

Langage HTML



<HyperText Markup Language>

*« Je n'ai fait que prendre le principe d'-
hypertexte et le relier au principe du
TCP et du DNS et alors – boum ! – ce fut
le World Wide Web ! »
Tim Berners-Lee*

Objectifs



- Se familiariser avec le codage HTML
- Acquérir une pratique minimale
- Etre capable de mener à bien la construction d'un site ou d'une page en respectant les bonnes pratiques
- Etre informé des dernières implémentations des standards du web
- Acquérir les capacités d'auto-formation nécessaires pour suivre les évolutions à venir

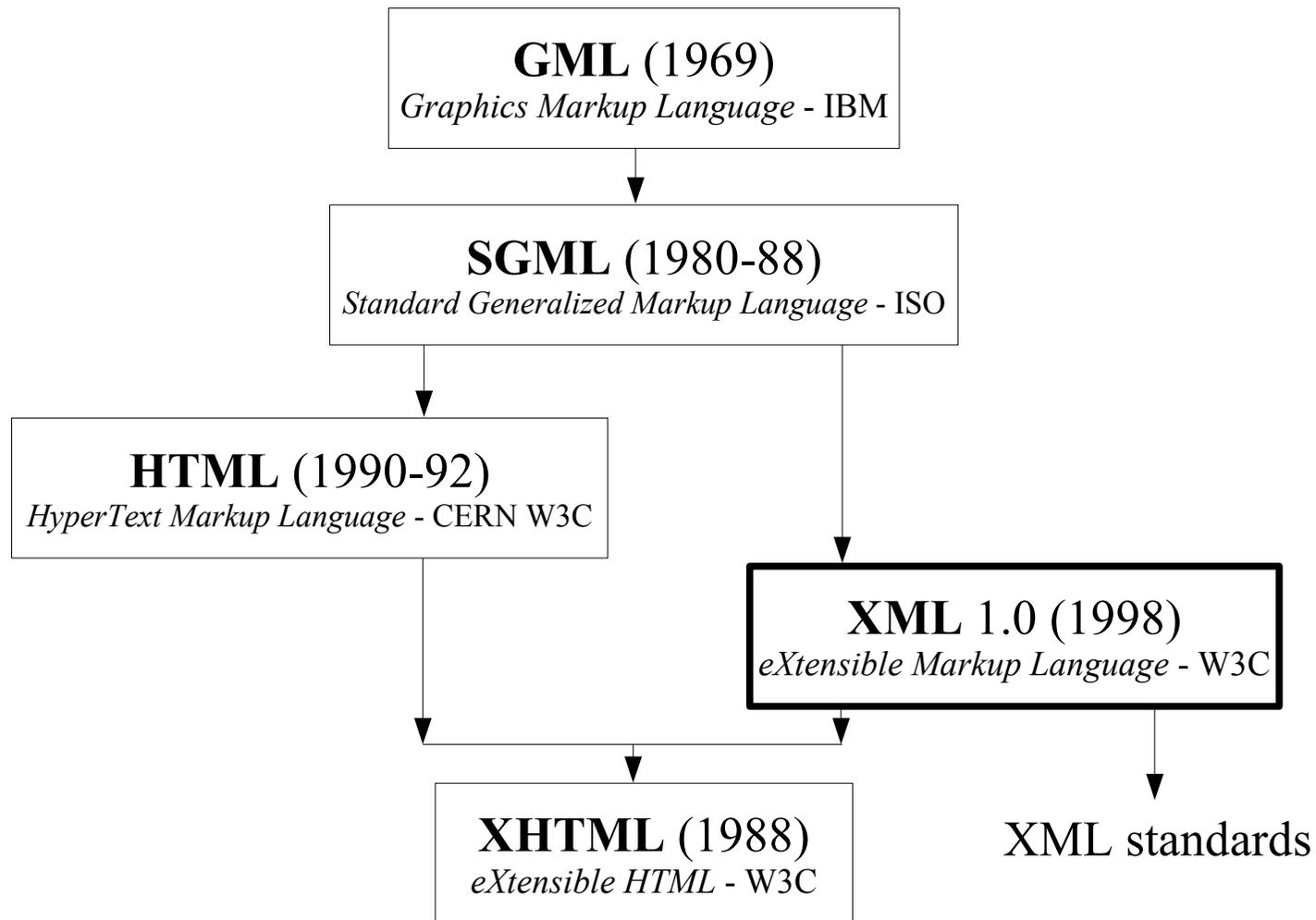
Introduction



- langage de description de document
- format de données conçu pour représenter des documents hypertextuels, appelés aujourd'hui « pages web »
- langage de balisage (*Markup Language*) qui permet d'écrire de l'hypertexte (liens)
- permet également de structurer sémantiquement et de mettre en forme le contenu des pages
- initialement dérivé du SGML (*Standard Generalized Markup Language*), jugé trop complexe.

