

# Publication (avancée) de données spatiales dans Internet

**GMT-20859 & SCG-66408**

**HTML & CSS**

**Thierry Badard & Frédéric Hubert**

Université Laval, Dép. des sciences géomatiques  
Québec, Canada

[{Thierry.Badard; Frederic.Hubert}@scg.ulaval.ca](mailto:{Thierry.Badard; Frederic.Hubert}@scg.ulaval.ca)



UNIVERSITÉ  
**LAVAL**

Cours donné le 15 septembre 2008

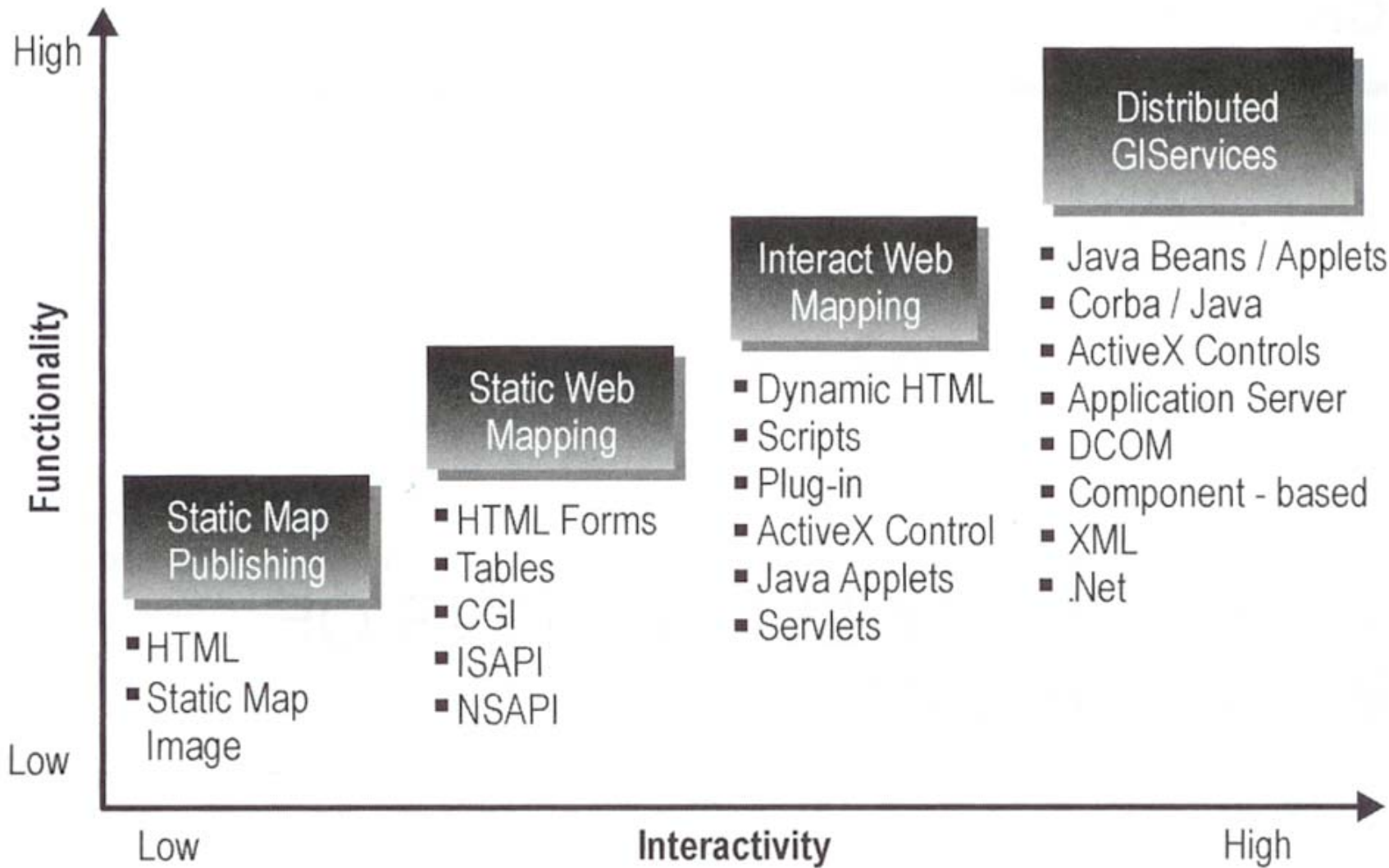


# *Contenu du cours.*

- Notions d'hypertexte et d'hypermédia
- HTML
  - Structure d'une page web
  - Éléments du langage
- Feuilles de style CSS
- Document Object Model
- Scripting côté client

