

# (X)HTML – CSS

LIONEL MÉDINI  
UFR INFORMATIQUE  
UNIVERSITÉ CLAUDE BERNARD LYON 1

D'après le cours de Yannick Prié  
2010-2011 – Master SIB  
M1 – UE 3 / Bloc 2

## Objectifs du cours

2

- Présenter des documents XML dans un navigateur web
  - Historique rapide de HTML
  - XHTML strict
  - CSS

## Le Web en un schéma

3

## HTML

4

- 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
  - Mé lange 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

5

- Séparation de la structure logique et de la présentation des 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

6

- 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 Working Draft (19 octobre 2010)

## HTML → XHTML : à savoir

7

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

## Plan

8

- XHTML
  - Structure générale d'un document XHTML
  - XHTML Strict et Transitionnal
  - Eléments les plus utiles
- CSS
  - Généralités
  - Règles et sélecteurs
  - Propriétés utiles
  - Placement des styles
  - Héritage et cascade

## Structure générale d'un document XHTML

9

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

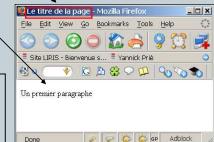
## Premier exemple

10

```
<?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">

<head>
  <title>Le titre de la page</title>
</head>
<body>
  <!-- un commentaire -->
  <p>Un premier
  paragraphe</p>
</body>
```

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 Transitionnal

11

- Transitionnal
  - 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

12

- 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

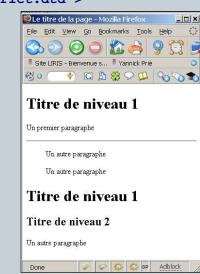
(13)

- **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
  - `<br />` → saut de ligne

## Corps : exemple

(14)

```
<?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

(15)

- **Emphase**
  - `<em> ... </em>` (emphasis)
- **Emphase forte**
  - `<strong> ... </strong>`
- **Exposants et indices**
  - `<sup> ... </sup>` (superscript)
  - `<sub> ... </sub>` (subscripted)

## Extraits, citations et références

(16)

- Citer quelque chose : **cite**
  - ... `<cite>Discours de la méthode</cite>` de `<cite>Descartes</cite>` ...
- Citation courte : **q**
  - ... `<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/">`
  - `Rappelle-toi Barbara</p>`
  - ...
  - `<p>Dont il ne reste rien.</p>`

## Autres balises de structuration

(17)

- **Texte spécial**
  - `<pre> ... </pre>` (preformatted text)
  - `<code> ... </code>`
    - \* pour mettre du texte représenté 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

(18)

- **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" />`

## Images : exemple

(19)

Images : exemple

...  
  
  
  
...  

- 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

(20)

Listes

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

## Listes : exemple

(21)

Listes : exemple

...  

- Un
- Deux
- Trois

4. Quatre
5. Cinq
6. Six

Sept

Le septième chiffre

Huit

Le huitième chiffre

...

## Liens hypertextes

(22)

Liens hypertextes

- 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
  - `<a href="URL" title="texte">`  
 ancre  
`</a>`
  - `href` : URL de destination du lien
  - `title` : texte qui apparaîtra dans une info-bulle si on survole le lien

## HTML : notion d'URL

(23)

HTML : notion d'URL

...  

- Uniform Ressource Locator
  - permet d'identifier une ressource sur le réseau
- Une ressource peut être
  - une page Web
  - une image (seule ou utilisée dans une page Web)
  - un programme
  - un fichier à télécharger...
- Une URL indique
  - un protocole (langage de communication entre deux programmes sur deux machines)
    - FTP (File Transfert Protocol),
    - HTTP (HyperText Transfert Protocol)...
  - l'adresse d'un serveur
  - un chemin dans l'arborescence des fichiers
- Forme générale : protocole://adresse/chemin
  - Exemples
 

`http://www.univ-lyon1.fr/`  
`http://www710.univ-lyon1.fr/~lmedini/SIB32/CM/5_6_HTML_et_CSS.pdf`

## Liens hypertextes : exemples

(24)

Liens hypertextes : exemples

...  

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

`<p><a href="URL" title="serveur web de l'UCBL">Cliquez-ici</a>`

`<p>pour atteindre le site de l'université Claude Bernard Lyon 1.</p>`

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

`<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://www3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">Cliquez-ici pour atteindre le site de l'Université Claude Bernard Lyon 1.</p>`

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

`<p>Evidemment, on peut mettre <a href="mailto:mailto:univ-lyon1@w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">Cliquez-ici pour atteindre le site de l'Université Claude Bernard Lyon 1.</p>`

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

`<p>Evidemment, on peut mettre pas mal de contenu (texte) dans une ancre.</p>`

Evidemment, on peut mettre pas mal de contenu (texte) dans une ancre.

Evidemment, on peut mettre pas mal de contenu (texte) dans une ancre.

## Tableaux

25

- **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

26

Type de quelques balises XHTML	
Nom de la balise	Type
div	block
span	inline
table	block

## Deux types d'éléments en HTML

27

- **Éléments 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.
- **Éléments inline**
  - Dans le flux des caractères
  - « Semblables » à des caractères
  - Exemple
    - em, img, strong, a, etc.

## Les éléments span et div

28

### • Utilité

- Regrouper des sous-parties de document
  - Pour leur donner un nom
  - Pour enclose (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
- <span> ... </span>
  - 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)

29

**Exemples avec <code>div</code> et <code>span</code>**

