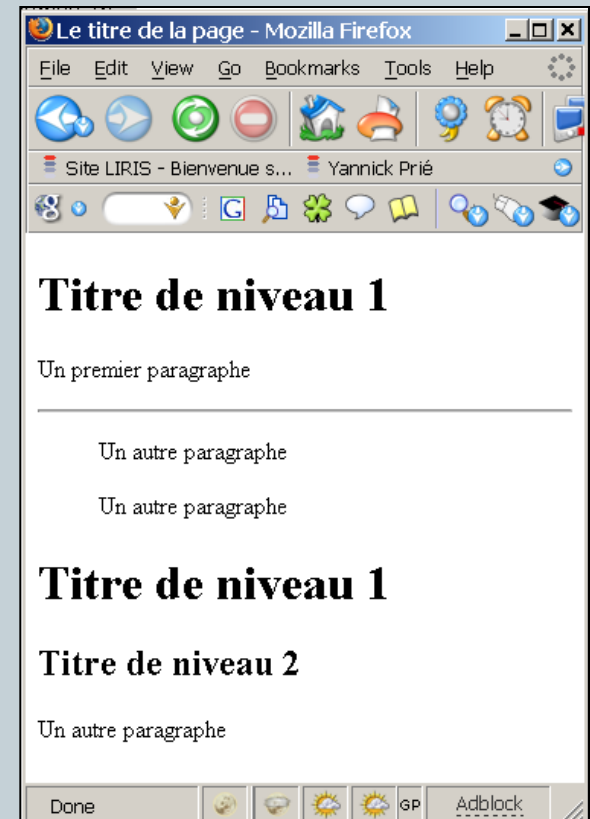
16

- **Elément body**
 - Toutes les informations visualisables
- **Structuration**
 - `<p> ... <p>` → paragraphe
 - `<h1> ... <h1>` → titre de niveau 1
 - ...
 - `<h6> ... <h6>` → titre de niveau 6
 - `<hr />` → ligne horizontale
 - `
` → saut de ligne

Corps : exemple

17

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
  <title>Le titre de la page</title>
</head>
<body>
  <h1>Titre de niveau 1</h1>
  <p>Un premier paragraphe</p>
  <hr /> <!-- Une ligne horizontale -->
  <blockquote><p>Un autre paragraphe</p>
  <p>Un autre paragraphe</p></blockquote>
  <h1>Titre de niveau 1</h1>
  <h2>Titre de niveau 2</h2>
  <p>Un autre paragraphe</p>
</body>
</html>
```



Mise en évidence

18

- **Emphase**
 - ` ... ` (emphasis)
- **Emphase forte**
 - ` ... `
- **Exposants et indices**
 - `^{...}` (superscript)
 - `_{...}` (subscripted)

Extraits, citations et références

19

- Citer quelque chose : **cite**
 - ... `<cite>Discours de la méthode</cite> de <cite>Descartes</cite>` ...
- Citation courte : **q**
 - Comme le disait `<cite>Ronsard</cite>` :
`<q>Mignonne allons voir si la rose...</q>`
- Citation longue : **blockquote**
 - `<h5><cite>Barbara</cite> de <cite>Jacques Prévert</cite></h5>`
`<blockquote cite="http://wwwuser.gwdg.de/">`
`<p>Rappelle-toi Barbara</p>`
...
`<p>Dont il ne reste rien.</p>`
`</blockquote>`

Autres balises de structuration

20

- Texte spécial
 - `<pre> ... </pre>` (preformatted text)
 - `<code> ... </code>`
 - ✦ pour mettre du texte représente du code informatique (par exemple du XML !)
- Indications sur le contenu textuel
 - Utilisation dans une feuille de style pour mettre en forme
 - ✦ **defn** (définition)
 - ✦ **acronym** (acronyme)
 - ✦ **abbr** (abréviation)
 - Utilisation par des logiciels pour connaître la sémantique du contenu textuel (utilisé ?)
 - Principe du web sémantique : généralisation, extériorisation des vocabulaires, *etc.*

Images

21

- Principe

- Un élément image (img) est lié à un fichier image (attribut src)
- Une image est une ressource externe, désignée par une URL, qui doit être chargée par le navigateur

- Exemple

- ```

```

- Images cliquables

- ```

<map id="mymap">
  <area      href="section1.html" alt="Route 20"
            shape="rect" coords="0,0,49,49" />
  <area      href="section2.html" alt="Route 35"
            shape="rect" coords="0,49,49,99" />
</map>
```

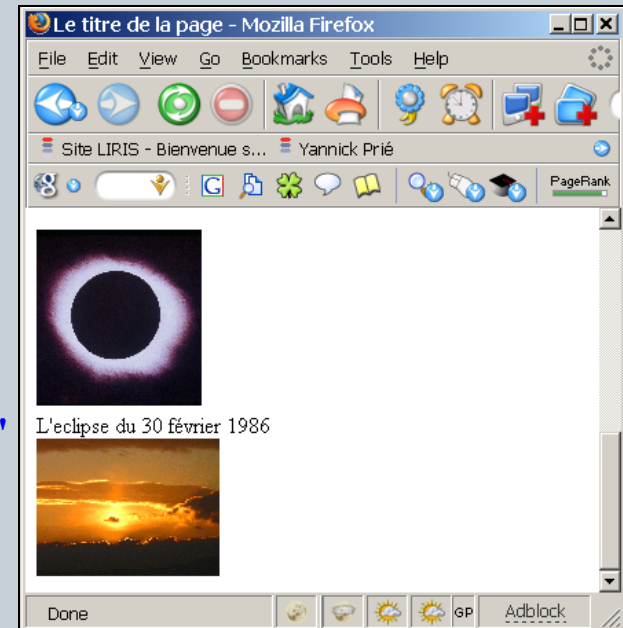
Images : exemple

22

```
...  
  
<br />  
 <!-- erreur ! -->  
<br />  
  
...
```

● Remarque importante

- il faut trois ressources pour afficher cette page
 - ✦ Fichier source HTML
 - ✦ Fichier `eclipse.jpg` (chemin local : dans le même dossier que le fichier source HTML)
 - ✦ Fichier `soleil.JPG` (disponible sur la machine `web.ccr.jussieu.fr`, en suivant le chemin `/cim2/SAXaussois/soleil.JPG`)
- trois requêtes HTTP en tout pour l'affichage



Listes

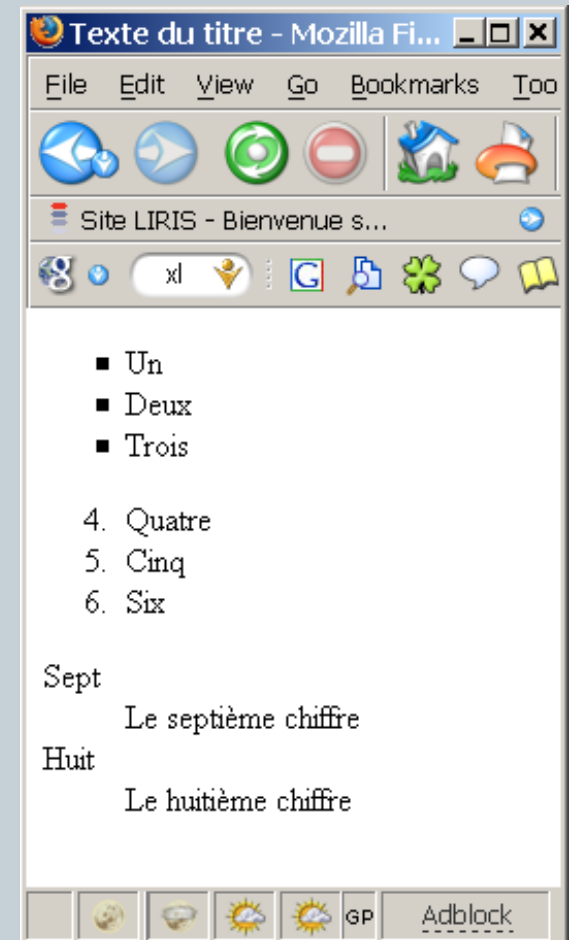
23

- Principe général
 - Un élément liste contient des élément items
- Listes classiques
 - Liste à puces
 - ✦ ` ... ` (unordered list)
 - Liste ordonnée
 - ✦ ` ... ` (ordered list)
 - Item de list
 - ✦ ` ... `
- Liste de définitions
 - Conteneur
 - ✦ `<dl> ... </dl>`
 - Terme de définition
 - ✦ `<dt> ... </dt>`
 - Description de définition (= définition elle-même)
 - ✦ `<dd> ... </dd>`

Listes : exemple

24