Remarque : HTML n'est ni un protocole ni un langage de programmation

Arbre généalogique



Historique



- **1991 : Origine**

HTML est une des trois inventions à la base du World Wide Web, avec HTTP (Hypertext Transfer Protocol) et les adresses web (URI/URL). Les premiers éléments du langage HTML comprennent le titre du document, les hyperliens, la structuration du texte en titres, sous-titres, listes ou texte brut ...

- **1993 : Apports de NCSA Mosaic**

Avec ce navigateur, HTML connaît deux inventions majeures : l'élément IMG qui permet d'intégrer des images et l'apparition des formulaires qui rend le web interactif.

- **1994 : Apports de Netscape Navigator**

Avec l'apparition de ce navigateur, le support de nombreux éléments de présentation est ajouté : attributs de texte, clignotement, centrage, etc. Le développement de HTML prend alors deux voies divergentes : maximiser l'impact visuel des pages web et étendre les capacités de description sémantique.

- **1997 : HTML 4.0**

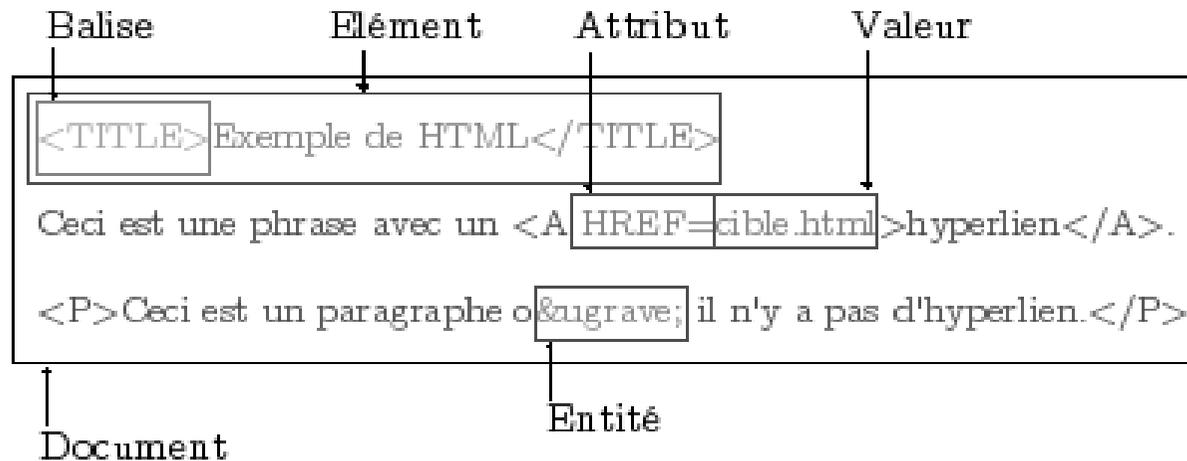
Le W3C publie la spécification HTML 4.0 qui standardise de nombreuses extensions supportant les styles et les scripts, les cadres (frames) et les objets (inclusion généralisée de contenu).

- **1999 : dernière spécification de HTML, la 4.01**

Syntaxe



- Les navigateurs appliquent la loi de Postel : "Soyez libéral dans ce que vous acceptez, et conservateur dans ce que vous envoyez" (RFC 791)
- Malgré les libertés prises avec la norme, la terminologie propre à SGML est utilisée : document, élément, attribut, valeur, balise, entité, validité, etc...



Créer des documents, un site (!)



