

# Cours Web - HTML5

Thierry Vaira

BTS SNIR Avignon

tvaira@free.fr © v0.2



# Sommaire

- 1 Introduction
- 2 Web
- 3 HTML
- 4 Feuille de style
- 5 Frameworks
- 6 Divers
- 7 Bonnes pratiques
- 8 Conclusion
- 9 Historique

# Objectifs

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



# Définition

- **HTML** signifie *HyperText Markup Language*.
- C'est un **langage de description de document** conçu pour représenter des **documents hypertextuels** (notion de liens), appelés aujourd'hui « pages web ».
- Il permet également de structurer sémantiquement et de mettre en forme le contenu des pages en utilisant des **balises** (*Markup Language*).
- Il est initialement dérivé du **SGML** (*Standard Generalized Markup Language*), jugé trop complexe.

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

# Technologies

L'utilisation conjointe d'un ensemble de technologies permet la réalisation de « documents web » :

- Document **HTML** (ou XHTML) pour la structure sémantique des informations ;
- Feuille de style **CSS** (*Cascading Style Sheets*) pour la présentation des informations ;
- Interface **DOM** (*Document Object Model*) et langage de programmation **JavaScript** pour afficher et interagir dynamiquement avec l'information présentée ;

Remarque : l'ensemble de ces technologies sont interprétés côté client (par le navigateur par exemple).

# Le World Wide Web

*« 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**

- Le web est un des services disponibles sur Internet.
- Internet est un **réseau de services**.
- Internet est basée sur l'**architecture client/serveur** :
  - Les services Internet sont tous fournis par des **serveurs**.
  - Les demandeurs du service sont nommés les **clients**.
  - Les **clients** émettent des **requêtes** en se basant sur un **protocole**.
  - Après traitement des requêtes, les serveurs renvoient des **réponses** en se basant sur un protocole.
- Chaque service est associé à un protocole (**web = HTTP**).



# Internet (I)

- Internet est un **réseau de réseaux**.
  - L'interconnexion des réseaux sur Internet est réalisé par des **routeurs** qui donnent une topologie de type **maillé**.
  - Internet est un réseau de type **WAN** (*Wide Area Network*).
  - Le plus souvent, on passe par un **FAI** (Fournisseur d'Accès Internet) ou **ISP** (*Internet Service Provider*) pour se connecter (raccorder) au réseau Internet.
- Les **aspects techniques** d'Internet sont couverts par des **RFC** (*Requests For Comment*)
  - Ce sont une série numérotée de documents électroniques destinées aux techniciens, ingénieurs, ...
  - Peu de RFC sont des standards, mais tous les standards d'Internet sont des RFC.
  - Les RFC sont rédigées pas des experts techniques. En mai 2008, le nombre de RFC a atteint les 5 000.
  - La première RFC (RFC 1), titrée "Logiciel hôte", a été publiée le 7 avril 1969 par Steve Crocker.

# Internet (II)

- En résumé, **Internet est un réseau informatique mondial qui rend accessibles au public des services** (comme le courrier électronique et le World Wide Web).
- Ses utilisateurs sont désignés par le néologisme « internaute ».
- Techniquement, Internet se définit comme le réseau public mondial utilisant le protocole de communication « **TCP/IP** » (signifiant : les protocoles de la famille TCP/IP).
- Lorsque les technologies Internet (TCP/IP, services, etc.) sont mises en oeuvre au sein de **réseaux privés** (entreprises, administrations, etc ...), on parle alors d'**Intranet**.

Attention, beaucoup d'utilisateurs confondent Internet et Web !



# Historique (I)

« *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** Historique du web en quelques dates :

- **1991 : Origine**, le *World Wide Web* est basé sur trois inventions, le **protocole de communication client/serveur HTTP** (*Hypertext Transfer Protocol*), les **adresses web** (*URI/URL*) et le **langage HTML** (*HyperText Markup Language*).
- **1993-1997 : Evolution du langage HTML par l'apports des navigateurs** (NCSA **Mosaic**, Netscape Navigator, Microsoft Internet Explorer, Mozilla Firefox, ...).
- **1994 : Naissance du W3C** (*World Wide Web Consortium*) qui est un organisme de normalisation chargé de promouvoir la compatibilité des technologies du World Wide Web telles que HTML, XHTML, XML, CSS, PNG, SVG et SOAP. Le W3C n'émet pas des normes au sens européen, mais des recommandations à valeur de standards industriels. Tim Berners-Lee préside aujourd'hui le W3C.