```
...  
<ul type="square">  
  <li>Un</li>  
  <li>Deux</li>  
  <li>Trois</li>  
</ul>  
<ol start="4">  
  <li>Quatre</li>  
  <li>Cinq</li>  
  <li>Six</li>  
</ol>  
<dl>  
  <dt>Sept</dt>  
  <dd>Le septième chiffre</dd>  
  <dt>Huit</dt>  
  <dd>Le huitième chiffre</dd>  
</dl>  
...
```



Liens hypertextes

25

- **Principe**
 - une ancre contenant le texte ou les éléments sur lesquels on peut cliquer
 - le clic redirige vers une URL
 - ✦ entraîne le chargement de la ressource désignée par l'URL
- **Syntaxe**
 - ``
ancre
``
 - **href** : URL de destination du lien
 - **title** : texte qui apparaîtra dans une info-bulle si on survole le lien

Liens hypertextes : exemples

```
<p>Quelques exemples de liens</p>
```

```
<p><a href="URL" title="serveur  
web de l'UCBL">Cliquez ici</a>  
pour atteindre le site de  
l'Université Claude Bernard  
Lyon 1.</p>
```

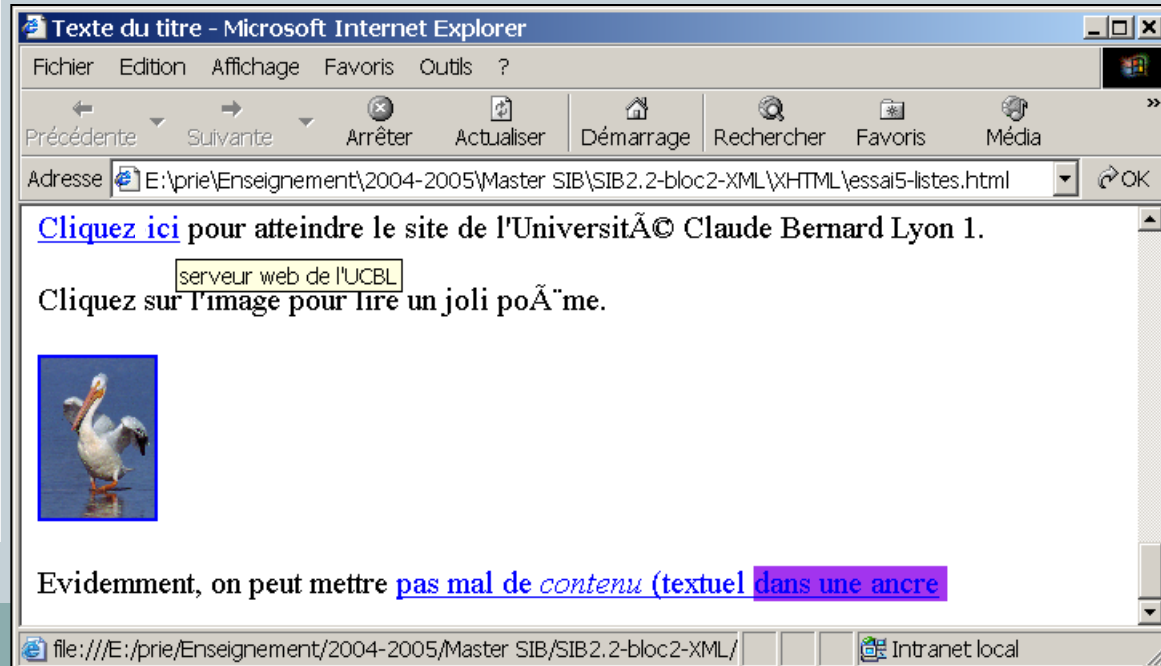
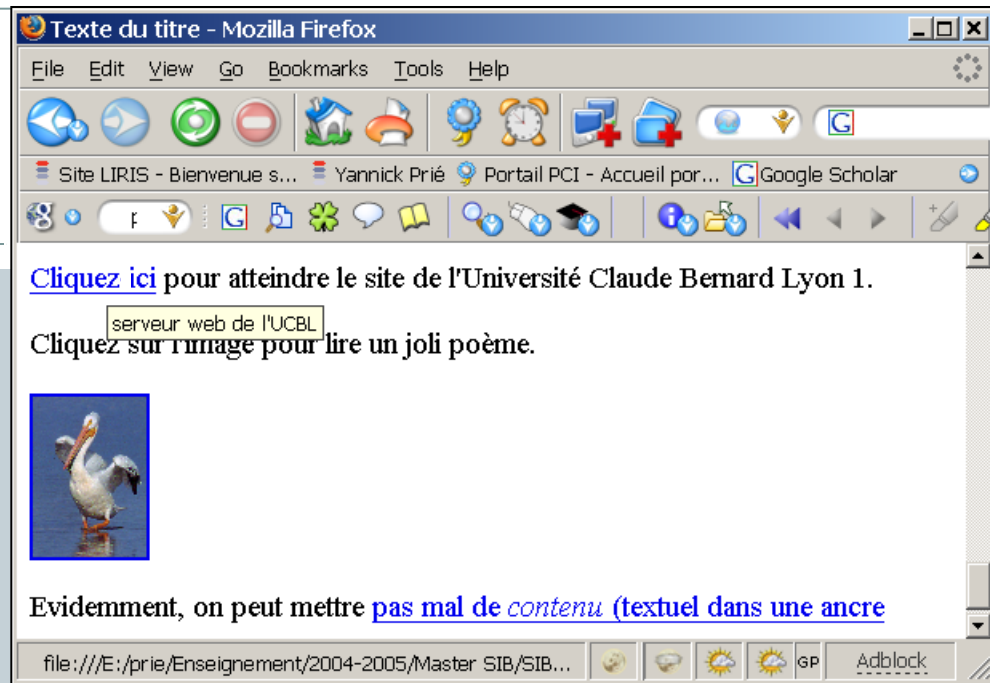
```
<!-- Remarque : on a ici un lien  
absolu (URL complète) -->
```