- Créer des documents HTML avec un éditeur de texte ou un logiciel spécialisé (extension conseillée : **.html**)
- Visualiser avec un navigateur (le code HTML est interprété côté client)
- Pour mettre un site en ligne, il faut : un accès internet (ou intranet) et un espace d'hébergement sur un serveur HTTP (chez un hébergeur payant ou gratuit par exemple)
- Mettre le site en ligne (déployer) : transférer sur le serveur HTTP les documents et l'arborescence, en utilisant par exemple ftp

Créer des documents, un site (II)



Avertissements

- Techniques : évitez les espaces et les caractères accentués dans les noms de fichiers et les chemins (par exemple, l'espace est codé %20 dans les URL), utilisez les chemins relatifs (les chemins absolus ne sont pas portables), utilisez des minuscules (les URL sont sensibles à la casse), ... voir les bonnes pratiques
- Légaux : respectez les droits d'auteur, les droits d'éditeur, les décisions de justice, les droits à l'image , ...

Structure d'un document (I)



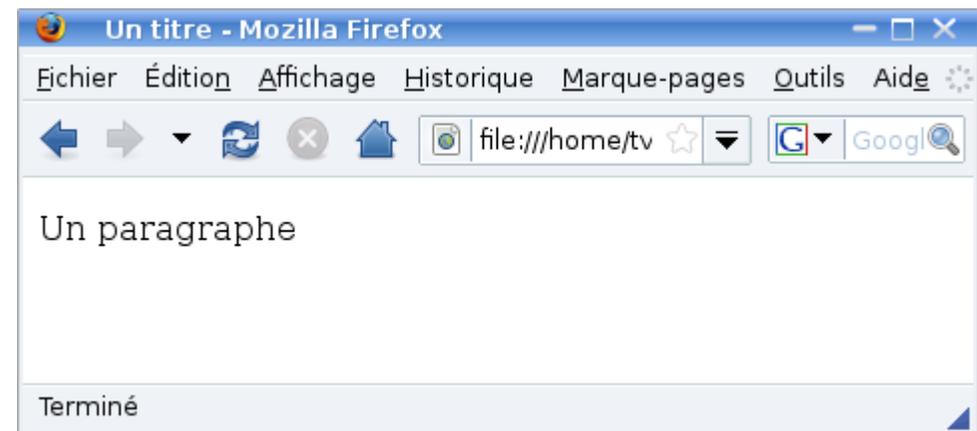
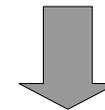
```
<html>
  <head>
    <title> Un titre </title>
  </head>
  <body>
    <!-- corps de la page -->
  </body>
</html>
```

en tête

corps

page

```
<HTML>
  <HEAD>
    <TITLE>Un titre</TITLE>
  </HEAD>
  <BODY>
    <P>Un paragraphe</P>
  </BODY>
</HTML>
```



Structure d'un document (II)



- Avec l'introduction des CSS (*Cascading Style Sheets*) et du DOM (*Document Object Model*), il a fallu considérer que les documents HTML ont une véritable structure en arbre, avec un élément racine contenant tous les autres éléments.

Source HTML	Modèle du document
<pre><!DOCTYPE html PUBLIC "-//IETF//DTD HTML 2.0//EN"> <html> <head> <title> Exemple de HTML </title> </head> <body> Ceci est une phrase avec un hyperlien. <p> Ceci est un paragraphe où il n'y a pas d'hyperlien. </p> </body> </html></pre>	

Structure d'un document (III)