# ***Notion d'hypertexte et d'hypermédia***

# Différentes architectures / technologies.

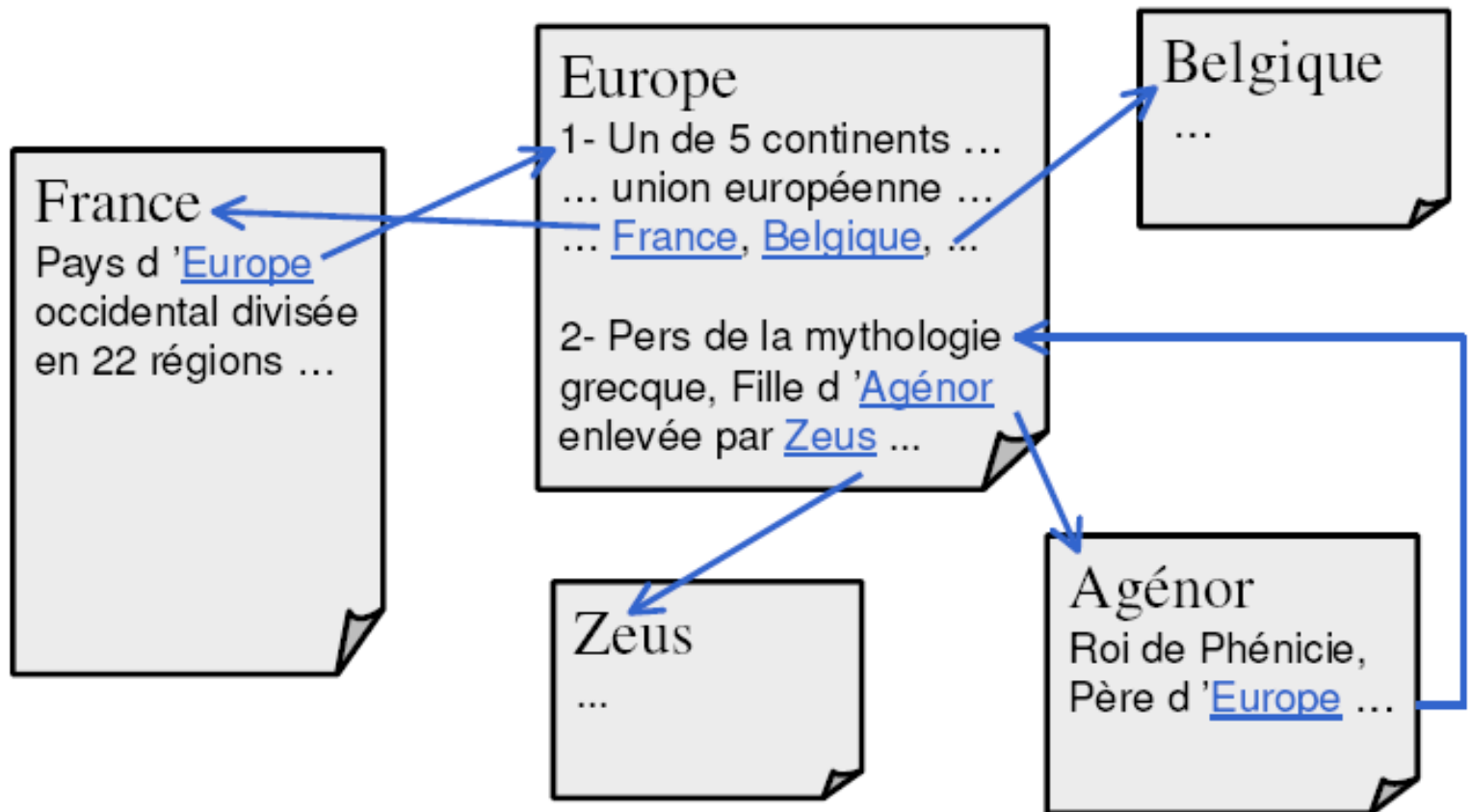


Sources : Peng & Tsou, 2003

# Notion d'hypertexte.

## ■ Hyperlien = Références entre documents

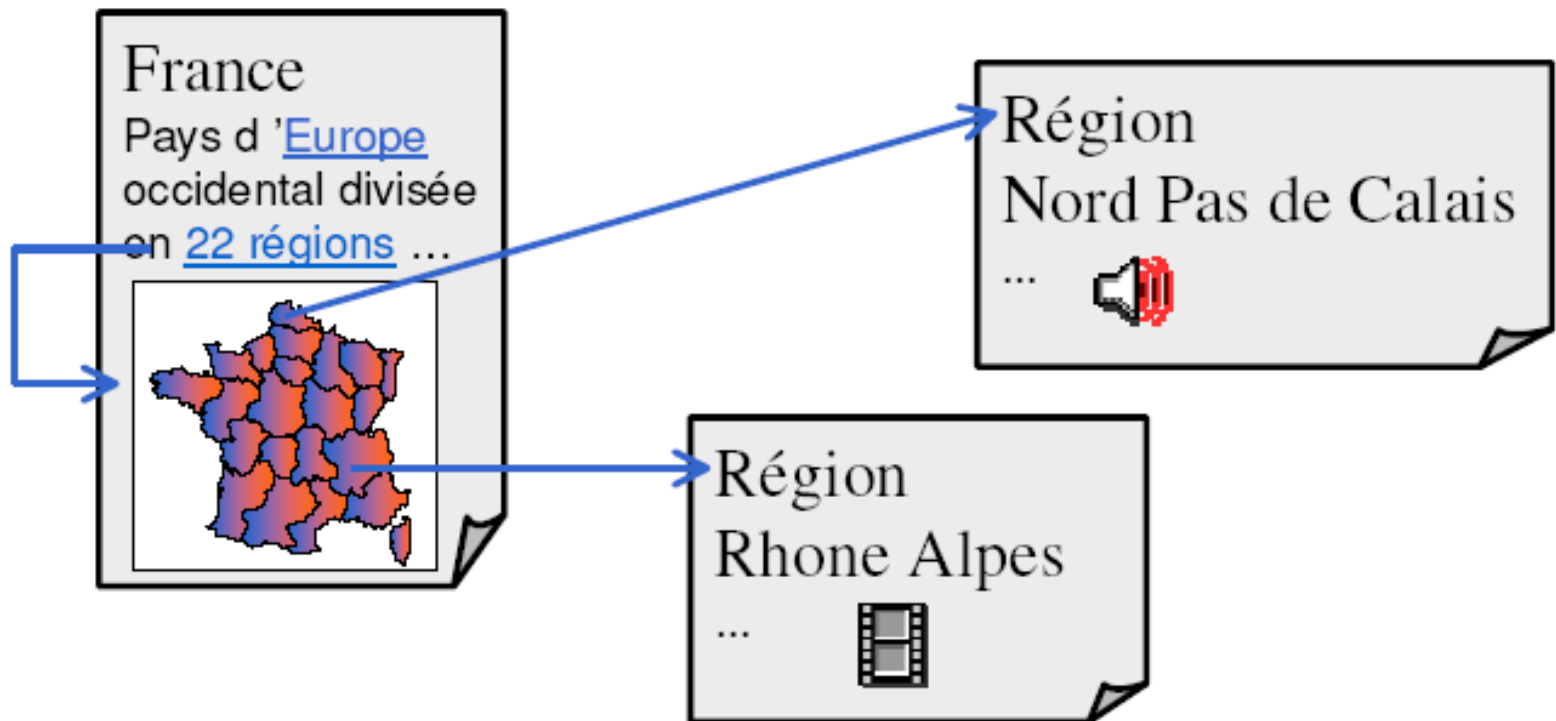
- navigation d'un document à l'autre



# Notion d'hypermédia.

## ■ Hypertexte enrichi du Multimédia

- Images, Sons, Vidéo, Animations contenant des zones sensibles (boutons ...)



# *HTML – HyperText Markup Language*

# HTML – HyperText Markup Language (1/2).

- Hypertext Markup Language
  - Hypertexte → lien dans document pointe vers autre document
  - Markup Language → code pour marquer des zones dans un document
- Né en 1989 au CERN (GE) basé sur SGML: Standard Generalized Markup Language
- Spécifié par le W3C (World Wide Web Consortium):  
<http://www.w3.org/TR/html4/>



# HTML – HyperText Markup Language (2/2)

- Langage de Description de Documents HyperTextuels
- Description textuelle du Document
  - Edition manuelle possible
  - Génération de documents par programmation (CGI, Scripting client)
  - Apparition d 'outils WYSIWYG d 'Authoring
    - HotDog, Netscape Gold, FrontPage, Macromedia DreamWeaver
- Structuration hiérarchique à la SGML <sup>\*</sup>
  - éléments HTML décrit par balisage (tag) du texte
    - `<ELEMENT ATTR=VALUE> ...</ELEMENT>`
  - il existe une DTD publique HTML
