

Publication en ligne

Master “Langues et commerce électronique”

dominique.dhoutaut@pu-pm.univ-fcomte.fr

Objectifs du module

- Concept de document composite et de navigation. Concept de client / serveur.
- Le besoin de normaliser. HTML (Hyper Text Markup Language), un langage de présentation des données et plus encore.
- CSS (Cascading Style Sheets) : La séparation du contenu et du contenant. La présentation d'une même information peut être modifiée très facilement.
- Un premier pas vers l'interactivité avec l'utilisateur : javascript

Les concepts

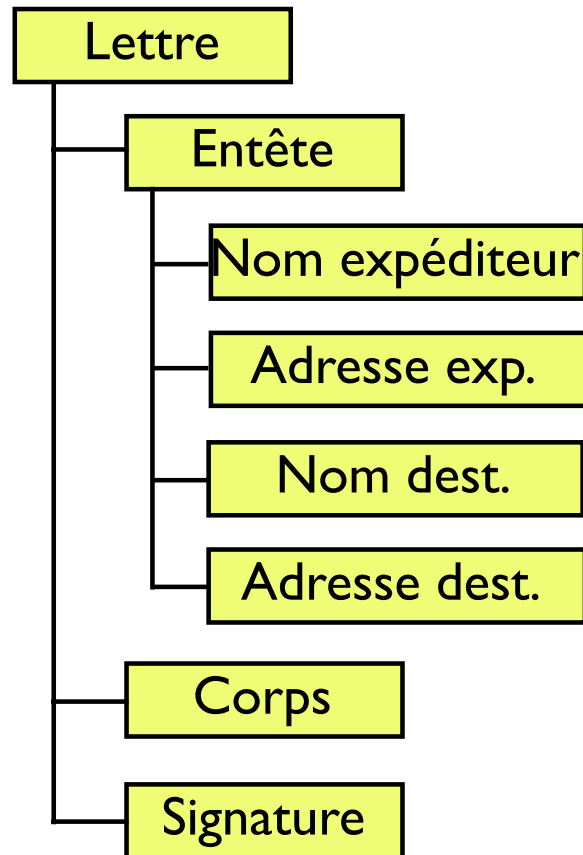
L'hypertexte : la navigation

- HTML provient à l'origine de chercheurs du CERN qui avaient besoin d'organiser des masses importantes de documents. Le World Wide Web Consortium (collège d'organismes de recherche, d'universités, d'entreprises) publie les normes du web (HTML4 puis XHTML et maintenant HTML5, entre autres)
- le “HT” de HTML signifie HyperText
- Notion de référence (par exemple, “pour le détail de tel procédé, aller voir le chapitre XX du livre YY”)
- Des éléments (mots clefs, images, ...) du document sont donc des pointeurs vers d'autres ressources, et l'on peut “naviguer” de l'une à l'autre.

Un document : de l'information structurée

- La très grande majorité des documents possèdent une certaine structure. Cette structure est en elle même porteuse d'information. La signification d'un mot peut changer en fonction de l'endroit où il se trouve. Quelques exemples :
- Un livre possède un titre, une table des matières, des chapitres, des paragraphes, des phrases
- Une lettre possède une entête, un corps et une signature. L'entête elle-même contient un nom et une adresse d'expéditeur ainsi que de destinataire.
- Une note de restaurant porte une adresse, un certain nombre de lignes de consommations et une somme

Un document : de l'information structurée



- La structure du document peut être représentée sous la forme d'un arbre.
- Le principe est de définir des éléments, et de dire de quoi ils peuvent être composés.
- Des langages existent pour décrire de telles structures. HTML est l'un d'eux.

HTML : décrire un document et sa structure dans un fichier texte

- Le HTML en lui même est un format de fichier texte pour décrire les documents. Le fait que ce ne soit que du texte facilite sa portabilité et son transfert.
- HTML utilise des balises (Markup Language) pour décrire le début et la fin des sections du document. Ses sections peuvent s'imbriquer. Un exemple simple :

```
<html>
  <head>
    <title>Titre de ma page</title>
  </head>
  <body>
    <p>Ceci est mon premier paragraphe</p>
    <p>Ceci est mon deuxième paragraphe</p>
    
  </body>
</html>
```

Le rôle du navigateur

- Le navigateur est le logiciel qui lit le fichier HTML brut et l'affiche sous une forme agréable à l'œil.
- Il interprète les balises HTML.
 - Ce qui est entre `<title>` et `</title>` sera affiché dans la barre de titre du navigateur
 - Les paragraphes seront séparés par une ligne vide et leur première ligne sera indentée.
 - L'image dont le nom est spécifié sera affichée
 - etc.

Le rôle du navigateur

- Il existe un grand nombre de balises HTML qui permettent
 - De spécifier la structure logique d'un texte (titre, sections, sous sections, paragraphes, tables). Le navigateur peut présenter les données en conséquence (un titre de section est en général écrit plus gros qu'un titre de sous-section, ...)
 - D'altérer directement la présentation (gras, italique, taille des caractères, couleur). Attention cependant, ces balises sont maintenant avantageusement remplacées par les CSS (qui seront abordées plus tard).
 - D'insérer des images
 - D'insérer des liens

Le rôle du navigateur

- Attention : les navigateurs n'ont pas nécessairement les mêmes capacités et une certaine latitude leur est laissée quand à l'interprétation des balises.
- Il existe de nombreux navigateurs différents (le plus utilisé est Internet Explorer de Microsoft si l'on compte ses multiples versions, mais il y en a des dizaines d'autres. Mozilla, Safari, Opera, etc.)
- Certains navigateurs proposent des fonctionnalités (des balises) qu'ils sont les seuls à reconnaître.
- Utiliser des standards (tels que édictés par le W3C - World Wide Web Consortium) évite une bonne part des problèmes d'incompatibilité avec les extensions propriétaires.

Le rôle du navigateur

- Rappel IMPORTANT sur le rôle du HTML :
 - HTML est un langage qui permet avant tout de décrire le **CONTENU** et la **STRUCTURE** des documents.
 - Historiquement, une très grande latitude était accordée aux navigateurs quant-à la mise en page finale (d'un navigateur à l'autre l'apparence pouvait changer considérablement, ce qui n'était pas du goût de nombreux commerciaux et développeurs de sites). Des balises ont donc été ajoutées, qui permettaient un meilleur contrôle de l'apparence (police, taille, couleur, ...). Mais ce faisant, la décorrelation entre le contenu et la présentation a été en partie perdue.
 - Aujourd'hui, grâce aux CSS qui se chargent de la présentation, le HTML doit à nouveau être utilisé surtout pour le contenu et la structure.