- Un document HTML valide déclare la version HTML qui est utilisée dans le document. La déclaration de type de document indique la définition de type de document (DTD) en vigueur pour le document.
- HTML 4.01 spécifie trois DTD et les auteurs doivent donc inclure l'une des déclarations de type de document suivantes dans leurs documents. Les DTD varient selon les éléments qui y sont gérés.
 - Strict comprend tous les éléments et attributs qui ne sont pas déconseillés :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```
 - Transitoire (conciliante) inclut la totalité du DTD strict auquel se rajoutent les éléments et attributs déconseillés (la plupart d'entre eux concernant la présentation visuelle) :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```
 - Jeu d'encadrement inclut la totalité du DTD transitoire plus les cadres :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
```

DTD



- La *Document Type Definition* (DTD), ou Définition de Type de Document, est un document permettant de décrire un modèle de document SGML ou XML.
- Une DTD indique les noms des éléments pouvant apparaître et leur contenu, c'est-à-dire les sous-éléments et les attributs. En dehors des attributs, le contenu est spécifié en indiquant le nom, l'ordre et le nombre d'occurrences autorisées des sous-éléments. L'ensemble constitue la définition des hiérarchies valides d'éléments et de texte.
- Le document sera jugé valide lorsqu'il possède et respecte sa DTD (*Definition Type Document*).
- Déclaration des noms de balises autorisés : `<!ELEMENT le_nom (le_contenu)>`
- Déclaration des attributs (noms et type) : `<!ATTLIST élément attribut_1 type_1 valeurDéfaut_1 ...>`

Eléments



- La version 4 de HTML décrit 91 éléments (ou balise).

`<BALISE>...</BALISE>` ou `<BALISE>`

- Les fonctionnalités implémentées par HTML peuvent être réparties ainsi : structure générale d'un document HTML, informations sur la langue, marquage sémantique, listes, tables, hyperliens, inclusion d'images, d'applets et d'objets divers, éléments de regroupement, style de la présentation, marquage de présentation du texte, cadres, formulaire pour l'insertion interactive de données, scripts
- La spécification HTML 4.01 : <http://www.la-grange.net/w3c/html4.01/>

Attributs



- Les attributs permettent de préciser les propriétés des éléments HTML.

`<BALISE NOM_ATTRIBUT="Valeur">...</BALISE>`

- Il y a 188 attributs dans la version 4 de HTML :
 - certains attributs s'appliquent à presque tous les éléments
 - d'autres attributs sont propres à un élément unique (ou similaire).
 - la plupart des attributs sont facultatifs.
 - quelques éléments ont cependant des attributs obligatoires (attributs SRC et ALT pour l'élément IMG)
 - certains attributs sont enfin de type booléen (exemple *selected*).
- Index des attributs : <http://www.w3.org/TR/html401/index/attributes.html>

L'en-tête du document



- L'entête (optionnel) du document HTML est délimité par l'élément HEAD qui contient des informations sur le document, tels que son titre (élément TITLE obligatoire), et d'autres données (les métadonnées) qui ne sont pas considérées comme faisant partie du contenu du document (BODY).
- L'élément META représente un mécanisme générique pour spécifier des métadonnées. On peut utiliser l'élément META pour identifier les propriétés d'un document (par exemple, l'auteur, la date d'expiration, une liste de mots-clés, etc.) et assigner des valeurs à ces propriétés.

L'élément META (I)



- L'élément META représente un mécanisme générique pour spécifier des méta-données (les propriétés du document). Chaque élément META spécifie un couple propriété/valeur. L'attribut *name* identifie la propriété et l'attribut *content* en spécifie la valeur. Par exemple :

```
<META name="Auteur" content="Thierry Vaira">
```

- L'élément META et les en-têtes HTTP : l'attribut *http-equiv* (à la place de *name*), a un sens particulier quand les documents sont ramenés via le protocole de transfert hypertexte HTTP. Par exemple :

```
<META http-equiv="Expires" content="Tue, 20 Aug 1996 14:25:27 GMT">
```

- L'élément META et les moteurs de recherche : une utilisation courante consiste à spécifier des mots-clés qu'un moteur de recherche peut utiliser pour améliorer la pertinence du résultat d'une recherche. Par exemple :

```
<META name="keywords" lang="fr" content="vacances, soleil">
```

L'élément META (II)



- On peut utiliser l'élément META pour spécifier les informations par défaut sur un document dans les cas suivants :
 - le langage de script par défaut ;
 - le langage de feuille de style par défaut ;
 - l'encodage de caractères du document.
- L'exemple suivant spécifie l'encodage de caractères du document comme étant "ISO-8859-15" :

```
<META http-equiv="Content-Type" content="text/html; charset=ISO-8859-15">
```

Le corps du document



- Le corps du document (BODY) représente le contenu du document.
- On peut assimiler le corps à un canevas dans lequel s'inscrit le contenu : le texte, les images, les couleurs, les graphiques, etc.
- Puisque les feuilles de style (CSS) sont désormais le moyen conseillé de spécifier la présentation d'un document, les attributs de présentation des éléments sont donc déconseillés.
- Exemple déconseillé : `<H1 align="center"> Comment tailler le bois </H1>`
- En utilisant le langage CSS, par exemple, on pourrait obtenir le même effet par :

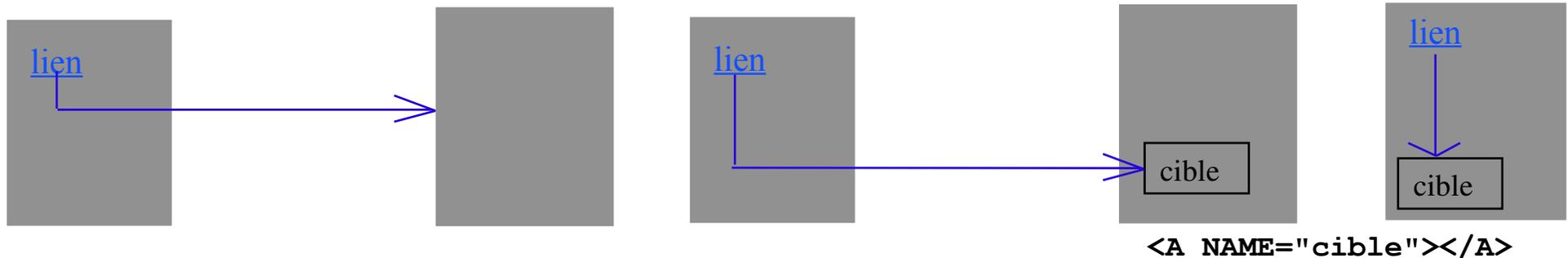
```
<HEAD>
<TITLE>Comment tailler le bois</TITLE>
<STYLE type="text/css">
  H1 { text-align: center}
</STYLE>
<BODY>
<H1>Comment tailler le bois</H1>
```