```

<h1>Exemples avec <code>div</code> et <code>span</code></h1>
<div id="premiere_strophe" style="color: purple; font-family: serif; font-size: 1.2em; margin-bottom: 1em; text-align: center; width: 100%;>
<span>Le Capitaine Jonathan, </span>
<span>Eant âgâ de dix-huit ans </span>
<span>Capitaine un jour un pâlican </span>
<span>Dans une île d'Extrême-orient. </span>
<span>Le pâlican de <span class="heros">Jonathan</span></span></div>
<p>Etant âgâ de dix-huit ans</p>
<p>Capture un jour un <span style="font-weight:bold">pâlican</span></p>
<p>Dans une île d'Extrême-orient.</p>
<div style="text-align: right;>
<span class="heros">Jonathan</span></div>
<p>A matin, pond un œuf tout blanc</p>
<p>Et il en sort un <span style="font-weight:bold">pâlican</span></p>
<p>Lui ressemblant étonnamment.</p>
<div style="text-align: justified;>
<span style="font-weight:bold">pâlican</span></div>
<p>Et ce deuxième œuf tout blanc</p>
<p>D'où sort, inévitablement</p>
<p>Un autre, qui en fait autant. </p>
<div style="text-align: right;>
<span style="font-style: italic; font-weight:bold">pendant</span> très longtemps</div>
<p>Cela peut durer pendant très longtemps </p>
<p>Si l'on ne fait pas d'omelette avant. </p>

```

**Exemples avec div et span**

Le Capitaine Jonathan,  
Eant âgâ de dix-huit ans  
Capitaine un jour un pâlican  
Dans une île d'Extrême-orient.

Le pâlican de Jonathan  
A matin, pond un œuf tout blanc  
Et il en sort un pâlican  
Lui ressemblant étonnamment.

Et ce deuxième pâlican  
Pond, à son tour, un œuf tout blanc  
D'où sort, inévitablement  
Un autre, qui en fait autant.

Cela peut durer pendant très longtemps  
Si l'on ne fait pas d'omelette avant.

## Formulaires

30

### • 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

(31)

```

<form method="post" action="http://serveur.com/script.php">
  <input type="text" name="Champ_saisie" value="Texte" />
  <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>
  <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 />
  <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 />
  <input type="reset" name="Annulation" value="Annuler">
  <input type="submit" name="Soumission" value="Soumettre">
</form>

```

## Frames

(32)

- 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**
  - Né pas utiliser car pose beaucoup de problèmes
    - gestion
    - indexation

## Frames : exemple

(33)

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>
</html>

```

## Conclusion sur XHTML

(34)

- 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 plugins
    - Applets JAVA, animations Flash, audio, vidéo, etc.
    - <object> ... </object>
  - Les sites web et leur gestion (conception, installation sur serveurs web...)