```
<p>Cliquez sur l'image pour lire  
un joli poème.</p>
```

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

```
<!-- Remarque : on a ici un lien  
relatif -->
```

```
<p>Evidemment, on peut mettre <a  
href="http://www.w3.org/TR/xht  
ml1/">pas mal de  
<em>contenu</em> (textuel  
<span style="background-color:  
#A234EE" éléments)</span> dans  
une ancre</a> </p>
```



Tableaux

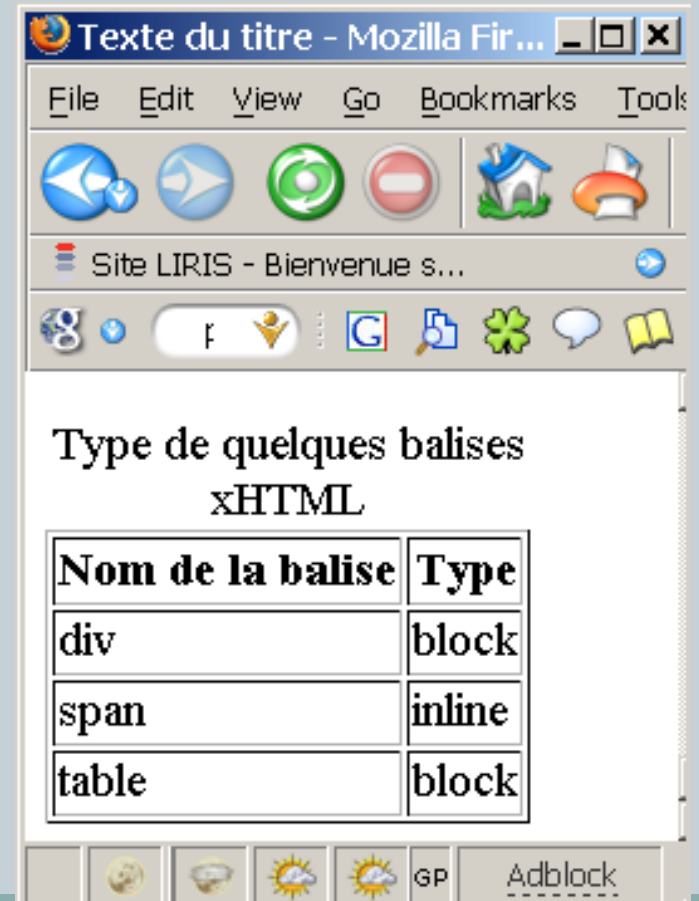
27

- Principe
 - Un tableau contient des lignes, lesquelles contiennent des cellules
- Éléments
 - `<table> ... </table>` (élément général)
 - `<tr> ... </tr>` (table row)
 - `<td> ... </td>` (table cell)
 - `<th> ... </th>` (table header)
 - `<caption> ... </caption>` (table caption)
 - ...

Tableaux : exemple

28

```
<table summary="Ce tableau présente des balises xHTML et leur type." border="1">
<caption>Type de quelques balises xHTML</caption>
<thead>
<tr>
  <th scope="col">Nom de la balise</th>
  <th scope="col">Type</th>
</tr>
</thead>
<tbody>
<tr>
  <td>div</td>
  <td>block</td>
</tr>
<tr>
  <td>span</td>
  <td>inline</td>
</tr>
<tr>
  <td>table</td>
  <td>block</td>
</tr>
</tbody>
</table>
```



Deux types d'éléments en HTML

29

- **Eléments *block***
 - Définissent des blocs physiques (des « boîtes ») dans le document
 - Induisent un saut de ligne à la fin de leurs contenus
 - Exemples
 - ✦ p, h1-h6, div, ul, ol, li, pre, hr, table, etc.
- **Eléments *inline***
 - Dans le flux des caractères
 - « Semblables » à des caractères
 - Exemple
 - ✦ em, img, strong, a, etc.

Les éléments span et div

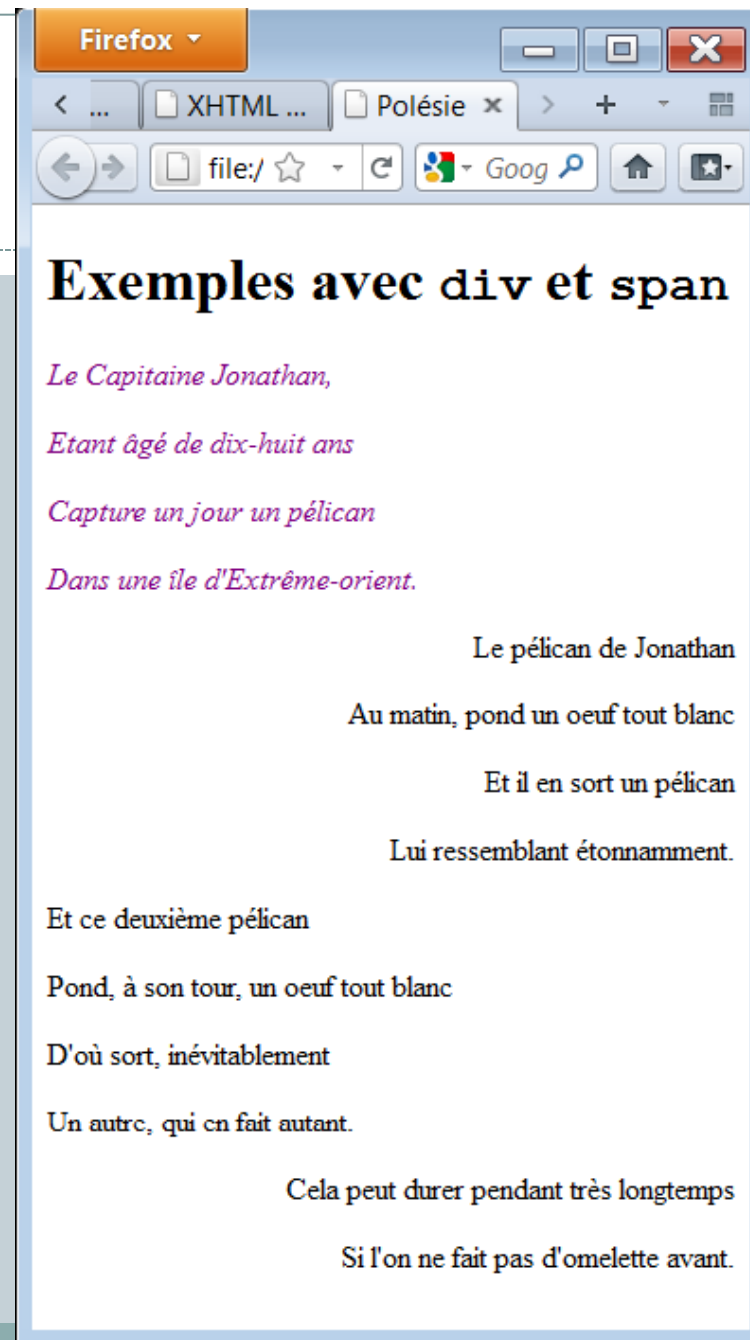
30

- **Utilité**
 - Regrouper des sous-parties de document
 - ✦ Pour leur donner un nom
 - ✦ Pour enclorre (contenir) du texte ou d'autres éléments XHTML
 - ✦ Pour les traiter globalement en leur affectant des styles, des événements...
 - Mais sans leur donner de sémantique particulière
- **` ... `**
 - Élément de type *inline*
 - Contient du texte ou d'autres éléments *inline*
- **`<div> ... </div>`**
 - Élément de type *block*
 - Contient du texte, d'autres éléments *inline* ou *block*
 - Retour chariot à la fin

div / span : exemple (Desnos)

31