```
<HEAD>
<TITLE>Comment tailler le bois</TITLE>
<STYLE type="text/css">
  H1.bois {text-align: center}
</STYLE>
<BODY>
<H1 class="bois">Comment tailler le
bois</H1>...
```

Les liens hypertextes (I)



- Un hyperlien (ou lien hypertexte) a une source (ou origine) et une destination (ou cible). L'activation de l'élément source (par un simple clic) d'un hyperlien permet de passer automatiquement à sa destination.



`lien`

`lien`

- La source d'un hyperlien est généralement un élément (mots, phrases, images) d'un document hypertexte.
- La destination peut être un autre élément du même document, il s'agit alors d'un hyperlien interne au document. La destination peut également être un autre document, voire un élément précis d'un autre document.
- Les hyperliens du web sont unidirectionnels.

Les liens hypertextes (II)



- Les hyperliens du web ne garantissent pas que la destination existe : dans ce cas on parle des liens dits brisés, cassés ou morts. Ils font alors référence à un document non trouvé (erreur 404).
- Le problème du changement d'adresse de la destination peut être résolu par diverses approches : les permaliens (utilisés notamment sur les blogs), la notion d'identifiant pérenne (par exemple le système ARK pour *Archival Resource Key*).
- Les liens web utilisent la notation des URI (*Uniform Resource Identifier*) pour identifier les documents référencés.
- Le *direct linking* (lien direct) ou *hotlinking* consiste à utiliser l'adresse d'un fichier publié sur un site web (le plus souvent une image) pour l'afficher sur un autre site.
- Même dans le cas où l'image est libre de droits, un tel lien est illégal s'il n'est pas expressément autorisé par l'éditeur du site web concerné, puisqu'il s'agit littéralement d'un vol de bande passante.

URI/URL



- Un URI (*Uniform Resource Identifier*) soit littéralement « identifiant uniforme de ressource », est une courte chaîne de caractères identifiant une ressource sur un réseau (par exemple une ressource Web) physique ou abstraite, et dont la syntaxe respecte une norme d'Internet mise en place pour le World Wide Web (voir RFC 3986).
- Un URL (*Uniform Resource Locator*) littéralement « localisateur uniforme de ressource », est une chaîne de caractères utilisée pour adresser les ressources du World Wide Web : document HTML, image, son, forum Usenet, boîte aux lettres électronique, etc. Elle est aussi appelée une adresse web.
- Le format d'une adresse web ou URL :

`protocole://[<login>:<mot_de_passe>]<serveur>[:<numero_port>]/[<chemin>/]<ressource>`

Exemples : `http://www.example.com/tim/page.html`

`http://192.168.52.83/index.php`

`ftp://ftp.is.co.za:21/rfc/rfc1808.txt`

L'attribut *id*



- L'attribut *id* assigne un identifiant unique à un élément.
- L'attribut *id* a plusieurs rôles dans HTML :
 - comme sélecteur dans une feuille de style ;
 - comme ancre cible de liens hypertextes ;
 - comme moyen d'appeler un élément particulier à partir d'un script ;
 - comme nom d'un élément OBJECT déclaré ;
 - pour un traitement universel par les agents utilisateurs (par exemple, pour identifier les champs lors de l'extraction des données d'un formulaire à partir d'un script PHP, etc.).

Les éléments DIV et SPAN



- Les éléments DIV et SPAN, en conjonction avec les attributs *id* et *class*, offrent un mécanisme générique de regroupement d'éléments qui rajoute de la structure aux documents.
- Ces éléments définissent le contenu comme étant en-ligne (SPAN) ou de bloc (DIV) mais n'imposent aucune autre expression de présentation sur le contenu.
- Les éléments de bloc créent des structures « plus grandes » : ils peuvent contenir des éléments en-ligne et d'autres éléments de bloc
- Les éléments en-ligne ne peuvent contenir que des données et d'autres éléments en-ligne.
- Ces éléments sont notamment très utilisés conjointement avec les feuilles de style (CSS).

Les feuilles de style (I)



- Les feuilles de style CSS (*Cascading Style Sheets* : feuilles de style en cascade) sert à décrire la présentation des documents HTML et XML.
- Les standards définissant CSS sont publiés par le W3C.
- Introduit au milieu des années 1990, CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web dans les années 2000.
- Les enjeux sont :
 - Séparer la structure d'un document de ses styles de présentation
 - Décliner les styles de présentation selon le récepteur
 - Permettre la cascade des styles
- Spécification CSS1 : <http://www.yoyodesign.org/doc/w3c/css1/index.html>
- Spécification CSS2: <http://www.yoyodesign.org/doc/w3c/css2/cover.htm>

Les feuilles de style (II)



- Une règle CSS consiste en deux parties : un sélecteur ('H1') et une déclaration ('color: blue'). La déclaration comprend deux parties : la propriété ('color') et la valeur associée ('blue').