# Historique (II)

- **1996 : Standardisation HTTP/1.0** (RFC 1945), cette version supporte les serveurs HTTP virtuels, la gestion de cache et l'identification.
- **1997 : Standardisation HTTP/1.1** (RFC 2068 puis dans la RFC 2616 en juin 1999), cette version ajoute le support des connexions persistantes et la négociation de type de contenu (format de données, langue).
- **1999 : dernière spécification de HTML, la 4.01.**
- **2014 : finalisation prévue de la spécification HTML5.** Le W3C encourage les développeurs Web à utiliser HTML 5 dès maintenant.
- **2016 : la version 5.1** est officiellement publiée
- **2017 : la version 5.2** est officiellement publiée



# Créer des documents, un site (I)

- 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 (*File Transfer Protocol*).



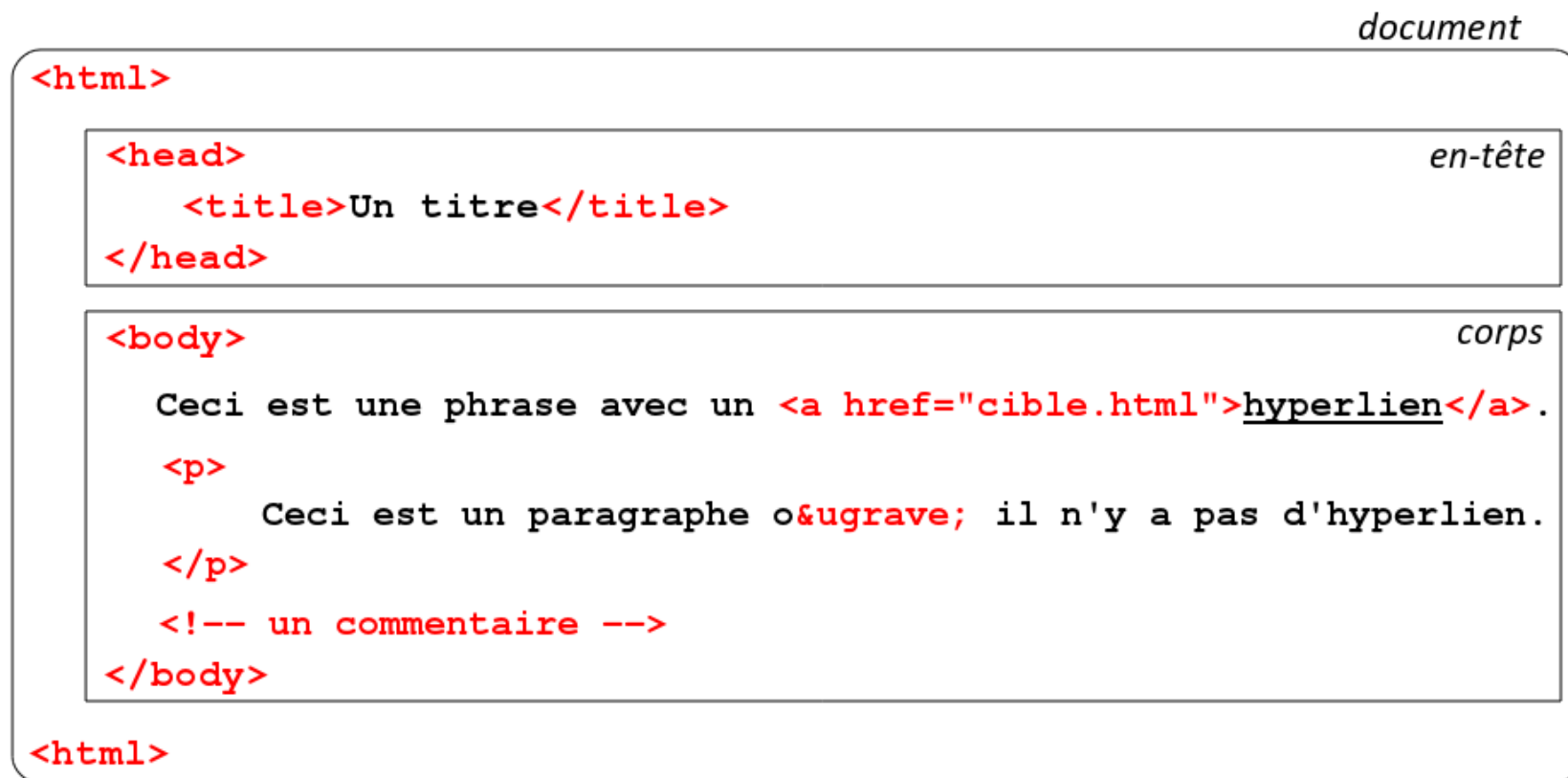
# Créer des documents, un site (II)

## Avertissements :

- Techniques : sélectionnez un encodage et déclarez-le, 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 , ...

