

Cours Web - HTML (1° partie)

Thierry Vaira

BTS IRIS Avignon

tvaira@free.fr © v0.1



Objectifs

- Se familiariser avec le codage 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 (cf. deuxième partie) ;

Remarque : l'ensemble de ces technologies sont interprétés côté client.

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.

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*).

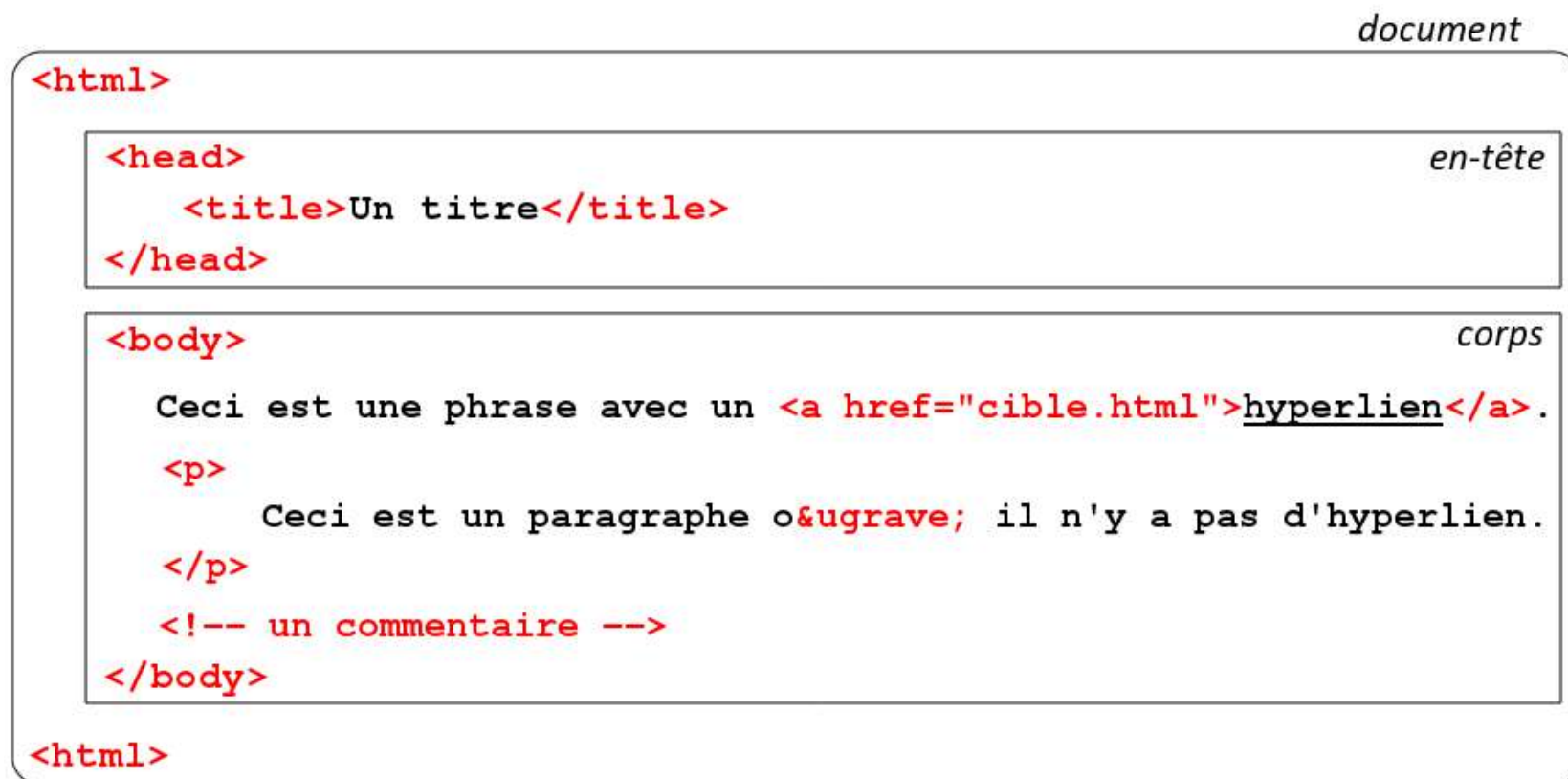
Créer des documents, un site (II)

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 (I)

- Un **document HTML valide** doit **déclarer la version HTML** qu'il utilise. On utilisera la dernière version spécifiée à ce jour, la **4.01**.
- 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 (**Strict**, Transitoire et Frameset) et les DTD varient selon les éléments qui y sont gérés.