```
sélecteur[, sélecteur2] { propriété1: valeur; propriété2: valeur }  
H1 { color: blue }
```

- Le sélecteur est le lien entre le document HTML et la feuille de style, tous les types d'éléments HTML pouvant être des sélecteurs.
- Un sélecteur est une chaîne de caractères qui identifie les éléments sur lesquels s'applique la règle correspondante. Un sélecteur est ou bien simple (ex. 'H1') ou bien contextuel (fait de plusieurs sélecteurs simples, p.ex. 'H1 B').
- Liste des propriétés et des valeurs :
<http://www.yoyodesign.org/doc/w3c/css1/index.html#css1-properties>

Les feuilles de style (III)



- Les différentes syntaxes pour les sélecteurs :
 - L'attribut *class* : on l'adresse en le faisant précéder d'un point '.'

```
H1.pastoral { color: #00FF00 }  
.pastoral { color: green } /* tous les éléments de la classe pastoral */
```
 - L'attribut *id* : on l'adresse en le faisant précéder d'un dièse '#'

```
H1#prenom { letter-spacing: 0.5em }  
#prenom { letter-spacing: 0.3em }
```
 - Les sélecteurs contextuels peuvent être spécifiés au travers des types d'élément, des attributs CLASS, des attributs ID ou d'une combinaison de ceux-ci :

```
DIV P { font: small sans-serif } /* les balises appartenant à DIV */  
  .reddish H1 { color: red }  
  #x78y CODE { background: blue }  
  DIV.sidenote H1 { font-size: large }  
  H1 B, H2 B, H1 EM, H2 EM { color: red } /* regroupement */
```
 - Les pseudo-classes d'ancre (élément <A>) :

```
A:link { color: red } /*lien non-visité*/ A:visited { color: blue } /*lien visité*/ ...
```

Les feuilles de style (IV)



- Pour que les feuilles de style puissent avoir une influence sur la présentation, les agents utilisateurs (les navigateurs par exemple) doivent en connaître l'existence.
- Cet exemple montre quatre façons de combiner le style au HTML :

```
<HTML>
```

```
<HEAD><TITLE>Titre</TITLE>
```

```
<LINK REL=STYLESHEET TYPE="text/css"  
  HREF="http://style.com/cool.css" TITLE="Cool">
```

①

```
<STYLE TYPE="text/css">  
  @import url(http://style.com/basic.css);  
  H1 { color: blue }  
</STYLE>
```

②

③