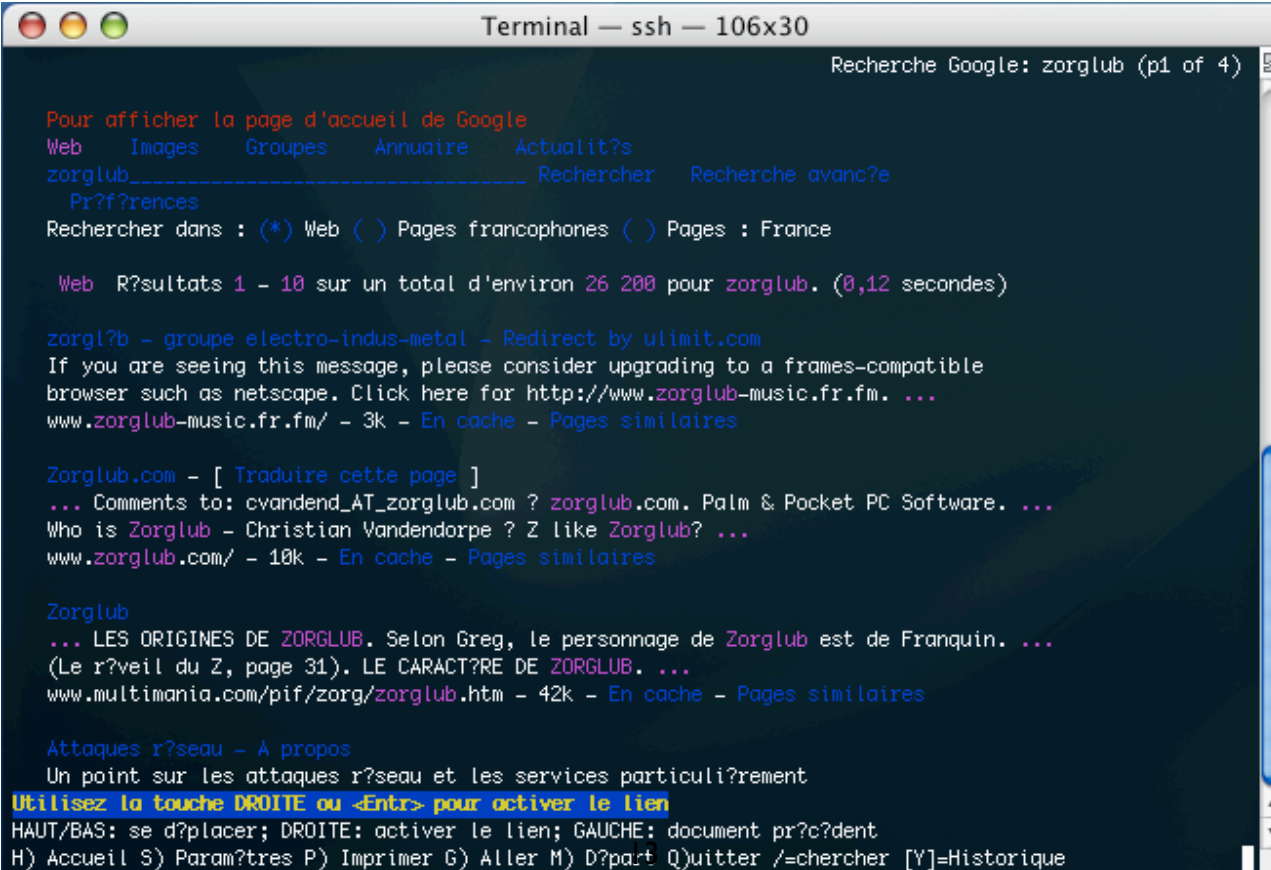
Le rôle du navigateur

- Rendu d'une page google avec safari :



Le rôle du navigateur

- Rendu de la même page google avec lynx :



```
Terminal — ssh — 106x30
Recherche Google: zorglub (p1 of 4)

Pour afficher la page d'accueil de Google
Web Images Groupes Annuaire Actualit?s
zorglub_____ Rechercher Recherche avanc?e
Pr?f?rences
Rechercher dans : (*) Web ( ) Pages francophones ( ) Pages : France

Web R?sultats 1 - 10 sur un total d'environ 26 200 pour zorglub. (0,12 secondes)

zorglub - groupe electro-indus-metal - Redirect by ulimit.com
If you are seeing this message, please consider upgrading to a frames-compatible
browser such as netscape. Click here for http://www.zorglub-music.fr.fm. ...
www.zorglub-music.fr.fm/ - 3k - En cache - Pages similaires

Zorglub.com - [ Traduire cette page ]
... Comments to: cvandend_AT_zorglub.com ? zorglub.com. Palm & Pocket PC Software. ...
Who is Zorglub - Christian Vandendorpe ? Z like Zorglub? ...
www.zorglub.com/ - 10k - En cache - Pages similaires

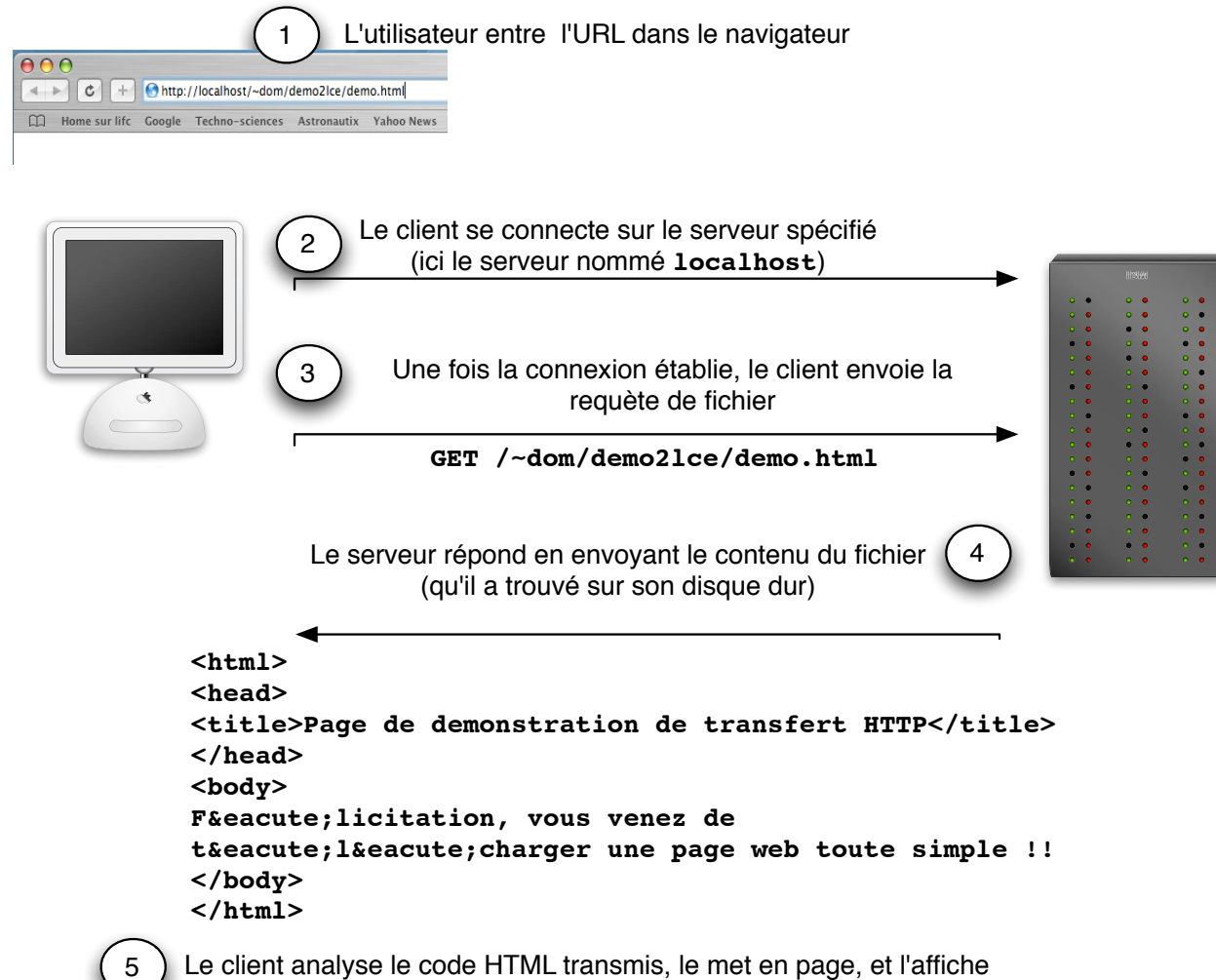
Zorglub
... LES ORIGINES DE ZORGLUB. Selon Greg, le personnage de Zorglub est de Franquin. ...
(Le r?veil du Z, page 31). LE CARACT?RE DE ZORGLUB. ...
www.multimania.com/pif/zorg/zorglub.htm - 42k - En cache - Pages similaires

Attaques r?seau - A propos
Un point sur les attaques r?seau et les services particuli?rement
Utilisez la touche DROITE ou <Entr> pour activer le lien
HAUT/BAS: se d?placer; DROITE: activer le lien; GAUCHE: document pr?c?dent
H) Accueil S) Param?tres P) Imprimer G) Aller M) D?part Q)uitter /=chercher [Y]=Historique
```