  - cependant les auteurs ne la respectent pas toujours
  - heureusement que les navigateurs sont tolérants
- Remarque
  - Les éléments HTML n 'ont pas tous une signification sémantique. Elle est surtout typographique !!! (voir SGML/XML)

Source: <http://www-adele.imag.fr/~donsez/cours/html.pdf>

\* SGML : *Standard Generalized Markup Language*

# HTML – Principe du HTML (1/2)

- Fonctionne sous la base de la balise (ou tag),
  - Ex: <html>, <p>, <table>
- Une balise formate le document en entier ou une partie du texte
- A (pratiquement) chaque balise correspond une balise fermante, ce qui délimite l'effet de la balise
  - <p>Ceci est un paragraphe.</p>
- Les retours de chariot ou les espaces blancs ne sont pas pris en compte

# HTML – Principe du HTML (2/2)

- Un document html peut être édité
  - dans un simple éditeur de texte,
  - à l'aide d'éditeurs plus ou moins sophistiqués (grand choix sur Internet)
  - généré par un traitement de texte ("Sauver sous...", mais le code n'est pas très bon)
- Nombreux éditeurs shareware ou freeware

Source: <http://atlcui.unige.ch/ntic/htmlbase.ppt>

## Exemples d'éditeurs :

- Adobe Dreamweaver : <http://www.adobe.com/fr/products/dreamweaver/>
- Microsoft Expression (anciennement Frontpage):  
<http://www.microsoft.com/france/expression/default.msp>
- PSPAD : <http://www.pspad.com/fr/>
- MS Notepad, MS Wordpad, GNU Emacs, Vi ...