```
<h1>Exemples avec <code>div</code> et  
<code>span</code></h1>  
<div id="premiere_strophe" style="color: purple ;  
font-style: italic">  
<p>Le Capitaine <span class="heros">  
Jonathan</span>,</p>  
<p>Etant âgé de dix-huit ans</p>  
<p>Capture un jour un <span style="font-weight:bold">  
pélican</span></p>  
<p>Dans une île d'Extrême-orient.</p>  
</div><div style="text-align: right">  
<p>Le pélican de <span class="heros">  
Jonathan</span></p>  
<p>Au matin, pond un oeuf tout blanc</p>  
<p>Et il en sort un <span style="font-weight:bold">  
pélican</span></p>  
<p>Lui ressemblant étonnamment.</p>  
</div><div style="text-align: justified">  
<p>Et ce deuxième <span style="font-weight:bold">  
pélican</span></p>  
<p>Pond, à son tour, un oeuf tout blanc</p>  
<p>D'où sort, inévitablement</p>  
<p>Un autre, qui en fait autant.</p>  
</div><div style="text-align: right">  
<p>Cela peut durer <span style="text-align:  
right">pendant</span> très longtemps</p>  
<p>Si l'on ne fait pas d'omelette avant.</p></div>
```



Firefox ▾

< ... XHTML ... Polésie x > + ▾ ☰

← → file:/ ☆ ▾ ↻ 🌐 Goog 🔍 🏠 ☆

Exemples avec div et span

Le Capitaine Jonathan,

Etant âgé de dix-huit ans

Capture un jour un pélican

Dans une île d'Extrême-orient.

Le pélican de Jonathan

Au matin, pond un oeuf tout blanc

Et il en sort un pélican

Lui ressemblant étonnamment.

Et ce deuxième pélican

Pond, à son tour, un oeuf tout blanc

D'où sort, inévitablement

Un autre, qui en fait autant.

Cela peut durer très longtemps

Si l'on ne fait pas d'omelette avant.

Formulaires

32

- Objectif
 - Permettre à l'utilisateur d'envoyer des informations à un serveur
 - Saisir les informations
 - ✦ listes déroulantes, cases à cocher, zones de texte, etc.
 - ✦ bouton pour remettre à zéro
 - Envoyer les informations
 - ✦ méthode GET
 - les informations passent par l'URL
 - Ex. : `http://www.google.com/search?sourceid=mozclient&ie=utf-8&oe=utf-8&q=html+4.01`
 - ✦ Méthode POST
 - les informations sont échangées par le protocole HTTP
- Eléments
 - `<form> ... </form>` → définit un formulaire
 - `<input> ... </input>` → entrée de formulaire
 - `<textarea> ...</textarea>` → zone de texte
 - `<select> ... </select>` → liste déroulante
 - `<option> ... </option>` → option du select

Formulaires : exemple

33

```
<form method="post" action="http://serveur.com/script.php">
```

```
<p><input type="text" name="Champ_saisie" value="Texte" /></p>
```

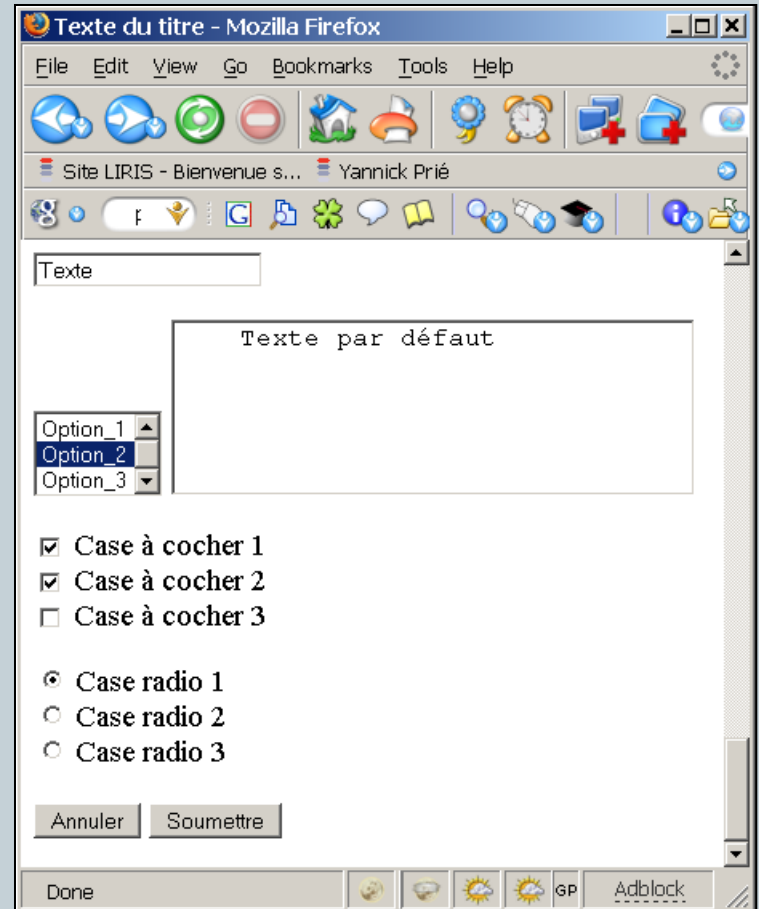
```
<p><select name="Liste_Choix" size="3">
  <option value="Option_1">Option_1</option>
  <option value="Option_2">Option_2</option>
  <option value="Option_3">Option_3</option>
</select>
```

```
<textarea name="Zone_Texte" cols="30" rows="5">
  Texte par défaut </textarea></p>
```

```
<p><input type="checkbox" name="case1" value="Case_1">
  Case à cocher 1<br />
<input type="checkbox" name="case2" value="Case_2">
  Case à cocher 2<br />
<input type="checkbox" name="case3" value="Case_3">
  Case à cocher 3<br /></p>
```

```
<p><input type="radio" name="Case_Radio"
  value="Case radio 1">Case radio 1<br />
<input type="radio" name="Case_Radio"
  value="Case radio 2">Case radio 2<br />
  <input type="radio" name="Case_Radio"
  value="Case radio 3">Case radio 3<br /></p>
```