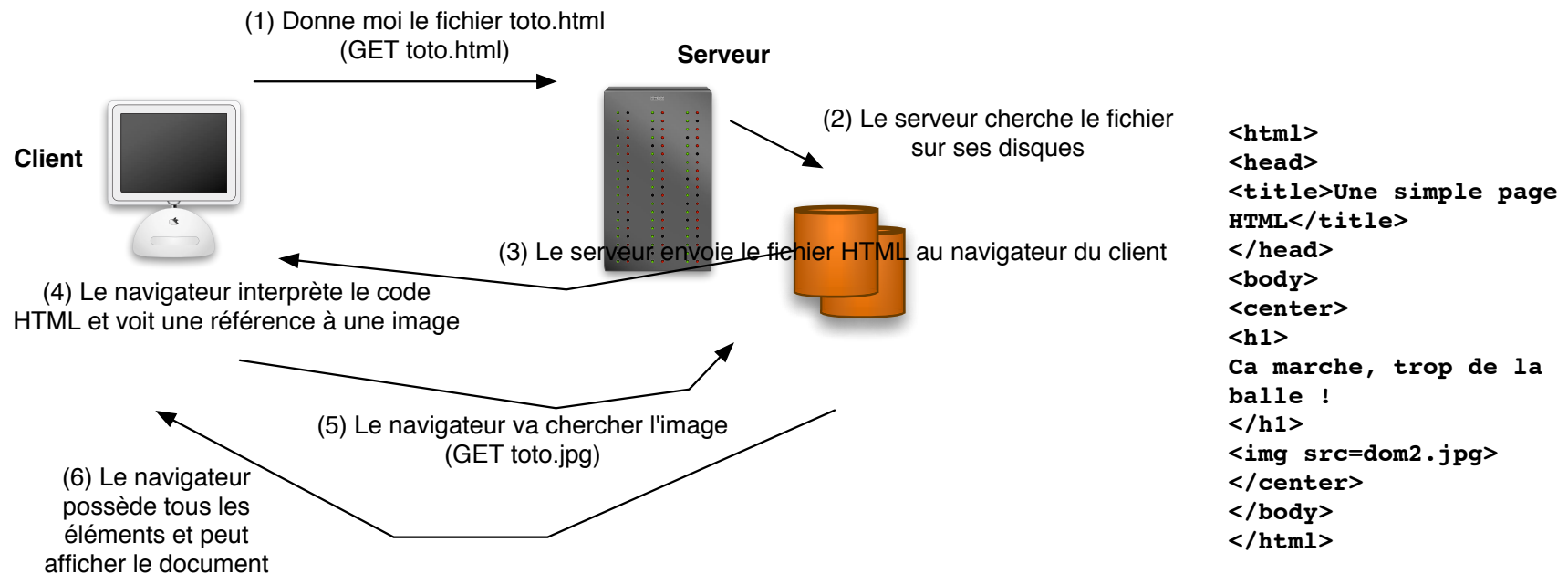
Où se trouvent les documents ? Concept de client / serveur

- Il faut pouvoir mettre les documents à la disposition du public.
- Les documents sont hébergés sur des ordinateurs appelés “serveurs”, reliés au réseau.
- Les documents possèdent une “adresse” nommée URL (Unified Resource Locator”) qui permet d’indiquer sur quel serveur et dans quelle partie de ce serveur trouver le document.
Exemple : <http://www.google.fr/index.html> (“le fichier index.html se trouvant sur le serveur www.google.fr”)
- Lorsque l’on donne une URL au navigateur, il se charge de contacter le serveur et de télécharger le fichier. Une fois le fichier copié localement il est analysé puis affiché.

Où se trouvent les documents ? Concept de client / serveur



Où se trouvent les documents ? Concept de client / serveur



- Le code HTML peut faire référence à d'autres objets situés sur le serveur (par exemple des images). Pour pouvoir afficher correctement le document, le navigateur va donc télécharger successivement le code HTML puis tous les objets auxquels il fait référence.

Où se trouvent les documents ? Concept de client / serveur

- Un standard définit la manière dont un navigateur doit demander les documents aux serveurs WEB, cela permet de rendre tous les navigateurs compatibles avec tous les serveurs. Le protocole standard d'échange entre le navigateur et le serveur WEB est le HTTP (HyperText Transfer Protocol)
- Les documents peuvent se trouver sur l'ordinateur client. Dans ce cas il est inutile de préciser le nom du serveur dans l'adresse.
- Dans ce module nous ne chercheront pas à déposer nos documents sur un serveur distant.

HTML

Les balises

html, xhtml et versions

- Depuis le début du web un certain nombre d'évolutions dans le langage HTML se sont succédées. Les versions utilisées actuellement sont
 - HTML 4.01 (plutôt remplacé par XHTML)
 - XHTML 1.0 et 1.1 (l'essentiel des pages web actuelles)
 - HTML 5 (pas complètement stabilisé, utilisé dans les nouveaux développements)
- Les versions successives ont introduit de nouvelles balises et rendus certaines autres obsolètes.
- HTML jusqu'à 4.01 est basé sur SGML
- XHTML et HTML 5 possèdent une structure plus régulière, issue de XML

Syntaxe générale

- HTML est un langage à balise, ce qui signifie que l'on parsemera le document de mot-clefs ayant une syntaxe bien particulière afin de donner des indications de structure et de présentation au navigateur.

- Les balises sont délimitées par les signes < et >. Un texte balisé aura cette apparence :

avant le texte <balise> Ceci est un texte entre balises
</balise> suite du texte

- Par exemple les balises et définissent la mise en gras du texte qu'elles encadrent.

texte maigre texte gras texte maigre

donne :

texte maigre **texte gras** texte maigre

Syntaxe générale

- Les commentaires : tout ce qui est compris entre les balises `<!--` et `-->` n'est pas interprété par les navigateurs

`<!-- ici je peux écrire n'importe quoi -->`

- Si l'on veut utiliser les symboles `<` ou `>` dans le document, il faudra utiliser un artifice pour éviter que le navigateur les interprète comme des débuts de balise. On utilise dans ce cas `>` (greater than) au lieu de `>` et `<` (lower than) au lieu de `<`.