# HTML – Premier exemple de document HTML

```
<HTML>
<HEAD>
  <TITLE>Document HTML</TITLE>
</HEAD>
<BODY BACKGROUND="fond.jpg">
  <H1 align=center>
    <IMG SRC="icon/logo.jpg">Titre
    <IMG SRC="icon/logo.jpg"></H1>
  <HR>
  <A HREF="/sommaire.htm">Sommaire</A>
  <A HREF="#CHAP1">Chapitre 1</A><BR>
  <A HREF="#CHAP2">Chapitre 2</A><BR>
  <H2 ID="CHAP1">Chapitre 1</H2>
    Texte libre (Paragraphe 1.1) <BR>
    Texte libre (Paragraphe 1.2) <BR>
  <H2 ID="CHAP2">Chapitre 2</H2>
    Texte libre (Paragraphe 2.1) <BR>
</BODY>
</HTML>
```



# HTML – Fichier HTML

- Le document a l'extension .html ou .htm
- Simple texte → facile à modifier
- La page principale dans un répertoire se nomme index.htm(l), (parfois aussi welcome.htm(l) ou default.htm(l))
  - C'est cette page qui est affichée par défaut si rien n'est spécifié comme dans

<http://geosoa.scg.ulaval.ca/~badard/>

équivalent de:

<http://geosoa.scg.ulaval.ca/~badard/index.html>

# HTML – Structure générale d'un doc. HTML

## Document HTML

### Entête

Titre : « Document HTML »

### Corps (couleur de fond = blanc)

Titre1 : « Titre »

Image : externe dans le fichier logo.gif

Ancre : « Sommaire » vers le fichier sommaire.htm

Titre : « Chapitre 1 »

Texte : « Texte Libre »



# HTML – Les balises

- Les balises sont emboîtées
  - `<p>Ceci est un paragraphe avec un mot <b>en gras</b>.</p>`
- Balises non sensibles à la casse: `<b>` = `<B>`, `<b>` `</B>` toléré
- Une balise peut avoir des attributs
  - `<p align="center">Paragraphe centré</p>`
- `<html>` `</html>` marque début et fin d'un doc
- `<head>` marque l'en-tête du document
- `<body>` marque le corps du document

Balises  
obligatoires !

## ■ Structure d'un doc html

```
<html>  
  <head>  
    <title>Mon titre</title>  
    ... autres infos  
  </head>  
  <body>  
    ... le corps du doc  
  </body>  
</html>
```



- `<title>` marque le titre du document
  - apparaît dans la barre de titre du navigateur (butineur ou browser) et utilisé pour les signets (bookmarks)
  - Attention: n'apparaît pas dans le corps du texte
- `<meta>` contient la méta-information
  - `<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">` contient le code de caractère utilisé par le document, assez important

# HTML – Caractères accentués et spéciaux (1/2)

## ■ = entité HTML

• caractère	code ISO8859-1	entité
• guillemet	&#34;	&quot; "
• et commercial	&#38;	&amp; &
• signe plus petit que <	&#60;	&lt;
• plus grand que	&#62;	&gt; >
• espace non sécable	&#160;	&nbsp; _
• livre sterling	&#163;	&pound; £
• c cédille	&#231;	&ccedil; ç
• e accent grave	&#232;	&egrave; è
• e accent aigu	&#233;	&eacute; é

## ■ Exemple

- a < b if c>d                      a &#60; b if c&#62;d                      a &lt; b if c&gt;d

# HTML – Caractères accentués et spéciaux (2/2)

©	&copy;	&#x00a9;	«	&laquo;	&#x00ab;
™	&trade;	&#x2122;	»	&raquo;	&#x00bb;
®	&reg	&#x00ae;	"	&quot;	&#x0022;
Esp. inséc	&nbsp;	&#x00a0;	&	&amp;	&#x0026;
<	&lt;	&#x003c;	¬	&not;	&#x00ac;
>	&gt;	&#x003e;	°	&ord;	&#x00ba;
¿	&iquest;	&#x00bf;	ª	&orda;	&#x00aa;
\$	-	&#x0024;	¢	&cent;	&#x00a2;
¥	&yen;	&#x00a5;	£	&pound;	&#x00a3;
¡	&iexcl;	&#x00a1;	€	&euro;	&#x20ac;

# HTML – La balise META (1/2)

## ■ Description du document

- Faciliter l'indexation
- Faciliter la sélection des documents

## ■ Metadata de HEAD décrivant le document

- `<META NAME="cle" CONTENT="valeur">`

`<meta NAME="keywords"`

`CONTENT="oclc,online,computer,library,center,home,menu,text-only ...`

`<meta NAME="DC.subject.keyword"`

`CONTENT="oclc,online,computer,library,center,home,menu,text-only, ...`

# HTML – La balise META (2/2)

## ■ Dublin Core, <http://purl.org/DC>

- 15 propriétés de base + des extensions

```
<TITLE>Dublin Core Metadata Element Set: Resource Page</TITLE>
```

```
<META name = "DC.subject" content = "dublin core metadata element set">
```

```
<META name = "DC.subject" content = "networked object description">
```

```
<META name = "DC.publisher" content = "OCLC Online Computer Library Center, Inc.">
```

```
<META name = "DC.author" type = "name" scheme = "AACR2" content = "Weibel, Stuart L..">
```

```
<META name = "DC.author" type = "email" content = "weibel@oclc.org">
```

```
<META name = "DC.author" type = "name" scheme = "AACR2" content = "Miller, Eric J.">
```

```
<META name = "DC.author" type = "email" content = "emiller@oclc.org">
```

```
<META name = "DC.title" content = "Dublin Core Element Set Reference Page">
```

```
<META name = "DC.date" type = "creation" scheme = "ISO" content = "1996-05-28">
```

```
<META name = "DC.form" scheme = "IMT" content="text/html">
```

```
<META name = "DC.language" scheme = "ISO 639" content="en">
```

```
<META name = "DC.identifiant" scheme = "URL" content = "http://purl.oclc.org/metadata/dublin_core">
```