```
<input type="reset" name="Annulation" value="Annuler">
<input type="submit" name="Soumission" value="Soumettre">
</form>
```



Frames

34

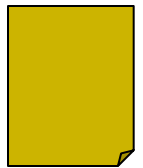
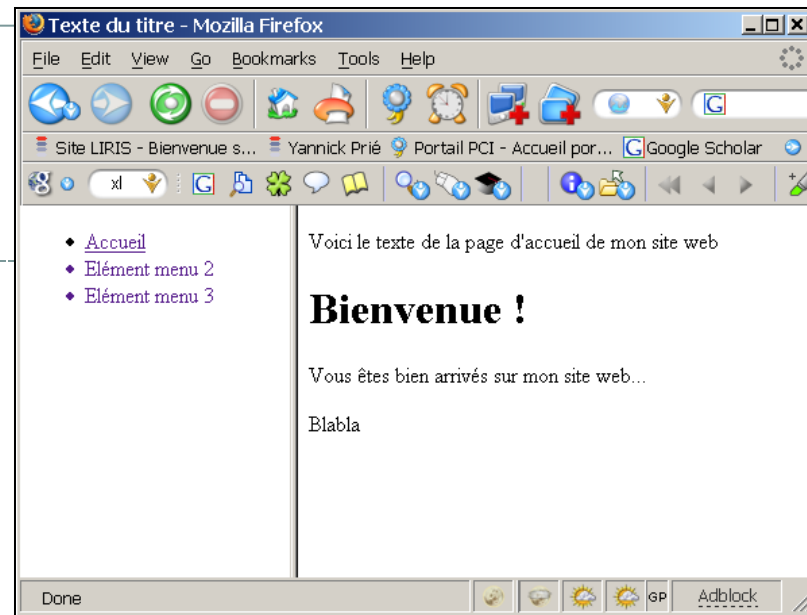
- **Principe**
 - diviser la fenêtre d'affichage en frames
 - afficher dans chaque frame une page HTML
 - Nombre de fichiers concernés
 - ✦ 1 pour la déclaration du frameset + 1 pour chaque frame
 - ✦ Ex : 3 frames → minimum 4 fichiers
- **En XHTML**
 - Le fichier principal obéit à la DTD frameset
 - Les autres sont en XHTML
- **Remarque**
 - Ne pas utiliser car pose beaucoup de problèmes
 - ✦ gestion
 - ✦ indexation

Frames : exemple

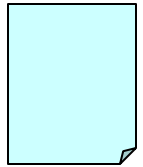
35

Le fichier principal fait appel à menu.html et accueil.html

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE HTML PUBLIC
  "-//W3C//DTD HTML 4.01 Frameset//EN"
  "http://www.w3.org/TR/html4/frameset.dtd">
<html>
<head>
<title>Texte du titre</title>
</head>
<frameset cols="200,*">
  <frame src="menu.html" name="cadre1" scrolling="no">
  <frame src="accueil.html" name="cadre2" scrolling="yes">
</frameset>
  <p>Ce projet utilise des cadres. Chez vous les cadres ne
  sont pas affichés.</p>
</frameset>
</html>
```



menu.html



accueil.html

Conclusion sur XHTML

36

- Possibilité de valider ses documents
 - <http://validator.w3.org/>
- Ce cours n'est pas une description complète des éléments et attributs de XHTML
 - Aller voir la référence
- Non abordés ici
 - Les scripts
 - ✦ Javascript
 - ✦ `<script> ... </script>`
 - Les objets pris en charge par des plug-ins
 - ✦ Applets JAVA, animations Flash, audio, vidéo, etc.
 - ✦ `<object> ... </object>`
 - Les sites web et leur gestion (conception, installation sur serveurs web...)

HTML5



- **Historique**

- A l'origine (fin 2003) : initiative d'Opera pour harmoniser Xforms et HTML dans son navigateur
- Par extension : un langage qui prendra en charge les différentes fonctionnalités des « Web applications »
- Proposition de création d'un GdT du W3C rejetée (début 2004)
- Création en 2006 du Web Hypertext Application Technology WG ([WHATWG](#)) sous la pression d'Apple, Mozilla et Opera
- Intégration au W3C : <http://www.w3.org/TR/html5/>

- **Principes de développement**

- Don't Reinvent The Wheel
- Pave The Cowpaths (!) : (X)HTML, CSS, DOM, EcmaScript...

HTML5



- **Détails des spécifications**
 - Un langage de représentation abstrait
 - ✦ Description de documents et d'« applications Web »
 - ✦ API d'interaction avec les représentations de ces ressources (DOM)
 - Deux syntaxes concrètes
 - ✦ HTML5 : pour les contenus de type HTML, destinés à être traités par un navigateur
 - ✦ XHTML5 : pour les contenus de type XML, destinés à être parsés puis traités par une application particulière
- **État actuel**
 - Date initiale de sortie de la spec. : avant octobre 2010...
 - Dernière version : Candidate Recommendation du 17 déc. 2012

Plan

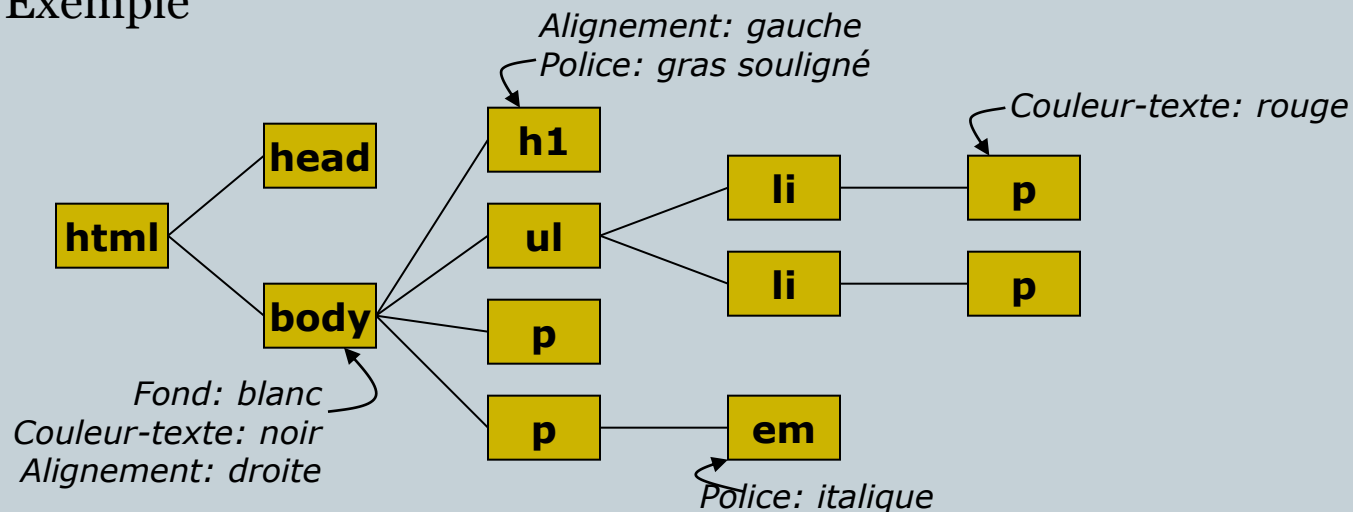
39

- **XHTML**
 - Structure générale d'un document XHTML
 - XHTML Strict et Transitionnel
 - Éléments les plus utiles
 - HTML5
- **CSS**
 - Généralités
 - Règles et sélecteurs
 - Propriétés utiles
 - Placement des styles
 - Héritage et cascade

CSS : généralités

40

- Objectif
 - Décrire comment un document XHTML doit être affiché
 - ✦ Remplace les éléments d'affichage des anciennes versions de HTML (séparation réelle du contenu et de sa présentation)
 - ✦ L'affichage est pris en charge par le navigateur (normalement)
- Principe
 - Décoration de l'arbre des éléments XHTML
 - ✦ Associer un certain nombre d'attributs de style à un élément
- Exemple