Les signes `<` et `>` délimitent les `balises` html

Les signes `<` et `>` délimitent les **balises** html

- http://www.w3schools.com/html5/html5_reference.asp

Codage des caractères

- La norme de base impose l'utilisation exclusive de la table de caractère ASCII (American Standard Code for Information Interchange) dans le fichier HTML. Chaque caractère est codé sur 7 bits (soit seulement 127 caractères différents autorisés)
- Dans ces 127 caractères on ne trouve pas les caractères particuliers à chaque langue (les caractères accentués français, le ß allemand, etc.)
- Des codes spéciaux sont utilisés pour ces caractères et autres symboles, exemples :

´	é
`	è
ô	ô
ã	ã

Codage des caractères

- Il est cependant possible de spécifier l'encodage pour tout le document. Il est alors possible d'utiliser directement tous les caractères que vous pouvez entrer sur votre clavier Dans ce cas, deux conditions doivent être réunies :
 - Les caractères dans le fichier HTML doivent effectivement être écrits suivant le codage choisi. Cela doit être explicitement demandé lors de l'édition ou la sauvegarde de ce fichier. Consulter la documentation de votre éditeur de texte !
 - Une balise dans le fichier HTML doit indiquer au navigateur quel codage il doit utiliser pour la suite du document.
 - en HTML 4.01 ou XHTML : `<meta http-equiv="content-type" content="text/html; charset=UTF-8">`
 - en HTML 5 : `<meta charset="UTF-8">`

DOCTYPE

- Lorsqu'un navigateur analyse un document HTML, il lui faut un moyen de déterminer quel est son format (HTML 4.01 ? XHTML ? HTML 5 ? etc.). Pour cela on utilise la déclaration `<!DOCTYPE>`
- Ce n'est pas une balise à proprement parler. Il existe un certain nombre de variations, dont :

- En HTML 4.01

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

- En XHTML 1.0

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

- En HTML 5

```
<!DOCTYPE html>
```


Validation

- Spécifier le DOCTYPE ne suffit pas, il faut aussi s'assurer qu'on le respecte !
- Suivant le DOCTYPE choisi, certaines balises ou certaines syntaxes peuvent être correctes ou interdites.
- Pour vérifier que l'on n'a pas fait d'erreur, on peut utiliser un validateur, c'est à dire un outil qui va vérifier et pointer les erreurs le cas échéant
- Utiliser par exemple <http://validator.w3.org>

Structure générale

- `<html>` Tous les documents HTML commencent par la balise `<html>` et se terminent par `</html>`

```
<html>  
  corps du document  
  ...  
</html>
```

- `<head>` Le titre du document et les scripts javascript sont intégrés dans une zone de prologue ou d'entête entre `<head>` et `</head>`. Le reste est entre `<body>` et `</body>`

```
<html>  
  <head>  
    <title>Bienvenue sur la page de toto</title>  
  </head>  
  <body>  
    corps du document  
    ...  
  </body>  
</html>
```

Attributs des balises

- Certaines balises peuvent admettre des attributs (ou paramètres)
- Le séparateur est l'espace.
- La valeur de l'attribut doit être entre “ ”
 - Exemple : ``
- Dans le cas où la valeur de l'attribut contient des “, alors il faut utiliser des ‘
 - Exemple : ``

Les divisions

- On rencontre couramment une subdivision des articles longs en chapitres, sous chapitres, section, etc. Souvent on utilise des numérotations de type 1, 1.1, 1.2.3, etc ...
- En HTML, on dispose de 6 niveaux de subdivision, grâce à la balise <h> où n prend une valeur comprise entre 1 et 6
- C'est le navigateur qui choisit la présentation exacte, mais <h1> correspond toujours au titre le plus "gros" et <h6> au plus "petit"

```
<h1>Nom de la s&eacute;rie</h1>
```

```
<h2>El Hazard</h2>
```

```
Nombre d'&eacute;pisodes = 26
```

```
<h2>Excel saga</h2>
```

```
Nombre d'&eacute;pisodes = 26
```

Nom de la série

El Hazard

Nombre d'épisodes = 26

Excel saga

Nombre d'épisodes = 26

Les divisions

- `
`
 - Le navigateur traite et affiche le texte en continu.
 - Dans le code HTML, des espaces consécutifs seront assimilés à un seul, et les sauts de ligne sont ignorés.
 - Il appartient à l'auteur de la page de demander explicitement les sauts de ligne, grâce à la balise `
`

Et un,`
` et deux,`
` et trois,`<br/ >` et quatre

Et un,
et deux,
et trois,
et quatre

Les divisions

- `<p> ... </p>`
 - Définit un paragraphe (on saute une ligne et on va à la ligne par rapport au texte précédent).
- `<hr />`
 - Séparation horizontale. Dessine en général un trait qui divise en largeur la fenêtre du navigateur

Jinnai et les bugroms
<hr />
bon, parlons maintenant de
choses sérieuses

Jinnai et les bugroms

bon, parlons maintenant de choses sérieuses

Les divisions

- Attributs possibles pour `<hr />`
 - `<hr size=valeur />` : épaisseur du trait en pixels
 - `<hr width=30% />` `<hr width=100 />` : largeur du trait en pour-cent relatifs à la fenêtre, où en pixels.
 - `<hr align=left />` : si la largeur est inférieure à 100%, indique de quel côté mettre le trait (left, center, right)
 - `<hr noshade />` : supprime l'effet de relief

Les divisions

- `<pre>` Indique au navigateur d'afficher le texte avec la mise en page exacte du fichier source. Les caractères utilisent une police non proportionnelle.

Ceci est un exemple de tableau au format texte

```
<pre>
```

```
+-----+-----+-----+
| lundi | mardi | mercredi |
|       |       |       |
+-----+-----+-----+
```

```
</pre>
```

Ceci est un exemple de tableau au format texte

```
+-----+-----+-----+
| lundi | mardi | mercredi |
+-----+-----+-----+
```

```
+-----+-----+-----+
| lundi | mardi | mercredi |
|       |       |       |
+-----+-----+-----+
```


Les listes

- La liste descriptive <dl> .. </dl>
- <dt> identification de l'élément
- <dd> description de l'élément

<dl>

<dt>Première loi</dt>

<dd>Un robot ne peut blesser un être humain ni, par son inaction, permettre qu'un humain soit blessé.</dd>

<dt>Seconde loi</dt>

<dd>Un robot doit obéir aux ordres donnés par les êtres humains, sauf si de tels ordres sont en contradiction avec la Première Loi.</dd>

<dt>Troisième loi</dt>

<dd>...</dd>

</dl>

Première loi

Un robot ne peut blesser un être humain ni, par son inaction, permettre qu'un humain soit blessé.

Seconde loi

Un robot doit obéir aux ordres donnés par les êtres humains, sauf si de tels ordres sont en contradiction avec la Première Loi.

Troisième loi

...