# HTML – La balise META : Exemple

```
<html>
```

```
<head>
```

```
<title>Dr. Thierry Badard - Professeur au Centre de Recherche en  
Géomatique / Professor at Research Center for Geomatics</title>
```

```
<meta name="generator" content="phpWebSite" />
```

```
<meta name="keywords" content="Thierry Badard, GeoSOA, ...,  
Laval University, Universite Laval" />
```

```
<meta name="description" content="Site web personnel du Dr. Thierry  
Badard - Personal website of Thierry Badard, Ph.D." />
```

```
<meta name="robots" content="index, follow" />
```

```
<meta name="author" content="Dr. Thierry Badard" />
```

```
<meta name="owner" content="Thierry.Badard@scg.ulaval.ca" />
```

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

```
<script type="text/javascript" language="JavaScript">
```

```
...
```

```
</script>
```

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

```
</head>
```

```
<body> ... </body>
```

```
</html>
```

# HTML – Corps d'un document HTML

- `<body>` marque le corps du document
- Attributs
  - `background=""` url d'une image
  - `text` couleur du texte
  - `link` couleur des liens
  - `vlink` couleurs des liens visités
  - `alink` couleur des liens actifs
- On préfère définir ces choses dans une feuille de style (notion de CSS)

# HTML – Couleurs

- Code RGB = Red Green Blue
- Signe # 6 chiffres hexas, 2 digits par couleur
- #000000 = noir (ou utiliser "black")
- #ffffff = blanc = white
- #ff0000 = rouge = red
- #0000ff = bleu = blue
- Etc.

- Autre notation possible:

`rgb(rouge[0-255], vert[0-255], bleu[0-255])`

- Exemple: `rgb(255,0,0) = #ff0000 = rouge`



# HTML – Principaux marquages de texte

- Un paragraphe est marqué par `<p> </p>`
- Les différents titres sont marqués par `<h1>`, `<h2>` ... `<h6>` (ordre décroissant)
- `<p>` ou `<h1>` etc: paramètre `align = "left", "right", "center"...`
- `<br>` insère un simple saut de ligne (pas fermée)
- `<address>` permet de faire un pied de page (copyright, adresse webmaster...)

# HTML – Les titres : Exemple

- `<H1>Chapitre 1</H1>`
- `<H2>Section 1.1</H2>`
- `<H3>Sous Section 1.1.1</H3>`

## ■ Exemple

`<H1>Table of Contents</H1>`

`<P><A href="#section1">Introduction</A><BR>`

`<A href="#section2">Some background</A><BR>`

`<A href="#section2.1">On a more personal note</A><`

`...the rest of the table of contents...`

`...the document body...`

`<H2 id="section1">Introduction</H2>`

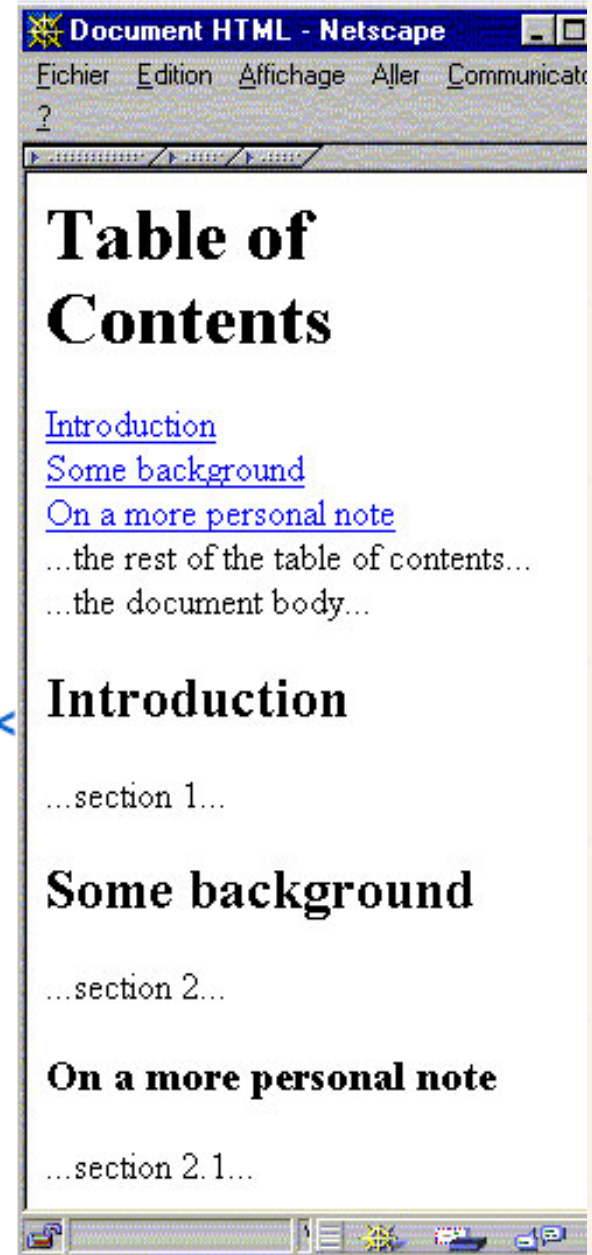
`...section 1...`

`<H2 id="section2">Some background</H2>`

`...section 2...`

`<H3 id="section2.1">On a more personal note</H3>`

`...section 2.1...`



# HTML – Les commentaires

- Entre `<!-- -->`
- Comme dans la programmation peut servir à désactiver du code
- Parfois pour du code interprété
  - `<!--#echo var="Last_modified" -->`
  - `<!--#include virtual="/info1/docbbox.html"-->`
- Parfois cache du code JavaScript pour des browsers archaïques
  - ```
<script type="text/javascript"><!--  
function popup(link,framename) {  
window.open(link,framename,  
'toolbar=0,location=0,directories=0,status=0,  
scrollbars=1,resizable=1,copyhistory=0,menuBar=0,  
width=740,height=520'); }  
--></script>
```