# 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 initiale, la terminologie propre à SGML est utilisée : **document**, **élément**, **attribut**, **valeur**, **balise**, **entité**, **validité**, etc...



# Version d'un document HTML : instruction DOCTYPE

- Un **doctype** (« type de document ») est une instruction au début des documents SGML et XML (et donc HTML) spécifiant sa DTD.
- La déclaration de type de document indique la **définition de type de document (DTD)** en vigueur pour le document.
- Un **document HTML valide** doit donc **déclarer la version HTML** qu'il utilise :

- Pour HTML 5, il faut préciser :

```
<!DOCTYPE html>
```

- HTML 4.01 spécifie trois DTD (**Strict**, Transitoire et Frameset) et les DTD varient selon les éléments qui y sont gérés. Par exemple :

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



# DTD (I)

- La *Document Type Definition* (DTD), ou Définition de Type de Document, est **un document permettant de décrire un modèle de document SGML, XML ou HTML**.
- 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*).

# DTD (II)

## Exemple de définition de la DTD HTML 4.01

```
<!ELEMENT HTML O O (HEAD,BODY) -- élément racine du document -->
```

Le modèle de contenu d'un élément est spécifié selon la syntaxe suivante :

O : balise optionnelle et - : balise obligatoire

( ... ) : Délimite un groupe.

A : A doit être présent, une fois seulement.

A+ : A doit être présent une ou plusieurs fois.

A? : A doit être présent zéro ou une fois.

A\* : A peut être présent zéro ou plusieurs fois.

+(A) : A peut être présent et -(A) : A ne peut pas être présent.

A | B : A ou B peuvent l'un ou l'autre être présents, mais pas les deux.

A , B : A et B doivent tous deux être présents, dans cet ordre.

A & B : A et B doivent tous deux être présents, dans n'importe quel ordre.





# Encodage

Le **processus de détection de l'encodage** a également été modifié et s'effectue dans l'ordre :

- la vérification de la présence d'**un en-tête HTTP *Content-Type*** :

```
// version courte en HTML5
```

```
<meta charset="utf-8">
```

```
// ou version longue
```

```
<meta http-equiv="Content-Type" content="text/html; charset=
    UTF-8">
```

- ensuite la détection du **BOM** (*Byte Order Mark*) en début de fichier, qui indique l'utilisation d'un encodage unicode ainsi que l'ordre des octets. En UTF-8, l'indicateur d'ordre des octets est codé par la séquence EF BB BF.

👉 Le choix de l'UTF-8 est désormais préconisé par le W3C.



# Éléments

- Le langage HTML utilise d'**éléments** (ou balise) pour **structurer et décrire un document**.

`<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 HTML5 en anglais :  
<https://html.spec.whatwg.org/multipage/>
  - La spécification HTML 4.01 en français :  
<http://www.la-grange.net/w3c/html4.01/>
  - L'ensemble des spécifications produites par le W3C : [www.w3.org/TR/](http://www.w3.org/TR/)

# Éléments supprimés

La version HTML5 a supprimé notamment les balises suivantes :

- `basefont`, `big`, `center`, `font`, `strike`, `tt` et `u` car leurs effets étaient purement représentatifs ce qui est le rôle de CSS.
- `frame`, `frameset` et `noframes` pour des problèmes d'accessibilité et d'utilisation
- `acronym`, `isindex` et `dir` (utilisé alors `ul`)
- `applet` est remplacé par `object`

# Éléments nouveaux

La version HTML5 a ajouté des nouvelles balises, notamment celles-ci pour la structure d'un document :

- `main` : définit le contenu principal de la page, il doit être unique dans la page.
- `section` : définit les sections du document, telles que les chapitres, en-têtes, pieds de page
- `article` : partie indépendante d'un site, comme un commentaire
- `header` : spécifie un en-tête (comme une introduction ou des éléments de navigation)
- `footer` : définit le pied de page d'un article ou un document
- `nav` : définit une section dans la navigation.

D'autres nouvelles balises importantes : `figure` (et `figcaption`), `audio`, `video`, `embed`, `time`, `canvas`, ...