Les listes

- Les listes ordonnées ... et non ordonnées ...

```
<ol>
  <li>Premier point
    <ul>
      <li type=square>Sous point 1</li>
      <li type=disc>Sous point 2</li>
    </ul>
  </li>

  <li>Deuxième point
  </li>
</ol>
```

```
1. Premier point
   ■ Sous point 1
   • Sous point 2
2. Deuxième point
```

Styles

- Un certain nombre de balises existent pour modifier la présentation, par exemple la taille, la police, ou la couleur du texte.
- Deux types de présentation coexistent dans HTML
 - Les styles physiques indiquent explicitement quel type de caractères utiliser.

```
<b>texte en gras</b><br>  
<i>texte en italique</i><br>  
<u>texte souligné</u><br>  
texte normal<br>  
<tt>texte type machine &agrave; &eacute;crire</tt>
```

texte en gras
texte en italique
texte souligné
texte normal
texte type machine à écrire

Styles

- Styles logiques : attention l'affichage peut dépendre des navigateurs
 - `<cite>...</cite>`
 - `<code>...</code>`
 - `<dfn>...</dfn>`
 - `...`
 - `<kbd>...</kbd>`
 - `<samp>...</samp>`
 - `...`
 - `<var>...</var>`
- Styles pour les paragraphes
 - `<blockquote>...</blockquote>`
 - `<address>...</address>`
 - `<pre>...</pre>`

Styles

Avec des dialogues aussi poignants que
<cite>Pour survivre à la guerre, il
faut devenir la guerre !</cite> et
<cite>Adrieeeeennnee !!</cite>

Pour éteindre l'ordinateur entrez la
commande : <code>shutdown -h now</code>

Youyou : <dfn>Petit canot, canot
pneumatique</dfn>

Il a été très
méchant

Il a été vraiment</
strong> très méchant

Avec des dialogues aussi poignants que *Pour survivre à
la guerre, il faut devenir la guerre !* et *Adrieeeeennnee !!*

Pour éteindre l'ordinateur entrez la commande :
shutdown -h now

Youyou : Petit canot, canot pneumatique

Il a été *très* méchant

Il a été **vraiment** très méchant

Styles

- Rappel :
 - Attention à ne pas utiliser les balises de style à mauvais escient.
 - Par exemple, si vous voulez que votre titre apparaisse en gros, utilisez `<h1>` (le cas échéant en utilisant une CSS pour indiquer que `h1` devra utiliser une grande police) plutôt que de spécifier directement la police dans le code source. (Spécifier manuellement la police fait perdre l'information structurelle. Un programme d'analyse automatique n'aurait aucun moyen de retrouver le titre de votre document par exemple).

Les liens

- Objectif : permettre la navigation de documents en documents en cliquant sur des zones actives. Ces zone actives peuvent être du texte, des images, ... on parle aussi de “pointeurs”
- Il est possible d’avoir des liens qui pointent vers une autre partie d’un même document (par exemple “revenir en haut de la page” dans le cas d’une très longue liste).On parle alors de lien “interne”.
- Lorsque l’on pointe vers un autre fichier HTML, on parle de lien “externe”
- Par défaut, les navigateurs mettent en valeur les liens pour qu’on les reconnaissent au premier coup d’œil (par exemple, texte souligné)

Les liens

- On utilise la balise `<a>`
- Suivant si l'on veut définir un lien ou une ancre, on utilise des attributs différents
 - href pour un lien
 - name pour définir une ancre (l'endroit où un lien pourra pointer)

Les liens

```
<a name="zebulon" />
```

```
lien vers le fichier <a href="essai.html">essai</a><br>
```

```
lien vers le fichier <a href="essai2.html">essai2 <b>glop  
glop</b></a><br>
```

```
lien vers l'ancre zebulon de <a  
href="essai.html#zebulon">essai, en haut de la page</a><br>
```

```
lien vers l'ancre calimero de <a  
href="essai2.html#calimero">essai2</a>
```

lien vers le fichier [essai](#)

lien vers le fichier [essai2](#) **glop glop**

lien vers l'ancre zebulon de [essai, en haut de la page](#)

lien vers l'ancre calimero de [essai2](#)

Les tableaux

- Les tableaux sont largement utilisés en HTML, pas seulement pour la présentation de données sous forme tabulaire, mais aussi directement pour la mise en page.
- Les cases d'un tableau peuvent contenir pratiquement n'importe quoi (texte, image, liste, autres tableaux ...)
- Un tableau est constitué de lignes énumérées de haut en bas, elles mêmes constituées de cases énumérées de la gauche vers la droite.
- En ce qui concerne la mise en page, ATTENTION, cette dernière est maintenant souvent avantageusement réalisée grâce aux CSS

Les tableaux

```
<table border>
<caption><font size="+1">Nombre de kilom&egrave;tres parcourus</caption>
<tr>
<td></td><th colspan=5>Kilom&egrave;tres</th>
</tr>

<tr>
<td></td><th>1er trimestre</th><th>2eme trimestre</th><th>3eme trimestre</th><th>4eme
trimestre</th><th><i>Total</i></th>
</tr>

<tr align=center>
<td align=left>A pied</td><td>100</td><td>70</td><td>160</td><td>100</td><td>430</td>
</tr>

<tr align=center>
<td align=left>En voiture</td><td>800</td><td>900</td><td>400</td><td>1500</td><td>3600</td>
</tr>
</table>
```

	Kilomètres				
	1er trimestre	2eme trimestre	3eme trimestre	4eme trimestre	<i>Total</i>
A pied	100	70	160	100	430
En voiture	800	900	400	1500	3600

Les tableaux

```
<table border>
<caption><font size="+1" color=red>Nombre de kilomètres parcourus</caption>
<tr><td></td><th colspan=5>Kilomètres</th></tr>

<tr bgcolor=lightblue>
<td></td><th>1er trimestre</th><th>2eme trimestre</th><th>3eme trimestre</th><th>4eme
trimestre</th><th><i>Total</i></th>
</tr>

<tr align=center>
<td align=left>A pied</td><td>100</td><td>70</td><td bgcolor=#FF5050>160</td><td>100</
td><td>430</td>
</tr>

<tr align=center>
<td align=left>En voiture</td><td>800</td><td>900</td><td>400</td><td bgcolor=#FF5050>1500</
td><td>3600</td>
</tr>
</table>
```

Nombre de kilomètres parcourus

	Kilomètres				
	1er trimestre	2eme trimestre	3eme trimestre	4eme trimestre	Total
A pied	100	70	160	100	430
En voiture	800	900	400	1500	3600