Feuille de style CSS

41

- Une feuille de style est composée d'un certain nombre de règles (*rules*)
- Une règle se compose
 - d'un sélecteur
 - d'une déclaration
- Une déclaration se compose d'un ensemble de propriétés/valeurs
- Remarque
 - L'ordre des déclarations est indifférent
 - Tous les styles ne peuvent pas s'appliquer à tous les éléments
 - ✦ Ex. : une image n'a pas de style de police

CSS : structure des règles

42

sélecteur déclaration



```
h1 { color: purple; }
```

propriété

valeur

```
h1 { font-family: Arial, sans-serif; font-style: italic }
```

séparateur de propriétés/valeurs

Exemple de règle CSS

43

```
body {  
    background: #FFFFFF;  
    color: black; /* commentaire */  
    margin-left: 5%;  
    margin-right: 5%;  
    font-family: Tahoma, Optima,  
    Arial, sans-serif;  
}
```

Types de sélecteurs

44

- Simples et groupes
- Classes
- Pseudo-classes
- Pseudo-éléments
- Contextuels

Sélecteurs CSS : simple et groupe

45

- Simple
 - Lié à un type d'élément HTML
 - Utilisation de son nom
 - Exemple
 - ✦ `h1 { text-align: center; }`
- Groupe
 - Regroupement de règles qui s'appliquent à plusieurs éléments
 - Exemple
 - ✦ `h2, p { font-family: Optima, Arial, sans-serif; }`
- Impossibilité de considérer différemment des éléments de même type

Sélecteurs CSS : id

46

- On peut assigner un identifiant à un élément HTML

```
<div id="menu" >
```

- On peut lui assigner un style par un sélecteur particulier dans la feuille de style

```
#menu { position: fixed; }
```

Sélecteurs CSS : classes

47

- On peut assigner une classe à un élément HTML

```
<h1 class="header" >
```

- Celle-ci spécifie un sélecteur particulier dans le feuille de style

```
h1.header { text-align: center; }
```

- Une classe peut s'appliquer à de multiples éléments

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

s'appliquera aussi à `<h2 class="header">`, *etc.*

Sélecteurs CSS : pseudo-classes

48

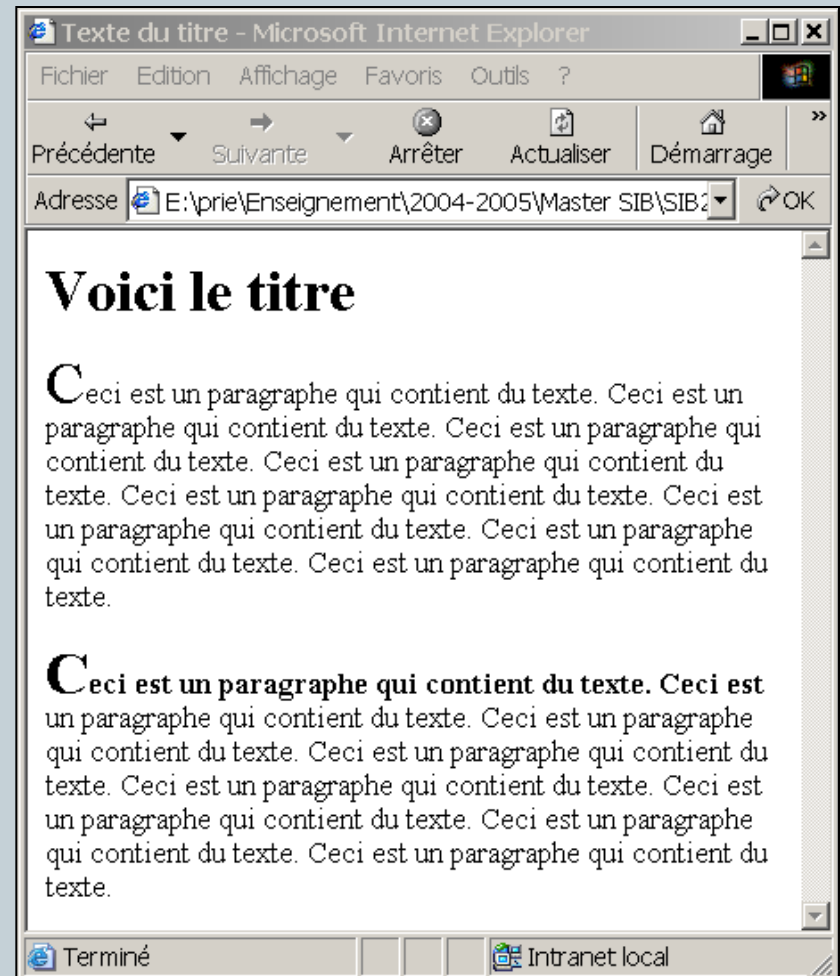
- Sélecteurs qui sélectionnent des éléments en fonction de leur état à un moment donné
- Exemple et intérêt principal
 - `a:link` — lien non visité et inactif
 - `a:hover` — lien sur lequel passe le pointeur de la souris
 - `a:active` — lien sur lequel on clique
 - `a:visited` — lien déjà visité
- Exemple

```
a:link {color: blue;}
a:visited {color: magenta;}
a:hover {color: red; text-decoration:none;
font-weight: bold;}
a:active {color: red;}
```

Sélecteurs CSS : pseudo-éléments

49

- **:first-letter**
 - première lettre dans un élément bloc (ex. p, h1, ...)
- **:first-line**
 - première ligne dans un élément bloc (ex. p, h1, ...)

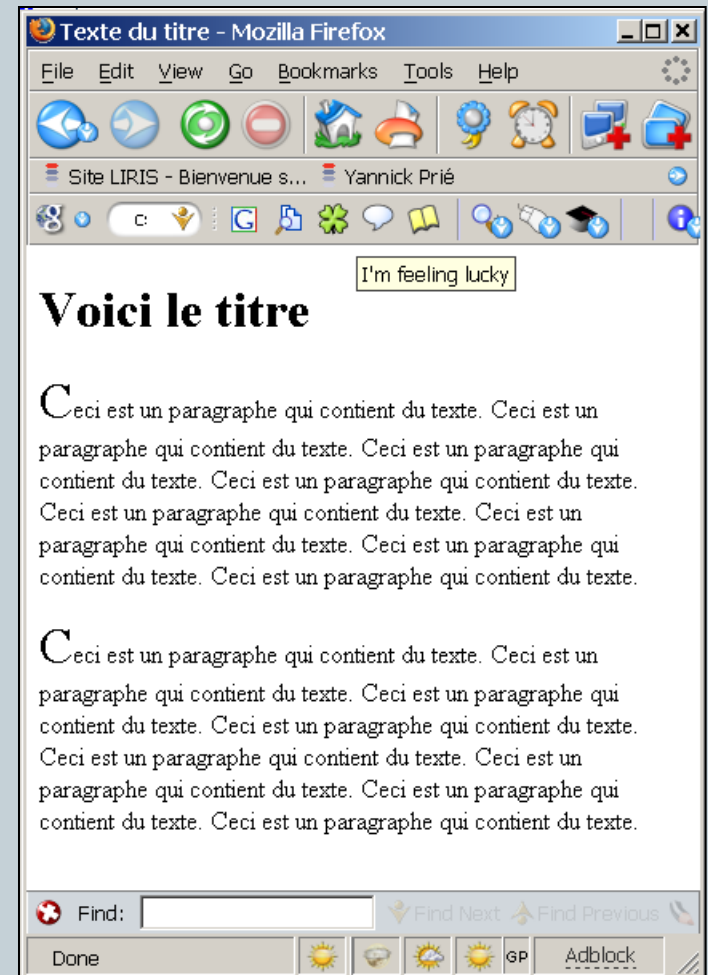


Pseudo-éléments : exemple

50

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
<html>
<head>
<title>Texte du titre</title>
<style type="text/css">