# Attributs

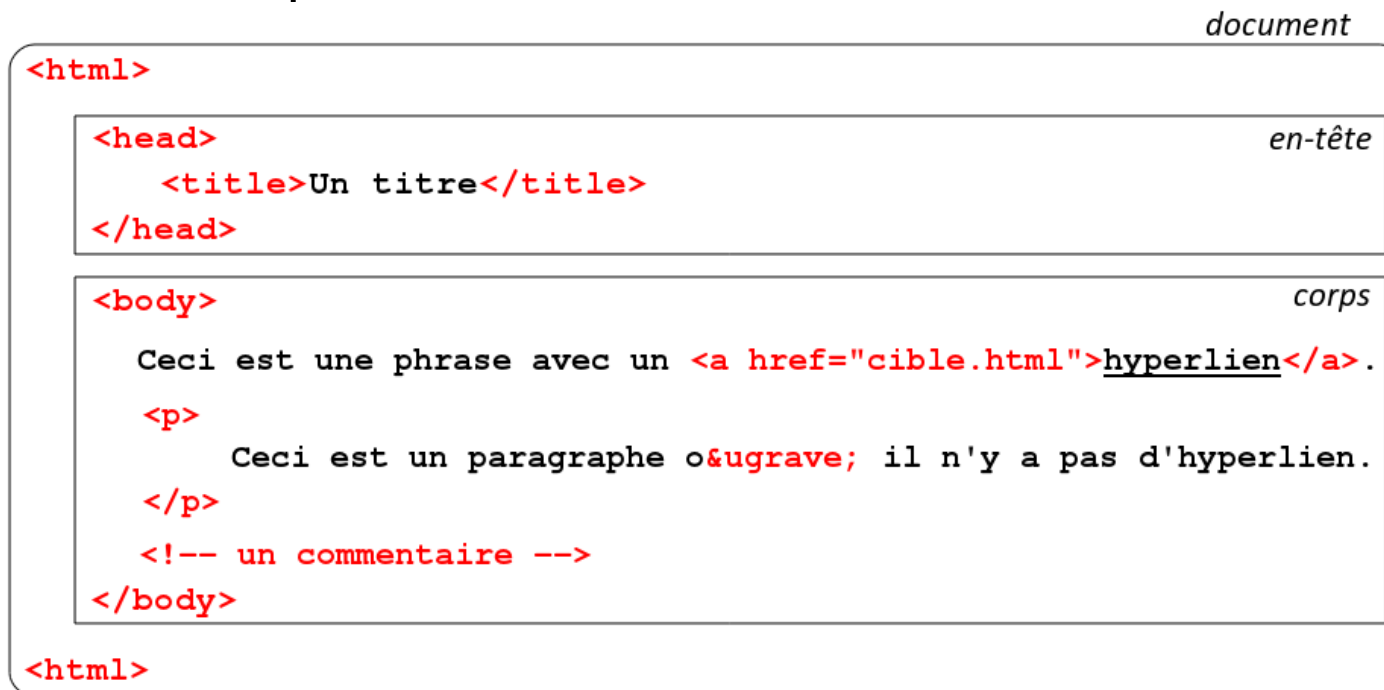
- Les attributs permettent de **préciser les propriétés des éléments HTML**.

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

- Il y avait par exemple **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).
- Spécifications :
  - La spécification HTML5 en anglais (de nombreux attributs ont été ajoutés) : <https://html.spec.whatwg.org/multipage/>
  - La spécification des attributs 4.01 en français : <http://www.w3.org/TR/html401/index/attributes.html>

# Un document

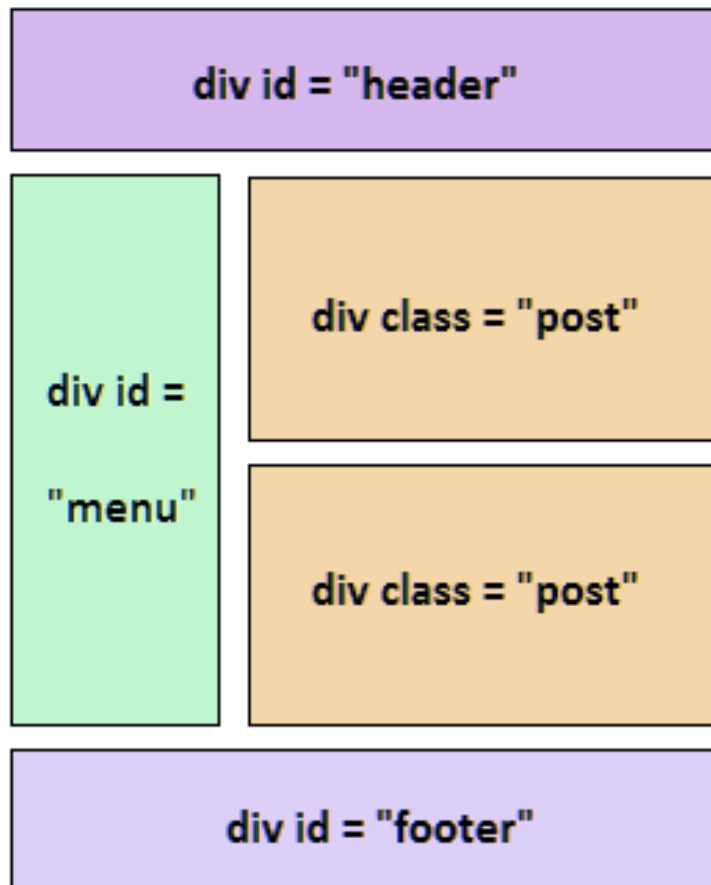
- Un document HTML est défini par une **balise racine** nommée HTML. Cette balise accepte l'attribut lang qui précise la langue utilisée pour le contenu de la page : `<html lang="fr">`
- Le document comprend deux parties : un **en-tête** (pour les métadonnées sur le document) délimité par HEAD et le **corps** du document défini par BODY.