```
</HEAD>
```

```
<BODY>
```

```
<H1>Ce titre est en bleu</H1>
```

```
<P STYLE="color: green">et le paragraphe est en vert.
```

④

```
</BODY>
```

```
</HTML>
```

XML



- XML signifie eXtensible Markup Language (Langage de balisage extensible)
- XML est standardisé par la spécification W3C : <http://www.yoyodesign.org/doc/w3c/xml11/>
- XML n'est pas un langage de programmation
- XML est un méta-langage exploitable pour créer d'autres langages
- XML bien formé signifie que le texte XML obéit aux règles syntaxiques de XML
- XML valide signifie que le texte XML est bien formé et répond à une structure définie par une DTD
- XML est destiné à l'échange d'informations et de documents
- XML est une solution pour la modélisation des contenus et la standardisation de modèles de contenus
- XML vs HTML : syntaxe plus stricte, aucune balise prédéfinie, fixe et figée (XML est un méta-langage), X pour eXtensible qui permet de définir ses balises selon ses besoins, gère tous les jeux de caractères, séparation stricte contenu/présentation
- XML est plus qu'un langage, c'est une famille de langages. Actuellement on estime que plusieurs centaines de « langages » basés sur XML ont été décrits : XHTML, XSL, XSLT, Xpath, XLink, XPointer, XML-Schema, RSS, MathML, SVG, OpenDocument, ...

XHTML



- Un document XHTML étant un document XML
- Il commence par une déclaration XML indiquant la version et le jeu de caractère utilisé.
- Un document XHTML strictement conforme doit recouvrir tous les critères suivants :
 - 1 . Il doit être validé par l'une des trois DTDs (Strict, Transitional ou Frameset).
 - 2 . L'élément racine du document doit être <html>.
 - 3 . L'élément racine du document doit nommer l'espace nominatif XHTML en utilisant l'attribut xmlns. L'espace nominatif pour XHTML est défini par <http://www.w3.org/1999/xhtml>.
 - 4 . Une déclaration DOCTYPE doit être présente dans le document avant l'élément racine et doit faire référence à l'une des trois DTDs. Par exemple :

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "DTD/xhtml1-strict.dtd">
```

XHTML



- Etant donné que XHTML est une application XML, quelques habitudes qui étaient parfaitement légales dans le HTML 4 doivent être changées.
- En résumé, les balises et les attributs suivent quelques règles simples :
 - les noms des balises doivent être écrits en minuscules ;
 - toute balise ouverte est fermée (par exemple `<html>...</html>` ou `
`) ;
 - les noms des attributs doivent être écrits en minuscules ;
 - les valeurs des attributs doivent être entre guillemets doubles.
- Spécification XHTML 1.0 : <http://www.la-grange.net/w3c/xhtml1/>

DHTML



- HTML dynamique, (*Dynamic HTML*, souvent abrégé DHTML), est un nom générique donné à l'ensemble des techniques utilisées par l'auteur d'une page web pour que celle-ci soit capable de se modifier elle-même en cours de consultation dans le navigateur Web.
- Techniquement, les modifications portent sur la représentation interne que le navigateur se fait de la page web. La représentation interne est initialement déterminée par le document HTML et les informations de style CSS. Les modifications sont décrites avec un programme informatique en langage de script comme *JavaScript*, qui accède à la représentation interne à travers l'interface de programmation *Document Object Model* (DOM).
- Le DHTML n'a jamais désigné une technologie précise (DHTML n'est pas un standard ou une recommandation du W3C). L'abréviation DHTML et le sens actuel sont apparus en 1997 lorsque Netscape et Microsoft l'utilisèrent comme slogan marketing vantant une évolution technologique. DHTML devait signifier que désormais toute la structure d'un document HTML était librement manipulable.

DOM



- Le *Document Object Model* (ou DOM) est une recommandation du W3C qui décrit une interface indépendante de tout langage de programmation et de toute plate-forme, permettant à des programmes informatiques (et à des scripts) d'accéder ou de mettre à jour le contenu, la structure ou le style de documents. Le document peut ensuite être traité et les résultats de ces traitements peuvent être réincorporés dans le document tel qu'il sera présenté.
- Le Modèle Objet de Document (DOM) est une interface de programmation d'applications (API) pour documents HTML et XML. Il définit la structure logique des documents et la manière dont un document est accédé et manipulé. Spécification : <http://xmlfr.org/w3c/TR/REC-DOM-Level-1/>
- DOM est conçu pour être utilisé avec n'importe quel langage de programmation.