Version d'un document HTML (II)

- Pour HTML 4.01, les auteurs doivent donc inclure l'une des déclarations de type de document suivantes dans leurs documents (avant la balise `<html>`) :

- **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">
```

- Frameset (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 (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.

Éléments

- La version 4 de HTML décrit **91 éléments** (ou balise). Ils permettent de **structurer 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 HTML 4.01 en français :

<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).
- La spécification des attributs 4.01 en français :

<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 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" :

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

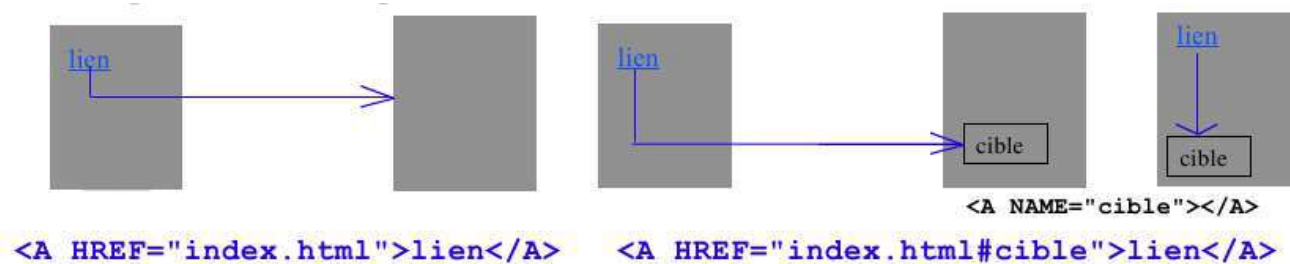
Le corps du document

- 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.

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 CSS1 et CSS2 en français :

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

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

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>

```

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).

<http://www.opquast.org/>

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/`

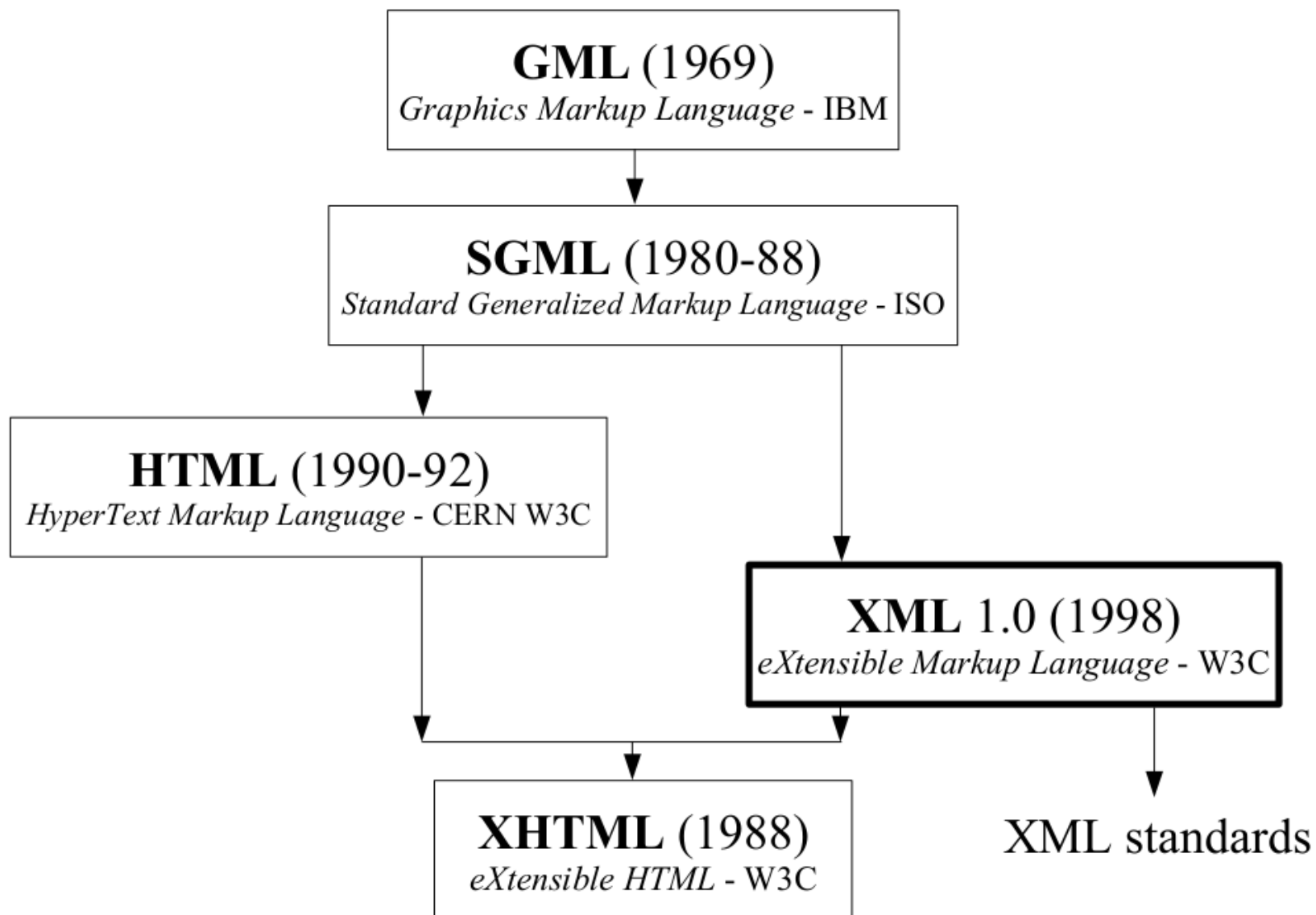
Lien : `http://validator.w3.org/checklink`

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 prévue 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.