# HTML4 vs HTML5

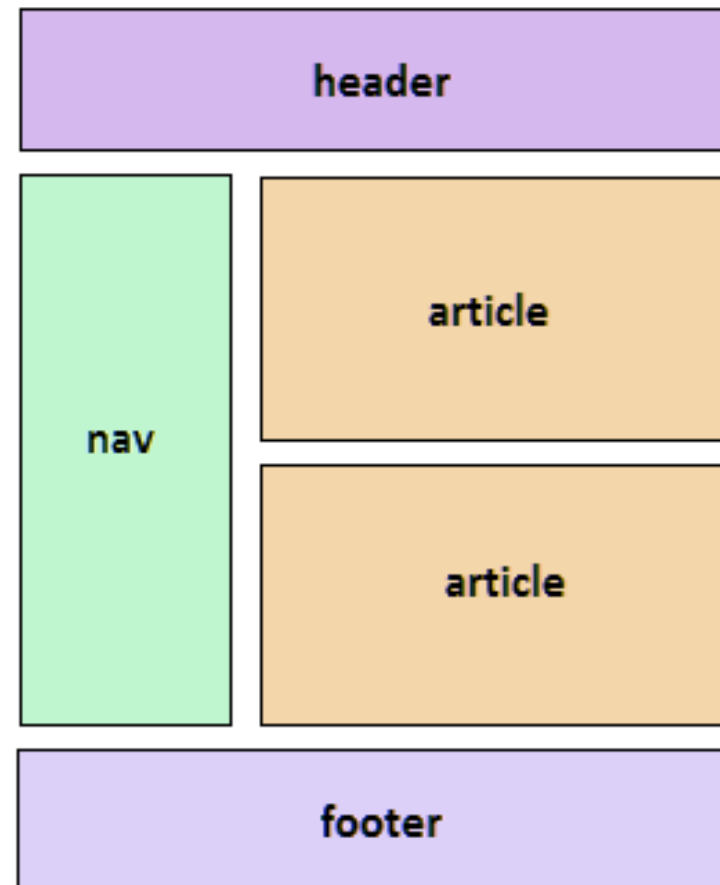
- Lien : <https://www.w3.org/TR/html5-diff/>

## HTML4



**VS**

## HTML5



# 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 et 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 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 "UTF-8" :

```
// HTML5
```

```
<meta charset="utf-8">
```

```
// ou :
```

```
<META http-equiv="Content-Type" content="text/html; charset=UTF-8">
```



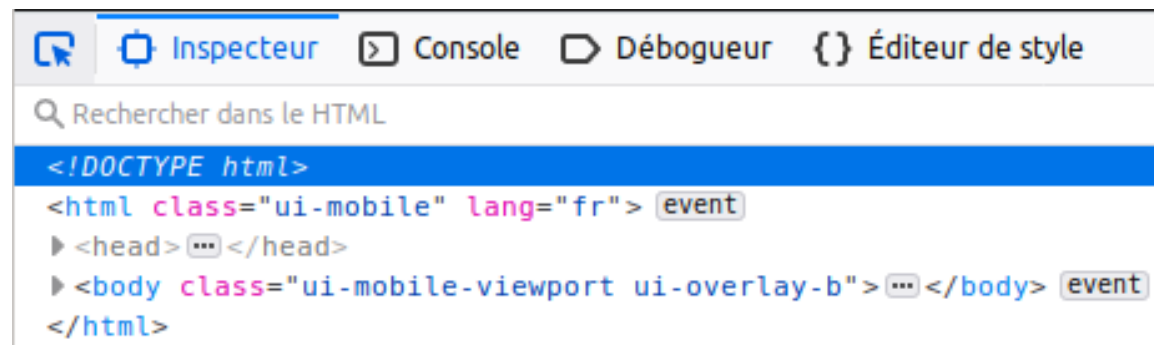
# Le corps du document

- Le corps du document (élément 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 proscrits.

Le W3C recommande aux auteurs et aux développeurs de séparer la structure et la présentation.