# HTML – Formatage du texte

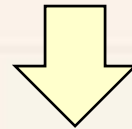
- `<sup>` pour mise en exposant (superscript)
- `<sub>` pour indices (subscript)
- `<b>` pour gras (bold)
- `<i>` pour italiques
- `<u>` pour souligné (underline)
- `<tt>` pour teletype (courrier)
- `<big>` pour texte plus grand
- `<small>` pour texte plus petit
- `<strike>` et `<s>` pour barré

# HTML – Police et taille du texte

- `<font>` pour changer la taille et la police
- **Attributs**
  - `color` pour modifier le style de couleur
  - `size`. Différentes tailles de 1 à 7, ou définir relativement à taille par défaut: -1, 0, +1 etc.
  - `face`. Définit la police. Plusieurs dans la liste par ordre de préférence. Seuls certaines polices sont reconnues univerelles (Times, Helvetica (Arial), Comic, Verdana, serif, sans-serif, courrier
- `<basefont>` pour définir taille et police de base dans le `<head>`
- De préférence utiliser les styles

# HTML – Police et taille du texte : Exemples

```
<p>  
  <font size="7">texte assez énorme</font><br>  
  <font size="1">texte assez minuscule</font><br>  
  <font size="+1">texte un peu plus grand que la normale</font><br>  
  <font size="-2">texte beaucoup plus petit que la normale</font>  
</p>  
  
<p>  
  <font color="#FF0000">texte rouge vif</font><br>  
  <font color="#00C000">texte vert</font><br>  
  <font color="#0000FF">texte bleus  
</p>
```



texte assez énorme

texte assez minuscule

texte un peu plus grand que la normale

texte beaucoup plus petit que la normale

texte rouge vif

texte vert

texte bleu

- `<hr>` (pas fermée)
  - `size` pour taille (épaisseur)
  - `width` taille en pixels ou en % (`width="50%"`)
  - `noshade="noshade"`
  - `align = "left", "right"...`

- Listes à puces `<ul>`
  - `type= "disc", "square", "circle"`
  - `compact="compact"`
- Listes numérotées `<ol>`
  - `type= "1", "A", "a", "i", "I"`
  - `start = "3"` (attention: tjs valeur num.:  
ça correspond à 3, C, c, iii ou III)
- Element de liste `<li>`



## HTML – Listes (2/2)

- Plusieurs niveaux emboîtables, types de liste également (indentation des niveaux juste pour clarté visuelle, pas obligatoire)

```
<ul>
```

```
<li>Module AB3:</li>
```

```
<ol type="a">
```

```
<li>Cours JAVA</li>
```

```
<li>Cours de logique formelle</li>
```

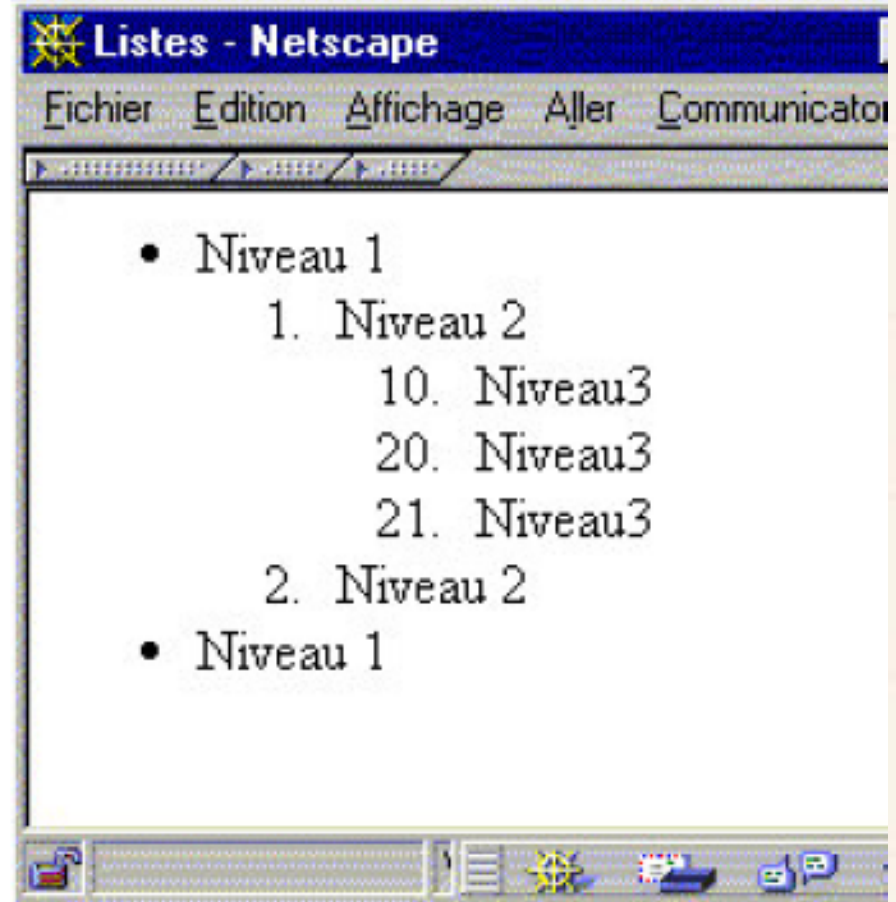
```
</ol>
```

```
<li>Module AB4:</li> ...
```

```
</ul>
```