Rechercher sur le Web Ctrl+K

Téléchargements Ctrl+Y

Modules complémentaires

Gestionnaire de session

Adblock

Console d'erreurs

Inspecteur DOM Ctrl+Maj+I

Informations sur la page Ctrl+I

Effacer mes traces... Ctrl+Maj+Suppr

Inspecteur DOM

Fichier Edition Rechercher Afficher Aide

http://192.168.52.2/tests/entrer.html

Document - DOM Nodes

nodeName	id	class
- #document		
html		
HTML		
HEAD		
BODY		
#text		
DIV	entrer	
#text		
IMG		
#text		
#text		

Objet - DOM Node

Nom du nœud : IMG

URI de l'espace de nom :

Type du nœud : 1

Valeur du nœud :

nodeName	nodeValue
alt	entrez !
src	./logo.gif

Ajax



- AJAX est un acronyme signifiant *Asynchronous JavaScript and XML* (« XML et Javascript asynchrones »), et désignant une solution informatique libre pour le développement d'applications Web.
- À l'image de DHTML, AJAX n'est pas une technologie en elle-même, mais un terme qui évoque l'utilisation conjointe d'un ensemble de technologies libres couramment utilisées sur le Web :
 - HTML (ou XHTML) pour la structure sémantique des informations ;
 - CSS pour la présentation des informations ;
 - DOM et JavaScript pour afficher et interagir dynamiquement avec l'information présentée ;
 - l'objet XMLHttpRequest pour échanger et manipuler les données de manière asynchrone avec le serveur Web.
 - XML pour remplacer le format des données informatives (JSON) et visuelles (HTML).

XMLHttpRequest



- XMLHttpRequest est un objet ActiveX ou Javascript qui permet d'obtenir des données au format XML, mais aussi HTML, ou encore texte simple à l'aide de requêtes HTTP.
- L'avantage principal est dans le côté asynchrone. La page entière ne doit plus être rechargée en totalité lorsqu'une partie doit changer ce qui entraîne un gain de temps et une meilleure interaction avec le serveur et donc le client.

HTML 5 vs XHTML



- Durant la première moitié des années 1990, l'évolution de HTML a dicté l'évolution du *World Wide Web*. Depuis 1997 et HTML 4, l'évolution de HTML a fortement ralenti ; 10 ans plus tard, HTML 4 reste utilisé dans les pages web. En 2008, la spécification du HTML 5 est à l'étude. La dernière spécification de HTML est la 4.01 (décembre 1999). Elle n'apporte que des corrections mineures à la version 4.0. Le développement de HTML est officiellement abandonné au profit de XHTML.
- En mars 2007, tirant la conséquence des réticences d'une partie de l'industrie et des concepteurs de contenus web face à XHTML 2.0, le W3C relance le développement de HTML.
- Une nouvelle spécification (*working draft*) HTML 5 est publiée en 2008 : <http://www.w3.org/TR/html5/>

Bonnes pratiques (I)



- Principes à suivre pour un site web de qualité (Ergonomie, Compatibilité, Accessibilité) :
<http://www.alsacreations.com/article/lire/544-Principes-a-suivre-pour-un-site-web-de->
- Pour être de qualité, un site doit être à la fois :
 - Utile (répondre à un besoin) et
 - Utilisable (par tous).
- Les bonnes pratiques web : <http://www.opquast.org/>
 - Respect des standards Web,
 - Architecture modulaire (séparation contenu/présentation/fonctionnement),

Bonnes pratiques (I)



- L'utilisateur moyen n'existe pas
 - Ciblez, structurez, prototypez, validez, testez, modifiez, (mesurez), validez, testez, modifiez,...
- Outils de validation (<http://www.w3.org/QA/Tools/#validators>) :
 - ◆ HTML : <http://validator.w3.org/>
 - ◆ CSS : <http://jigsaw.w3.org/css-validator/>
 - ◆ Lien : <http://validator.w3.org/checklink>
 - ◆ Correction, conversion de code : <http://www.w3.org/People/Raggett/tidy/>

L'enfer, c'est les autres : connaissez vos ennemis !



« Plus gros le logo ! »

« Ma femme n'aime pas le rouge »

« Mon fils a dit que c'était nul. »

Le patron

« Mettez une vidéo du patron sur la page d'accueil »

« Il y a un bouton vers le rapport annuel ? »

Le directeur communication

« On peut externaliser la production de contenus en Inde ou en Chine ? »

Le directeur financier

"Il faudrait leur demander de remplir ce petit formulaire (50 questions) avant d'aller plus loin"

Le service marketing

"Non"

"Impossible"

"Pas le temps"

Le service informatique