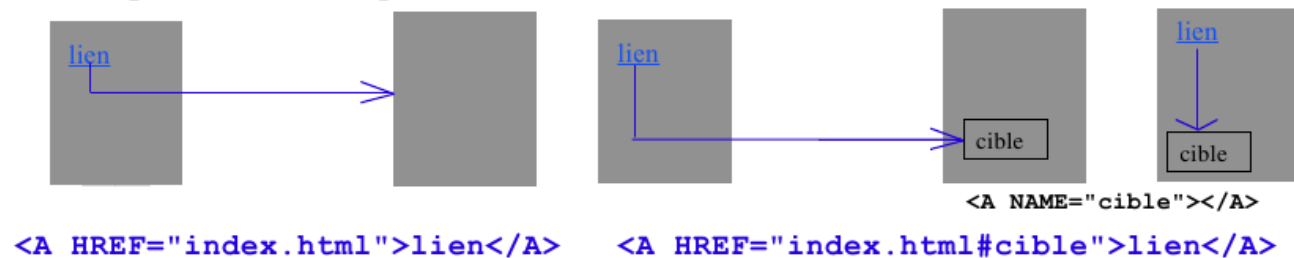
# DOM (*Document Object Model*)

- 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.
- Le 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 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/>
- On peut « voir » le DOM d'un document avec les outils de développement web inclus dans les navigateurs modernes :



# 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.
- 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://192.168.1.1:8080/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) : les éléments en-ligne ne peuvent contenir que des données et d'autres éléments en-ligne.
  - ou de **bloc** (DIV) : les éléments de bloc créent des structures « plus grandes » : ils peuvent contenir des éléments en-ligne et d'autres éléments de bloc.
- Ces éléments n'imposent aucune présentation sur le contenu et sont notamment très utilisés conjointement avec les **feuilles de style** (CSS).

# Feuille de style (I)

- Les feuilles de style CSS (*Cascading Style Sheets*) servent **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
- Spécifications CSS (en français) :

<http://www.yoyodesign.org/doc/w3c/css1/index.html>

<http://www.yoyodesign.org/doc/w3c/css2/cover.htm>

<https://www.w3.org/Style/CSS/Overview.fr.html>



# Feuille 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 :
  - simple (ex. 'H1') ou
  - contextuel (fait de plusieurs sélecteurs simples, p.ex. 'H1 B').
- Liste des propriétés et des valeurs pour CSS1 :

<http://www.yoyodesign.org/doc/w3c/css1/index.html#css1-properties>



# Feuille de style (III)

- 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 '#'

```
#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 */
```

```
#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*/
```



# Feuille 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 regroupe quatre façons de combiner le style CSS à un document HTML :

```
<HTML>
```

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

```
    <LINK REL=STYLESHEET TYPE="text/css" HREF="cool.css" TITLE="Cool"> <!--  
      1 -->
```

```
    <STYLE TYPE="text/css"> <!-- 2 -->
```

```
      @import url(http://style.com/basic.css); /* 3 */
```

```
      H1 { color: blue }
```

```
    </STYLE>
```

```
  </HEAD>
```

```
  <BODY>
```

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

```
    <P STYLE="color: green">et le paragraphe est en vert.</P> <!-- 4 -->
```

```
  </BODY>
```

```
</HTML>
```



# Quelques frameworks

- **Bootstrap** est une collection d'outils utiles à la création du *design* (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs. <https://getbootstrap.com/>
- **Foundation** est une collection d'outils destiné au développement web frontal de sites internet et d'applications web.  
<https://foundation.zurb.com/>
- **Blueprint** est un *framework* CSS destiné à réduire le temps de développement et assurer la compatibilité sur de nombreux navigateurs, lors de l'utilisation de feuilles de style en cascade (CSS).  
<http://www.blueprintcss.org/>
- **jQuery Mobile** est un *framework* d'interface optimisé pour les appareils mobiles tactiles. Le Framework jQuery Mobile est compatible avec d'autres *frameworks* mobiles tels que PhoneGap.  
<http://jquerymobile.com/>

# XML

- XML (*eXtensible Markup Language*) est standardisé par la spécification W3C :  
<http://www.yoyodesign.org/doc/w3c/xml11/>
- 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 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, ...