# HTML – Listes : Exemple

```
<UL>
  <LI>Niveau 1
  <OL>
    <LI>Niveau 2
    <OL>
      <LI VALUE="10">Niveau3
      <LI VALUE="20">Niveau3
      <LI>Niveau3
    </OL>
    <LI>Niveau 2
  </OL>
  <LI>Niveau 1
</UL>
```



# HTML – Listes de définitions

- Ouverture d'une liste <dl>
- Terme à définir <dt>
- Définition <dd>

```
<dl>
```

```
<dt>Impressionnisme</dt>
```

```
<dd>Courant de l'art pictural du XIXe siècle.  
Peintres les plus connus: Van Gogh, Claude  
Monnet, Edouard Manet, Paul Cézanne...</dd>
```

```
<dt>Fauvisme</dt>
```

```
<dd>...</dd>
```

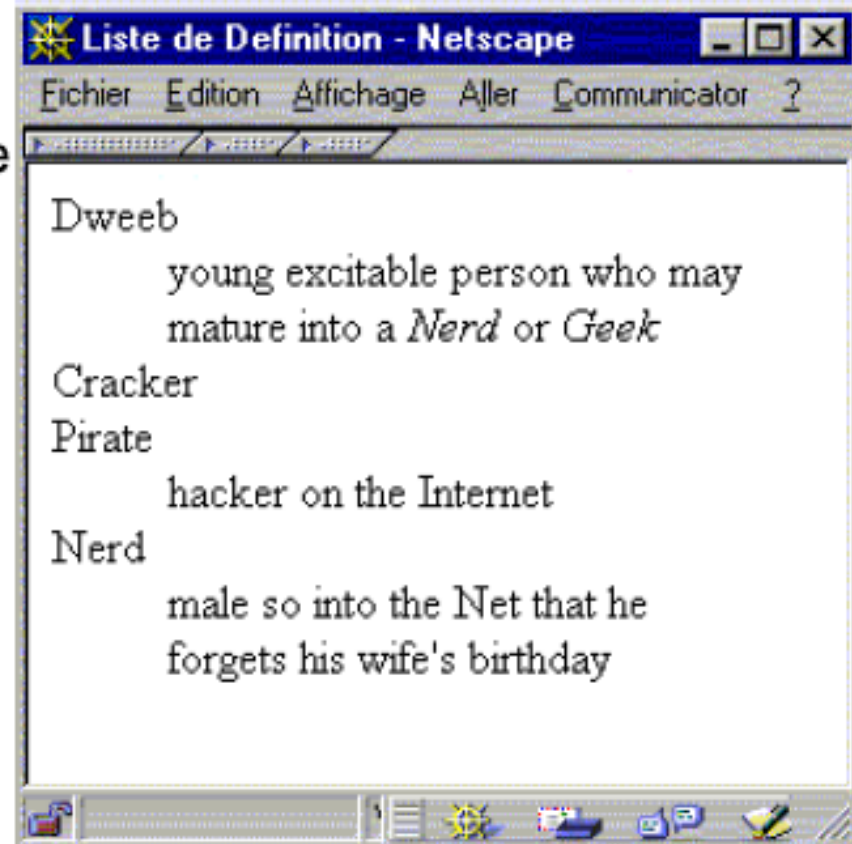
```
</dl>
```

# HTML – Listes de définitions : Exemple

```
<DL>
  <DT>Dweeb
  <DD>young excitable person who may mature
    into a <EM>Nerd</EM> or <EM>Geek</EM>

  <DT>Cracker <DT>Pirate
  <DD>hacker on the Internet

  <DT>Nerd
  <DD>male so into the Net that he forgets
    his wife's birthday
</DL>
```



# HTML – Les hyperliens

- HyperLien vers un autre document (ou une autre partie du document)
  - Sur un texte, le lien se présente sous la forme d 'un texte souligné de couleur bleue
  - Sur une image, un cadre de couleur bleue entoure celle-ci

## ■ Les ancres A (*Anchor*)

For more information about W3C, please consult the

```
<A href="http://www.w3c.org/">W3C Web site</A>.
```

## ■ Les liens LINK

```
<HEAD>
```

```
...other head information...
```

```
<TITLE>Chapter 5</TITLE>
```

```
<LINK rel="prev" href="chapter4.html">
```

```
<LINK rel="next" href="chapter6.html">
```

```
</HEAD>
```