DIV

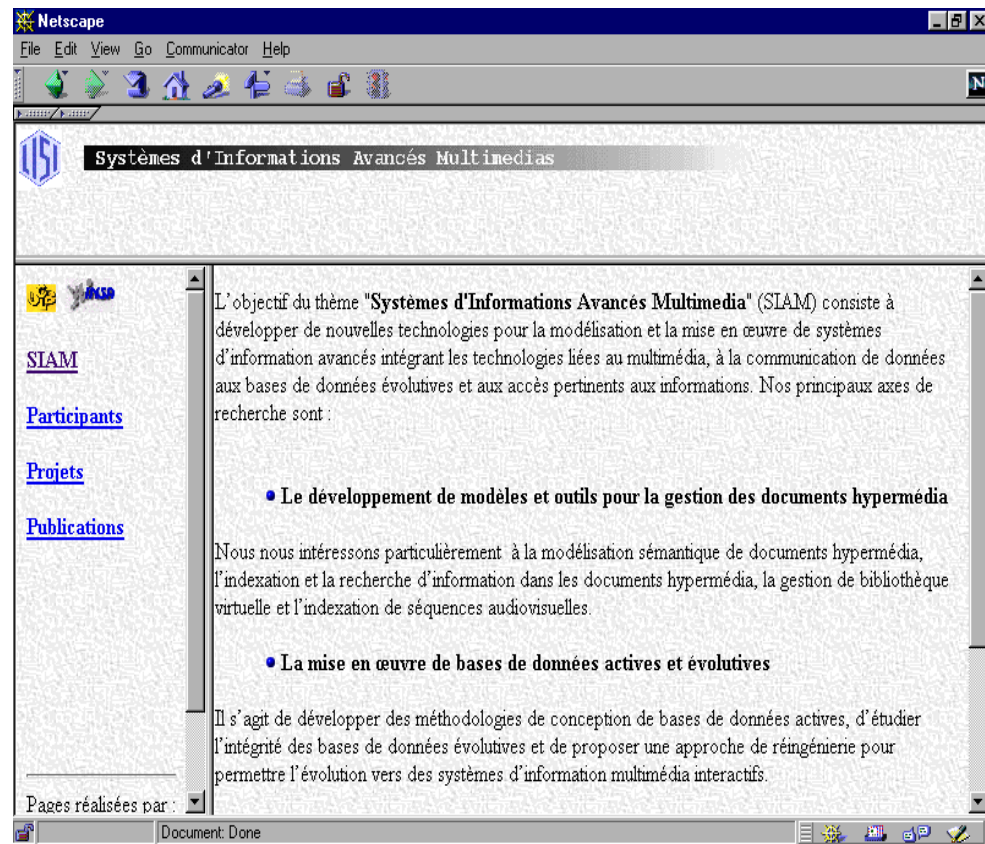
- Apparues lors de l'une des évolutions de HTML, les balises DIV et SPAN sont très utiles, en particulier lorsqu'on utilise conjointement les CSS
- En terme de mise en page, <div> définit l'alignement de la section balisée à l'intérieur de la page. Elle se comporte comme un début/fin de paragraphe. Il est donc possible d'avoir des paragraphes à l'intérieur d'un <div>, mais pas l'inverse.

HTML - frames

- Mutidocuments ou multicares (frames)
 - `<frameset>`
 - `<frame>`
- Une page principale ne contient que les balises `<frameset>` et `<frame>`, afin de spécifier la mise en page. Pour chacun des cadres, elle contient une référence sur le fichier html qui devra être affiché dedans.
- Attention, l'utilisation de ces balises de cadre et particulière, il ne faut pas mettre de balise `<body>` dans le fichier principal.

HTML - frames

```
<html>
<frameset rows="20%, 80%">
  <frame src="bandeau.html"
    name="bandeau"
    SCROLLING=NO
    marginheight=1>
  <frameset
    cols="20%, 80%"
    FRAMEBORDER=yes
    BORDER=1>
    <frame
      src="navigation.html"
      name="navigation" >
    <frame
      src="Intro.html"
      name="description"
      marginheight=14>
  </frameset>
</frameset>
</html>
```



HTML - frames

- Aide à la navigation / consultation
- `<frameset>`
 - `<frame>` ...
- Dans les fenêtres, liens avec un paramètre TARGET dans les ancres, utilisant les noms donnés dans les balises `<frame>` pour forcer l'affichage dans tel ou tel cadre
 - `cliquez ici`
 - `_blank` : nouvelle fenêtre sans nom
 - `_self` : valeur par défaut
 - `_parent`
 - `_top` : remplace toutes les frames de la fenêtre ouverte

Feuilles de style (Cascading Style Sheets)

CSS

- *Maintenir une cohérence de présentation / une cohérence graphique*
 - *Problématique quand la taille du site web augmente*
 - *Quand les pages sont maintenues par plusieurs personnes*
 - *Quand les pages les plus anciennes datent de plusieurs années*
- *Dans les navigateurs raisonnablement récents*
- *Définir des conventions de présentation qui sont appliquées automatiquement à chaque page du site*
- *En cas de modification de la feuille de style, toutes les pages qui l'utilisaient changent de présentation (-> mise à jour grandement simplifiée, cohérence assurée)*

CSS

- *Il est possible de définir une cascade de feuilles de styles. Exemple :*
 - *Une feuille de style générale qui définit la police de caractère et les couleurs utilisées pour le menu.*
 - *Puis des feuilles de style pour différencier par une couleur de fond différente chaque section du site.*
 - *Les utilisateurs peuvent avoir leur propre feuille de style (ex : très grande polices, ...) qui peuvent tout redéfinir*
- *A noter : gain en place par rapport à une définition des attributs d’affichage en “dur”*

CSS

- *Principe : plutôt que de spécifier les paramètres d’affichage “en dur” pour chaque élément de la page, on définit des règles indiquant comment les balises doivent être affichées.*
- *Ces règles peuvent porter sur une balise précise (“Le contenu des balises <h1> doit toujours être affiché de telle manière”).*
- *Il est aussi possible d’utiliser des mécanismes de classes, plus subtils, permettant d’appliquer un style à des groupes de balises.*
- *Exemple : pour définir les couleurs de texte et de fond par défaut :*

```
<style type="text/css">  
  body { color: black; background: white; }  
</style>
```

CSS

- *Plusieurs localisations possibles pour ces informations de style*
 - *En tête du fichier HTML, dans la section d'entête <head>...</head>*

```
<style type="text/css">  
  body { color: black; background: white; }  
</style>
```

- *Dans un fichier séparé (et donc beaucoup plus facile à modifier et à maintenir) avec une balise spéciale y faisant référence seulement dans <head>...</head>*

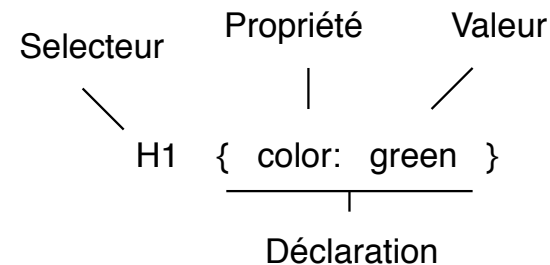
```
<link type="text/css" rel="stylesheet" href="style.css">
```

- *Le cas échéant (utilisation avec parcimonie), directement dans les éléments*

```
<p style="color:black; background:white">
```

CSS

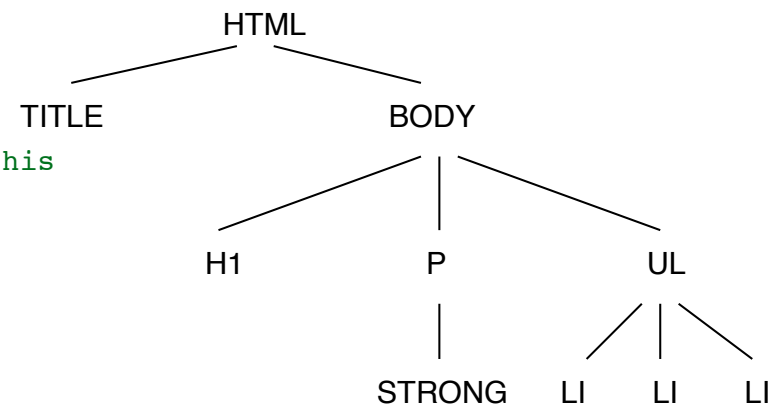
- *Les règles CSS ont un formalisme précis. Chaque règle possède un sélecteur, suivi d'un ou plusieurs couples propriété / valeur.*
- *Il y a de nombreuses propriétés*
 - *Couleur*
 - *Marges*
 - *Polices de caractères*
 - *...*
- *CSS est normalisé par le W3C. Les nouvelles normes CSS ne remplacent pas les anciennes, elles les complètent. CSS3 est découpé en de nombreux modules indépendants dont l'état de normalisation est plus ou moins avancé.*