</style>
</head>
<body>
<h1>Voici le titre</h1>
<p>Ceci est <!-- ... --> du texte. </p>
      Ceci <!-- ... --> qui contient
du texte. Ceci est un paragraphe qui
contient du texte. </p>
</body>
</html>
```



Sélecteurs CSS : contextuels

51

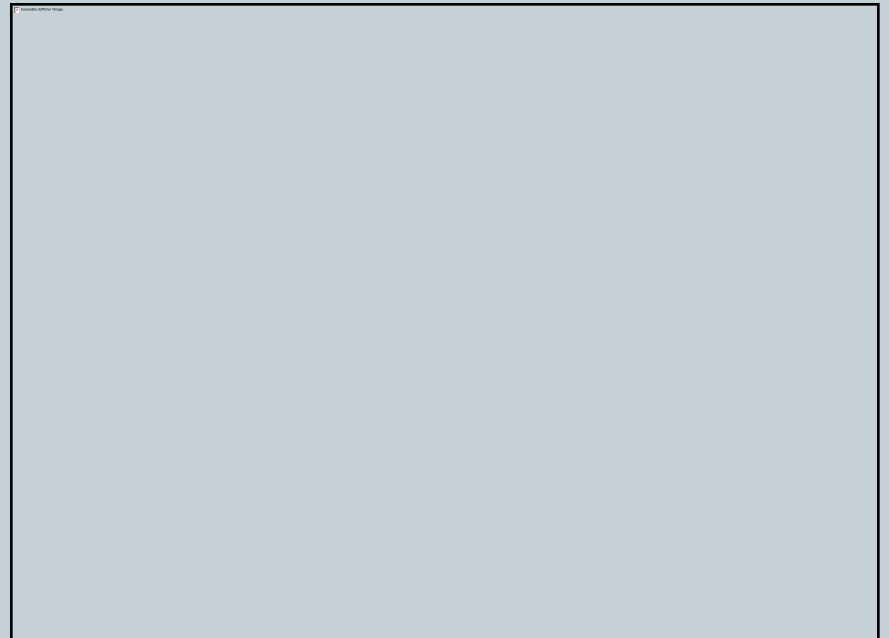
- Sélecteurs qui ne sélectionnent que des éléments dans un certain contexte

Style :

```
h1 em { color: red; }
```

XHTML :

```
<h1>Ceci est un texte  
de header <em>ce texte  
est mis en  
évidence</em> celui-ci  
ne l'est pas.</h1>  
<p>Dans ce paragraphe,  
<em>ceci est mis en  
évidence</em></p>
```



Quelques propriétés de textes

52

- **font-size:**
 - small | medium... | % | x pt
- **font-family:**
 - fontname1, fontname2 (*si la première n'est pas disponible*), familynome (*serif, sans-serif, etc.*)
- **font-weight:**
 - bold | lighter
- **font-style:**
 - italic, oblique
- **text-align:**
 - left | center | right | justify
- **text-indent:** (*retrait de première ligne*)
 - % | x cm

Les couleurs

53

- color:
 - red | blue...| hexcode
- background-color:
 - red | blue...| hexcode

■ Black = "#000000"

■ Silver = "#C0C0C0"

■ Gray = "#808080"

□ White = "#FFFFFF"

■ Maroon = "#800000"

■ Red = "#FF0000"

■ Purple = "#800080"

■ Fuchsia = "#FF00FF"

■ Green = "#008000"

■ Lime = "#00FF00"

■ Olive = "#808000"

■ Yellow = "#FFFF00"

■ Navy = "#000080"

■ Blue = "#0000FF"

■ Teal = "#008080"

■ Aqua = "#00FFFF"

Types d'éléments

54

- **Éléments blocs**
 - Prennent la forme d'un bloc dans la page
 - ✦ ensemble de lignes
 - ✦ ne peuvent être contenus que dans d'autres éléments blocs
 - Exemple
 - ✦ p, img, ul, table, h1, **div**...
 - Propriétés de bloc
- **Éléments inline**
 - S'inscrivent dans la continuité des éléments
 - ✦ ne forcent pas un changement de ligne
 - ✦ peuvent être inclus dans n'importe quel élément
 - Exemple
 - ✦ a, em, **span**...
- **Éléments de listes**
 - éléments HTML qui ont un marqueur (bullet, number) et un ordre

Rappels sur div et span

55

- **div**
 - élément contenant d'autres éléments, servant à définir un bloc
 - servira essentiellement à faire du positionnement de bloc
- **span**
 - élément contenant d'autres éléments, inline
 - servira essentiellement à regrouper des caractéristiques globales pour des éléments textuels

Propriétés de styles graphiques

56

- S'appliquent aux éléments blocs
- Attribut **position**
 - **relative**
 - ✦ bloc positionné par rapport à sa position normale, dans le flux (ex. décalage de 20 pts à droite)
 - **absolute**
 - ✦ bloc placé précisément par rapport aux bords de la fenêtre d'affichage
 - ✦ la position peut être modifiée dynamiquement (déplacement)
 - **fixed**
 - ✦ bloc placé précisément par rapport aux bords de la fenêtre d'affichage
 - ✦ Reste immobile en cas de défilement
 - **static**
 - ✦ bloc placé en fonction de sa position dans l'ordre des balises, ne peut pas être modifiée dynamiquement

Propriétés de styles graphiques (2)

57

- Attribut **clip**
 - définit une zone de rognage du bloc
 - par exemple pour rogner une image
- Attribut **visibility** et **display**
 - indiquent si (visible|hidden) et comment un élément est affiché
- Attribut **z-index**
 - en cas de superposition de blocs d'affichage, indique l'ordre dans lesquels le navigateur doit les afficher (cf. logiciel de dessin)
- Remarques
 - les distances s'expriment en pixels (**px**), points (**pt**), unités métriques (**cm**, **mm**), ou pourcentages de la taille de la fenêtre (%)

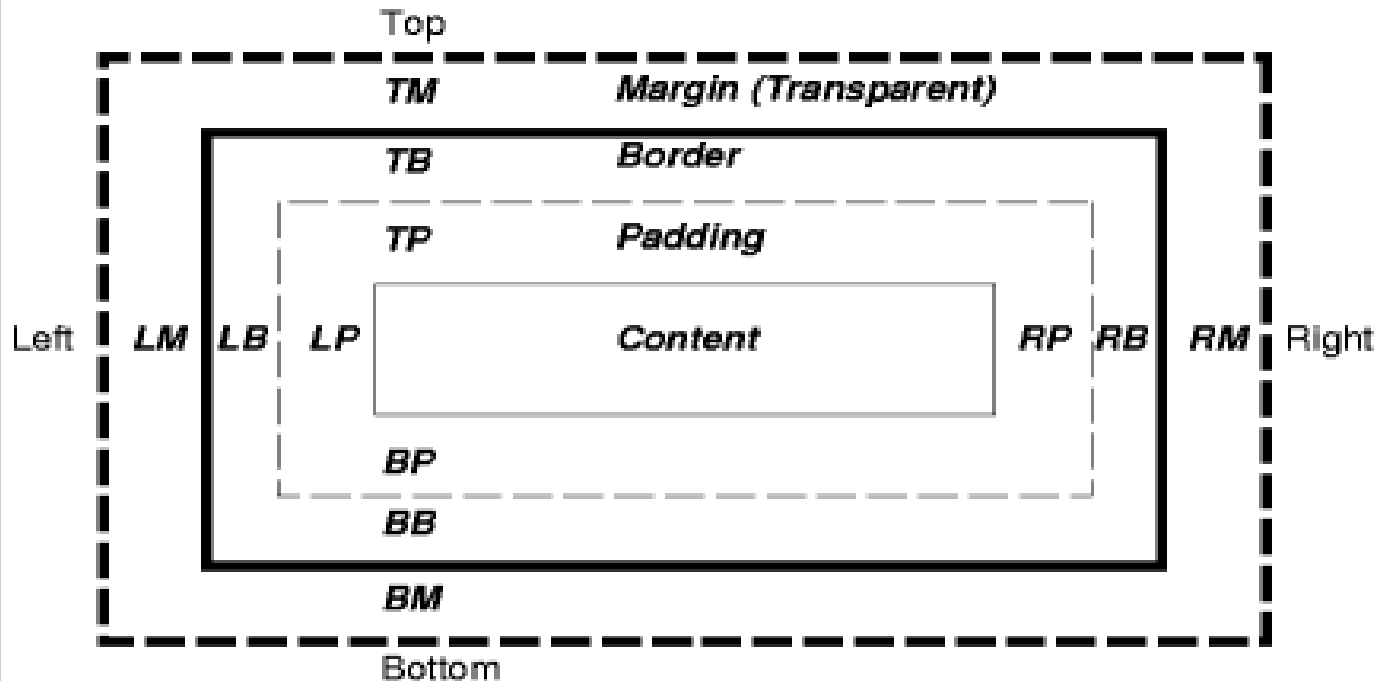
Propriétés de styles graphiques (3)

58

- **Arrière-plan**
 - background-color
 - ✦ couleur de fond
 - background-image
 - ✦ image de fond
 - background-repeat
 - ✦ types de répétition de l'image
 - background-attachement
 - ✦ arrière-plan fixe quand on utilise l'ascenseur
 - background-position
 - ✦ position par rapport au coin supérieur gauche
 - background
 - ✦ attribut de résumé
- **Exemple**
 - `p { background: image.gif fixed repeat-y }`

Marge, bordures, padding

59



- Margin edge
- Border edge
- - - Padding edge
- Content edge

CSS définit des attributs pour définir précisément ces zones

Styles intégrés

60

- Déclarés comme attribut **style** d'un élément
- Exemple
 - `<h1 style="font-family: Arial; font-style: italic;">Un texte qui se retrouvera en arial italique</h1>`

Feuilles de styles incorporées

61

- La feuille de style est déclarée dans l'en-tête (**head**) du document XHTML
- Elle s'appliquera aux éléments dans **body**
- Exemple