# Exemple XML

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>
<menu id="file" value="File">
  <popup>
    <menuitem value="New" onclick="CreateNewDoc()"/>
    <menuitem value="Open" onclick="OpenDoc()"/>
    <menuitem value="Close" onclick="CloseDoc()"/>
  </popup>
</menu>
```



# XHTML

- XHTML (*Extensible HyperText Markup Language*) est un langage de balisage servant à écrire des pages pour Web.
- Un document XHTML est un document XML contenant de l'HTML
- Spécification XHTML 1.0 :  
<http://www.la-grange.net/w3c/xhtml11/>
- XHTML a été conçu à l'origine pour être le successeur de HTML
- Finalement le W3C a officiellement annoncé l'abandon du développement du XHTML 2 afin de se consacrer entièrement à HTML5.
- Un document XHTML5 est un document XML contenant de l'HTML5

# DHTML

- DHTML (*Dynamic HTML*) 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. 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 DOM (*Document Object Model*).
- 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.



# JavaScript

- **JavaScript** (qui est souvent abrégé en « JS ») est un langage de script léger, orienté objet, principalement connu comme le langage de script des pages web.
- Le standard pour JavaScript est **ECMAScript**. JavaScript ne doit pas être confondu avec le langage de programmation Java.
- Le code JavaScript a besoin d'un objet global pour y rattacher les déclarations (variables et fonctions) avant d'exécuter des instructions. La situation la plus connue est celle de l'objet `window` obtenu dans le contexte d'une page web.
- Du code JavaScript peut être intégré directement au sein des pages web (avec les balises `<script></script>`, pour y être exécuté sur le poste client. C'est alors le navigateur web qui prend en charge l'exécution de ces programmes appelés scripts.
- JavaScript peut également être utilisé comme langage de programmation sur un serveur HTTP. On pourra citer Node.js, une plateforme polyvalente de développement d'applications réseau.

# 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.

# Ajax

- **AJAX** est un acronyme signifiant *Asynchronous JavaScript and XML* 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/JSON pour le format des données informatives et visuelles

# JSON

- **JSON** (*JavaScript Object Notation*) est un format de données textuelles dérivé de la notation des objets du langage JavaScript.
- Il permet de représenter de l'information structurée comme le permet XML par exemple.
- Un document JSON comprend deux types d'éléments structurels : des ensembles de paires « nom » (alias « clé ») / « valeur » et des listes ordonnées de valeurs.
- Ces mêmes éléments représentent trois types de données : des objets, des tableaux et des valeurs génériques de type tableau, objet, booléen, nombre, chaîne de caractères ou null.
- Il est notamment utilisé comme langage de transport de données par AJAX et les services Web. D'autres solutions sont possibles comme XML.

# Exemple JSON