<http://www.w3.org/TR/REC-CSS1> (CSS1, 1996 ~50 propriétés)
<http://www.w3.org/TR/REC-CSS2> (CSS2, 1998, ~70 propriétés)

CSS - Notion d'héritage

```
<html>
  <head>
    <title>Bach's home page</title>
  </head>
  <body>
    <h1>Bach's home page</h1>
    <p>Johann Sebastian Bach was a
      <strong>prolific</strong> composer. Among his
        works are:
    <ul>
      <li>the Goldberg Variations
      <li>the Brandenburg Concertos
      <li>the Christmas Oratorio
    </ul>
  </body>
</html>
```



```
<style type="text/css">
  h1, h2, p, li { color: green }
</style>
```

```
<style type="text/css">
  body { color: green }
</style>
```

Bach's home page

Johann Sebastian Bach was a prolific composer. Among his works are:

- the Goldberg Variations
- the Brandenburg Concertos
- the Christmas Oratorio

Sélecteurs

- *Le sélecteur indique à quel élément sera appliqué le style.*
- *Dans un document html, le sélecteur est normalement une balise définie en html (ou un groupe de balises)*
- *Il est possible d'écrire plusieurs règles consécutives*

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

```
h2 { color : red; }
```

- *Il est aussi possible d'utiliser une écriture condensée*

```
h1, h2 {color : red; }
```

```
/* se lit "appliquer la règle à tous les  
éléments h1 ET à tous les éléments h2 */
```


Identifiants

- *Il est souvent souhaitable de traiter de manière particulière une certaine instance d'un élément. Par Exemple nous souhaitons que tous les h1 soient écrits en noir, sauf un.*
- *Dans ce cas il est possible de donner un identifiant **unique** à l'élément qui nous intéresse, et de spécifier une règle pour lui.*
- *Dans le code html il faut spécifier l'identifiant de l'élément en utilisant l'attribut "id"*

```
<h1 id="leTitreRouge">Sommaire</h1>  
<h1> Chapitre 1 </h1>
```

- *Et dans la feuille de style on utilise comme sélecteur le symbole "#" suivi de l'identifiant choisi*

```
#leTitreRouge { color : red; } /* sélectionne  
l'unique élément du document dont  
l'identifiant est "leTitreRouge" */
```

Classes

- *Il est par ailleurs souvent souhaitable de traiter de manière particulière un groupe d'éléments. Par exemple nous souhaitons que certains paragraphes soient encadrés (mais pas tous).*
- *Dans le code html il faut spécifier la classe à laquelle appartient l'élément en utilisant l'attribut "class"*

```
<p>Ici un paragraphe normal</p>  
<p class="important">Ici un texte encadré</p>  
<p class="important">Ici un autre</p>
```

- *Et dans la feuille de style on utilise comme sélecteur le symbole "." suivi du nom de la classe choisi*

```
/* sélectionne tous les éléments du document dont la classe  
est "important" et les encadre*/
```

```
.important { border : solid; }
```

- *Le nom des classes est libre. Il faut utiliser des noms explicites, qui vont faciliter la travail de correction des erreurs et de maintenance*

Classes

- Un élément peut posséder plusieurs classes, les styles ainsi définis se cumulent

```
<p class="important enVert">Blablabli blablabla</p>
```

```
/* ici le texte sera écrit en vert et sera encadré */  
.important { border : solid; }  
.enVert { color : green; }
```

- Il est aussi possible d'être restrictif sur le type de l'élément

```
/* tous les éléments importants sont encadrés */  
.important { border : solid; }
```

```
/* les h1 de classe "important" ont un fond jaune */  
h1.important { background-color : yellow; }
```

```
/* les p de classe "important" ont un fond rouge */  
p.important { background-color : red }
```

Sélecteurs avancés

- *Il est possible de sélectionner des éléments en fonction de leur ascendance*
- *Par exemple je souhaite que les titres qui se trouveraient dans une division d'id "menu" soient traités différemment des autres titres*

```
/* "les éléments h1 dont un ascendant quelconque possède  
l'identifiant "menu" (ce peut être l'élément père, grand-père,  
etc.) */
```

```
#menu h1 { color : red }
```

- *Il est très courant de structurer le document html avec des divisions ou des spans. On y range ensuite les contenus, et l'on en modifie le style en fonction des divisions ou spans dans lesquels ils se trouvent*
 - *Une division pour le menu*
 - *Des division pour chaque article présenté sur le site*
 - *etc.*

Sélecteurs avancés

- *Il est aussi possible de spécifier une ascendance directe grâce à l'opérateur ">"*

/ les éléments en gras directement contenus dans un paragraphe de classe "important" */*

p.important > b { color : red }

- *Il est possible de sélectionner des éléments en fonction de l'élément qui précède dans le code html en utilisant l'opérateur "+"*

/ Met en italique les paragraphes qui suivent immédiatement un titre de type h1 */*

h1 + p { font-style : italic; }

Prépondérance des règles

- *Il est tout à fait possible que des règles soient antinomiques (un règle indique que les h1 sont écrits en vert, une autre règle indique qu'ils doivent être écrits en vert).*
- *Il existe un ordre de priorité bien défini qui empêche les ambiguïtés. C'est ce que l'on appelle "la cascade".*
- *Tout d'abord, les règles peuvent être définies à trois endroits.*
 - *Les règles par défaut contenues dans le navigateur sont les moins prioritaires. Elles sont appliquée quand rien d'autre n'est spécifié*
 - *Les règles du développeur du site écrasent les règles par défaut. Le développeur spécifie ses règles dans le code html ou dans un fichier de style attaché*
 - *Les règles de l'utilisateur, si elles sont définies, peuvent écraser toutes les autres règles (par exemple un utilisateur peut forcer la taille de la police, indépendamment de ce que voulait le concepteur du site).*

Prépondérance des règles

- *Mais dans un même niveau de priorité, il est encore possible que des règles s'opposent.*
- *C'est donc la **spécificité** qui l'emporte. Plus une règle est spécifique, plus elle est prioritaire.*

```
/* tout doit être écrit en noir. Spécificité minimale */  
* { color black }
```

```
/* mais tous les h1 sont écrits en bleu */  
h1 { color : blue }
```

```
/* sauf ceux dont la classe est "important" */  
h1.important { color : yellow; }
```

```
/* le h1 dont l'identifiant est "leTitreRouge" est traité différemment */  
h1#important { color : red; }
```

Prépondérance des règles

- Plus exactement, la spécificité d'une règle peut être écrite sous la forme de 4 nombres, par exemple 0,0,0,0
- Si le style est déclaré en ligne, ajouter 1,0,0,0

```
<p style="color : red"> blablabla </p>
```
- Pour chaque valeur de l'attribut "id" donné dans le sélecteur, ajouter 0,1,0,0
- Pour chaque valeur de l'attribut "class" donné dans le sélecteur, ajouter 0,0,1,0
- Pour chaque élément donné dans le sélecteur, ajouter 0,0,0,1

Prépondérance des règles

h1 { color : red }	0,0,0,1
p em { color : purple }	0,0,0,2
.important	0,0,1,0
p.important	0,0,1,1
#leSommaire	0,1,0,0
#leSommaire h1	0,1,0,1
etc.	