```
<html>
  <head>
    <title>Texte du titre</title>
    <style type="text/css">
      .important {color: red;}
      h1 {font-family: Arial; font-style: italic; }
      em { color: green }
    </style>
  </head>
  <body>
    <!-- éléments sur lesquels s'appliqueront les styles -->
  </body>
</html>
```

Feuilles de styles liées

62

- Les règles se trouvent dans un fichier extérieur
- On indique au navigateur où se trouve cette ressource de style, en utilisant un élément **link** dans l'en-tête
 - `<link rel="stylesheet" type="text/css" href="fichier-de-style.css" />`
- Intérêt
 - Plusieurs documents XHTML peuvent faire appel à la même feuille de style

Héritage de styles

63

- Par défaut
 - Les styles d'un élément sont hérités par ses éléments descendants
 - Exemple :
 - ✦ `<p style="color: red ;">Du texte mis en évidence, pas mis en évidence.</p>`
 - ✦ Du texte *mis en évidence*, pas mis en évidence.
- A condition que l'héritage ait un sens
 - *i.e.* que les caractéristiques soient applicables à l'élément enfant
 - ✦ un positionnement de bloc n'a pas d'intérêt pour un élément `em` qui y est contenu
 - si un style est défini spécialement pour un élément (ex. `em { color: blue; }`), l'héritage ne se fait pas
 - ✦ Du texte *mis en évidence*, pas mis en évidence.

Cascading style sheets : cascade

64

- On peut avoir concurrence entre plusieurs styles définis dans de multiples endroits
 - styles par défaut (1- navigateur)
 - fichiers CSS externes (2- spécifications globales au site)
 - élément **head** du document XHTML (3- spécification globales au doc.)
 - attributs **style** des éléments (4- spécification locales)
 - style utilisateur (5- spécification de l'utilisateur)
- Notion de cascade ou ordre de priorité des styles
 - trouver toutes les déclarations qui s'appliquent à un élément
 - les classer par spécificité
 - ✦ système de poids qui s'ajoutent
 - les classer par ordre d'apparence
 - ✦ plus une déclaration apparaît tard, plus elle a de poids
- Exemple
 - style (4) > style (4 hérité) > style (3) > style (2) > style (1)

Présentation avec des CSS

65

Information

XHTML
Doc

Présentation

Cascading
Style Sheet

User-defined
Cascading
Style Sheet

Web Server



Avantages des feuilles de style

- cohérence des présentations
- facilité de maintenance
 - modifier le contenu sans toucher au style
 - modifier le style sans toucher au contenu

Présentation avec XSLT

66

Information

XML
Doc

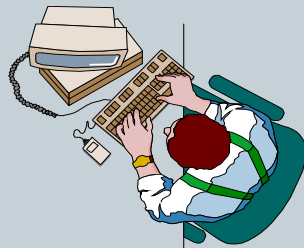
Transformation

XSLT

Presentation

XHTML
& CSS

Web Server



Avantages

- gestion de données dans des documents XML qui suivent une DTD précise
- XHTML + CSS

Conclusion sur CSS

67

- Actuellement : CSS 3
 - Recommandation du 29 septembre 2011
 - Amélioration des sélecteurs
 - Nombreuses nouvelles caractéristiques de style
 - ✦ Animations, transitions...
- Les navigateurs gèrent diversement les feuilles de style !
 - Tester avec plusieurs navigateurs
- Aller voir les références

Conclusion

68

- Présentation rapide de XHTML/CSS
- Après, il faut pratiquer
 - XHTML strict + CSS
- Très nombreuses ressources sur le Web
 - éditeurs, assistants, tutoriaux, forums...
 - références :
 - ✦ Les recommandations : <http://www.w3.org/>
 - ✦ Des tutoriaux : <http://www.w3schools.com/>
<http://www.laltruiste.com/> (en français)
<http://web.developpez.com/> (en français)
 - ✦ Les validateurs : <http://validator.w3.org/>
<http://jigsaw.w3.org/css-validator/>

Remerciements

69

- Professional Web Authoring With XHTML and CSS
– Roy Tennant