```
{
  "menu": {
    "id": "file",
    "value": "File",
    "popup": {
      "menuitem": [
        { "value": "New", "onclick": "CreateNewDoc()" },
        { "value": "Open", "onclick": "OpenDoc()" },
        { "value": "Close", "onclick": "CloseDoc()" }
      ]
    }
  }
}
```

# jQuery

- **jQuery** est une bibliothèque JavaScript libre et multiplateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web.
- Le but de la bibliothèque étant le parcours et la modification du DOM
- Elle contient de nombreuses fonctionnalités ; notamment des animations, la manipulation des feuilles de style en cascade (accessibilité des classes et attributs), la gestion des évènements, etc.
- L'utilisation d'Ajax est facilitée et de nombreux *plugins* sont présents.
- jQuery a connu un large succès auprès des développeurs Web et son apprentissage est aujourd'hui un des fondamentaux de la formation aux technologies du Web. Il est à l'heure actuelle la bibliothèque *front-end* la plus utilisée au monde (plus de la moitié des sites Internet en ligne intègrent jQuery).
- Émergence de nouvelles librairies : React (JavaScript) et Vue.js





# 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-qualite-Ergonomie-Compatibilite-Accessibilite.html>

- Pour être de qualité, un site doit être à la fois :
  - Utile (répondre à un besoin) et
  - Utilisable (par tous).
- Les bonnes pratiques web :
  - Respect des standards Web,
  - Architecture modulaire (séparation contenu/présentation/fonctionnement).
  - *Responsive web design* ?

<https://checklists.opquast.com/fr/>



# Bonnes pratiques (II)

- Démarche générale : Ciblez, structurez, prototypez, validez, testez, modifiez, (mesurez), validez, testez, modifiez, ...
- Méthodologie TP :
  - Ciblez, structurez, prototypez avec un **éditeur de texte**
  - validez avec un **outil de validation**
  - testez avec un **navigateur**
- 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/`

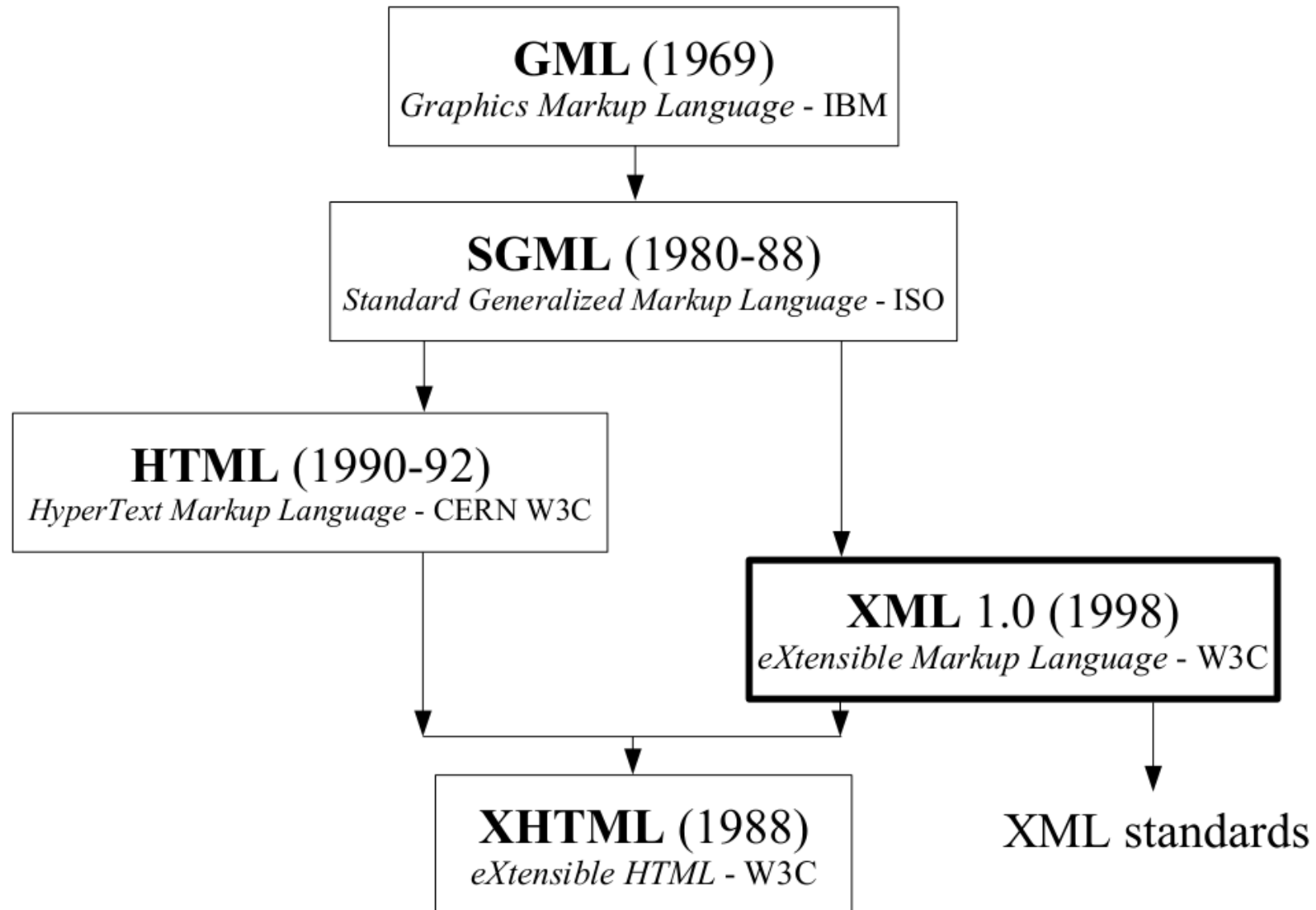


# L'enfer, c'est les autres : connaissez vos ennemis !

- Le patron :
  - « *Plus gros le logo !* »
  - « *Ma femme n'aime pas le rouge* »
  - « *Mon fils a dit que c'était nul.* »
- Le directeur communication :
  - « *Mettez une vidéo du patron sur la page d'accueil* »
  - « *Il y a un bouton vers le rapport annuel ?* »
- Le service marketing :
  - « *Il faudrait leur demander de remplir ce petit formulaire (50 questions) avant d'aller plus loin* »
- Le directeur financier :
  - « *On peut externaliser la production de contenus en Inde ou en Chine ?* »
- Le service informatique :
  - « *Non* »
  - « *Impossible* »
  - « *Pas le temps* »



# Arbre généalogique



# Historique (1/2)

En six dates :

- 1991 : 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 : Avec le navigateur NCSA Mosaic, 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 : Avec l'apparition de Netscape Navigator, 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.



# Historique (2/2)

- 1997 : 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**
- 2014 : finalisation de la spécification HTML5. De nombreux éléments et attributs sont ajoutés et le langage comprend 8 nouvelles API (dessin 2D, audio/vidéo, drag & drop, ...). Le W3C encourage les développeurs Web à utiliser HTML 5 dès maintenant.