- *La résolution se fait en partant de la gauche. Entre deux règles, celle dont le chiffre de gauche est le plus élevé est prioritaire. Si les deux chiffres sont identiques, on regarde le chiffre immédiatement à droite, et ainsi de suite.*
- *Si les spécificités sont identiques, la règle déclarée en dernier est prioritaire (soyez donc attentifs lorsque vous utilisez plusieurs fichiers css pour une même page web)*

CSS - dimensions

- Les dimensions (des marges, des caractères, des images, etc.) peuvent être exprimées dans différentes unités:
 - En pourcentage par rapport à l'élément conteneur
 - En millimètres, pouces, etc.
 - En pixels
 - En em

CSS - Des exemples

- *Gestion des marges à gauche et à droite*

```
<style type="text/css">
  body { margin-left: 10%; margin-right: 10%; }
  h1 { margin-left: -8%;}
  h2,h3,h4,h5,h6 { margin-left: -4%; }
</style>
```

- *Gestion des blancs entre les paragraphes*

```
h2 { margin-top: 8em; margin-bottom: 3em; }
```

- *Indentation en début de paragraphe*

```
p { text-indent: 2em; margin-top: 0; margin-bottom: 0; }
```

CSS - Des exemples

```
<HEAD>
<TITLE>Page de test pour les css</TITLE>
<LINK type="text/css" rel="stylesheet" href="test.css">
</HEAD>
```

```
<BODY>
<H1>Titre principal (H1)</H1>
```

```
<H2>Sous titre (H2)</H2>
```

```
<P>Ceci est du texte normal sur quelques lignes.
Remarquez la présence d'une indentation au début
du paragraphe (grâce à "text-indent").</P>
```

```
<H2 class=special>Sous titre (H2, class=special)</H2>
```

```
<P>Ceci est du texte normal dans un paragraphe.</P>
```

```
<P>Ceci est aussi du texte normal dans un paragraphe.</P>
```

```
<H2>Sous titre(H2)</H2>
```

```
<P>Encore un peu de texte dans un paragraphe (P).</P>
```

```
<P class="encadrelarge">Ce texte-ci est aussi dans un
paragraphe, mais de la classe "encadrelarge". Remarquez
que l'indentation a été héritée du paragraphe "normal".</P>
```

```
</BODY>
```

```
H1 {      color:yellow;
          margin-left:-8% }
```

```
BODY {    color:white;
          background:darkblue;
          margin-left:10%;
          margin-right:10%;
          font-family:verdana,sans-serif }
```

```
H2,H3,H4,H5,H6 { color:yellow;
                  margin-left:-4% }
```

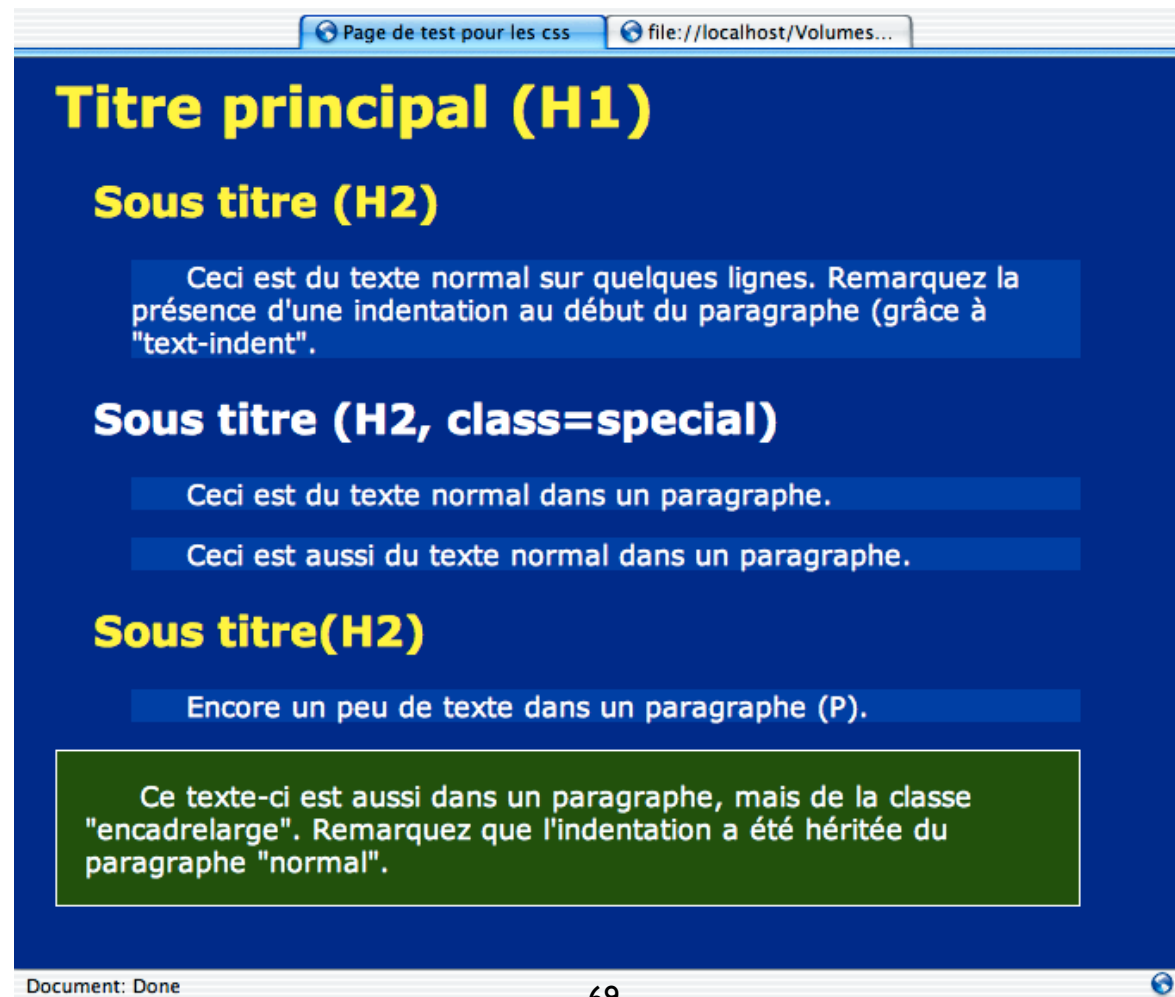
```
H2.special {  color:white }
```

```
P {          text-indent:2em;
          background:#4040A0 }
```

```
P.encadrelarge { padding:1em;
                  margin-left:-8%;
```

```
background:darkgreen;
                  border:solid;
                  border-width:thin }
```

CSS - Des exemples



CSS - Des exemples



Titre principal (H1)

Sous titre (H2)

Ceci est du texte normal sur quelques lignes. Remarquez la présence d'une indentation au début du paragraphe (grâce à "text-indent").

Sous titre (H2, class=special)

Ceci est du texte normal dans un paragraphe.

Ceci est aussi du texte normal dans un paragraphe.

Sous titre(H2)

Encore un peu de texte dans un paragraphe (P).

Ce texte-ci est aussi dans un paragraphe, mais de la classe "encadrelarge". Remarquez que l'indentation a été héritée du paragraphe "normal".