## Plan

(35)

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

## CSS : généralités

(36)

- Objectif**
  - Décrire comment un document XHTML doit être affiché
    - Remplace les éléments d'affichage des anciennes versions de HTML
    - 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**

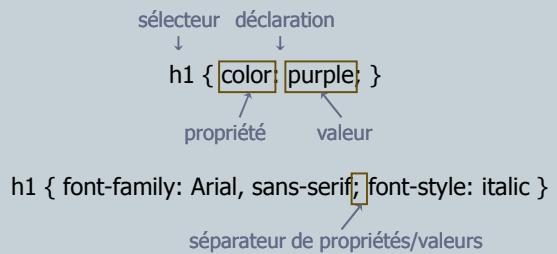
## Feuille de style CSS

(37)

- 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 règles 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

(38)



## Exemple de règle CSS

(39)

```

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

## Types de sélecteurs

(40)

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

## Sélecteurs CSS : simple et groupe

(41)

- 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 : classes

(42)

- 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

43

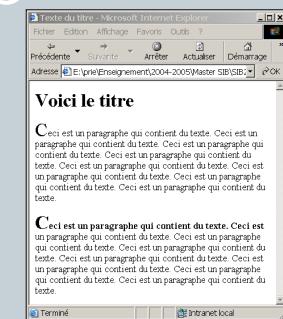
- Sélecteurs qui sélectionnent des éléments en fonction de leur état à un moment donné
- Exemple et intérêt principal
  - `a:link` — lien non visité et inactif
  - `a:hover` — lien sur lequel passe le pointeur de la souris
  - `a:active` — lien sur lequel on clique
  - `a:visited` — lien déjà visité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

44

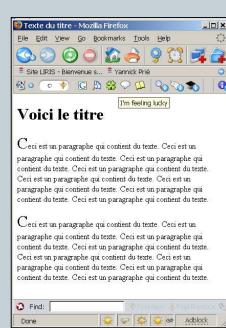
- `:first-letter`
  - première lettre dans un élément bloc (ex. p, h1, ...)
- `:first-line`
  - première ligne dans un élément bloc (ex. p, h1, ...)



## Pseudo-éléments : exemple

45

```
<?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

46

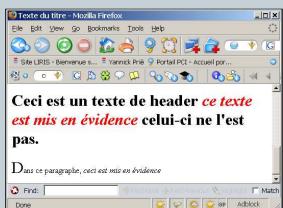
- Sélecteurs qui ne sélectionnent que des éléments dans un certain contexte

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

47

- `font-size`:
  - `small` | `medium`... | `%` | `x pt`
- `font-family`:
  - `fontname1, fontname2` (*si la première n'est pas disponible*), `familyname` (*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

48

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

■ Black = "#000000"	■ Green = "#008000"
■ Silver = "#C0C0C0"	■ Lime = "#00FF00"
■ Gray = "#808080"	■ Olive = "#808000"
■ White = "#FFFFFF"	■ Yellow = "#FFFF00"
■ Maroon = "#800000"	■ Navy = "#000080"
■ Red = "#FF0000"	■ Blue = "#0000FF"
■ Purple = "#800080"	■ Teal = "#008080"
■ Fuchsia = "#FF00FF"	■ Aqua = "#00FFFF"

## Types d'éléments

49

- 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

50

- 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

51

- S'appliquent aux éléments blocs

- Attribut **position**

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

## Propriétés de styles graphiques (2)

52

- 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)

53

- Arrière-plan

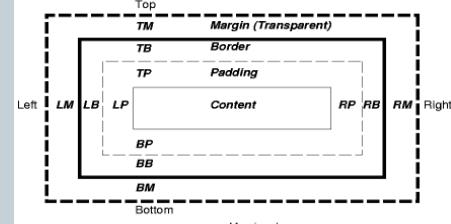
- **background-color**
  - couleur de fond
- **background-image**
  - image de fond
- **background-repeat**
  - types de répétition de l'image
- **background-attachment**
  - arrière-plan fixe quand on utilise l'ascenseur
- **background-position**
  - position par rapport au coin supérieur gauche
- **background**
  - attribut de résumé

- Exemple

- p { background: image.gif fixed repeat-y }

## Marge, bordures, padding

54



CSS définit des attributs pour définir précisément ces zones

## Styles intégrés

55

- 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

56

- 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

57

- Les règles se trouvent dans un fichier extérieur
- On indique au navigateur où ce 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

58

- Par défaut
  - Les styles d'un élément sont héritées par ses éléments descendants
  - Exemple :
    - × `<p style="color: red ;">Du texte <em>mis en évidence</em>, 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.`

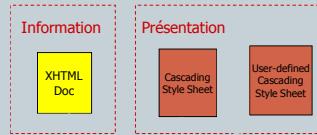
## Cascading style sheets : cascade

59

- 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 un déclaration apparaît tard, plus elle a de poids
- Exemple
  - style (4) > style (4 hérité) > style (3) > style (2) > style (1)

## Présentation avec des CSS

60



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