# HTML – URL (Uniform Resource Locator)

## ■ Désignation d'un document sur le Web

- URL absolue

`http://www.univ-valenciennes.fr/limav/index.html`

`ftp://ftp.lip6.fr/pub`

`http://glardax:8000/~donsez/index.htm#rech`

`http://glardax:8000/limav/donsez/donsez.jpg`

- Format d'une URL pour HTTP, FTP

`protocole://dnsmachine:port/rep/fic#target`

NB: URI (Uniform Resource Identifier) = URL

## ■ Remarque

`news:comp.text.sgml`

`mailto:donsez@univ-valenciennes.fr`

# HTML – URL relative

## Résolution [RFC1808] d'une URL relative au document courant

Document Courant = `http://glardax:8000/limav/index.html`

`/limav/indexgb.htm`

`http://glardax:8000/limav/indexgb.htm`

`../pub`

`http://glardax:8000/pub`

`~dousez/dousez.jpg`

`http://glardax:8000/~dousez/dousez.jpg`

`#rech`

`http://glardax:8000/limav/index.html#rech`

`dousez`

`http://glardax:8000/limav/dousez`

`<BASE ...>` URL de base pour la résolution

`<base href="http://houebe/limav/index.html">`

# HTML – Créer un hyperlien

- Créer un lien (sur image ou texte) `<a>`
- Paramètre `href` → URI/URL de la page à atteindre

```
<a href="http://www.latl.unige.ch/info3/ntic/">La page du cours NTIC</a>
```

- Paramètre `target="_blank"` → nouvelle fenêtre navigateur



# HTML – Notion de port TCP/IP

- Port TCP/IP → lieu de connection logique parfois associée à un protocole particulier (20 ou 21 = ftp, 80 ou 8000 = http)
- On peut installer plusieurs serveurs web sur même machine mais écoutent différents ports
- On peut attribuer un port à une partie de site → éviter d'avoir long chemin pour accéder
- Adresse serveur : port

`http://geosoa.scg.ulaval.ca:8080`

`http://132.203.82.194:80`

# HTML – Les ancres (1/2)

- Marquer une section dans un doc (dans docs longs, appel de notes vers notes etc.)
- Un lien fait ensuite appel à cette ancre (dans le même doc ou dans un autre) → le navigateur affiche le document au bon endroit
- Paramètre `name` dans `<a>`
- Eventuellement balise unique
  - `<a name="monAncre" />`
  - `<a name="sec3.1">3.1</a>`

## ■ Lien vers ancre

- `<a href="#monAncre">Lien</a>` (même doc)
- `<a href="http://www.unige.ch/toto.html#monAncre">` (vers autre doc)

## ■ On peut combiner ancre et lien dans la même balise

- `<a name="appel1" href="#note1">1</a>...`  
et plus loin  
`<ol><li><a name="note1"/>Ceci est une  
note <a href="appel1">Retour à  
l'appel</a>`

# HTML – Les images (1/2)

- Insérer une image `<img>`
  - Source de l'image (avec URI) `src`
  - Description alternative `alt`. Peut servir comme indice car affiché si la souris survole l'image
  - Taille en pixels `width` et `height`. Permet d'accélérer l'affichage du texte en cas de connexion lente
  - Espace autour `hspace` et `vspace` (pixels)
  - Bordure autour `border` (en pixels) `border="0"` si image sert de lien
  - Alignement `align=`  
"`bottom|middle|top|left|right|`  
`absmiddle|baseline|texttop|absbottom`"

## ■ Élément IMG

```
<IMG SRC="/icon/logo.gif" ALT="My Company Logo">
```

- image incluse dans le document
- Attribut optionnel

positionnement par rapport au texte

» ALIGN = TOP, MIDDLE, BOTTOM.

Forcer la taille de l'image

» width=180 height=60

» *évite les reformatages intempestifs  
au chargement du document*

- Remarque

- Type d'image généralement supporté  
GIF, JPEG, GIF Animé, XBM

- Plug-In pour les autres

signifie le chargement du PlugIn et son installation (Fat-Browser)

# HTML – Les images : Exemple

```
<html>
<head>
<title>Texte du titre</title>
</head>
<body>
<h1>Ours dansant</h1>
<p></p>

</body>
</html>
```

**Ours dansant**



# HTML – Les images « mappées » (1/2)

## ■ ISMAP et /cgi-bin/imagemap

- une CGI récupère la position du click dans l'image
- et retourne une page
- Non standalone, Configuration compliquée, Obsolète

## ■ USEMAP et élément <MAP>

- gérer par le client
- HREF url ou appel d'une fonction javascript

## ■ Applet Java Imagemap

- personnalisation des formes, des actions ou des effets visuels
  - voir l'applet `imagemap.class` dans les démos du JDK

# HTML – Les images « mappées » (2/2)

## ■ Déclaration d'une MAP

```
<map name="bureau">
  <area shape="polygon" alt="index de ces pages"
    coords="114,152,534,150,607,335,100,295,134,296,133,216,94,216,111,151"
    href="thehtml.htm">
  <area shape="circle" alt="fichier d' aide" coords="317,111,33" href="help.htm">
  <area shape="rect" alt="ICQ" coords="15,53,87,182" href="javascript:play('ping.au')">
  <area shape="polygon" coords="99,16,227,2,235,18,241,96,112,113,98,33"
    href="links.htm">
  <area shape="rect" coords="103,219,130,291" href="mailto:donsez@uvhc.fr">
  <area shape="rect" coords="0,0,639,359" href="author.htm" >
</map>
```

## ■ Définition d'une image sensible

```

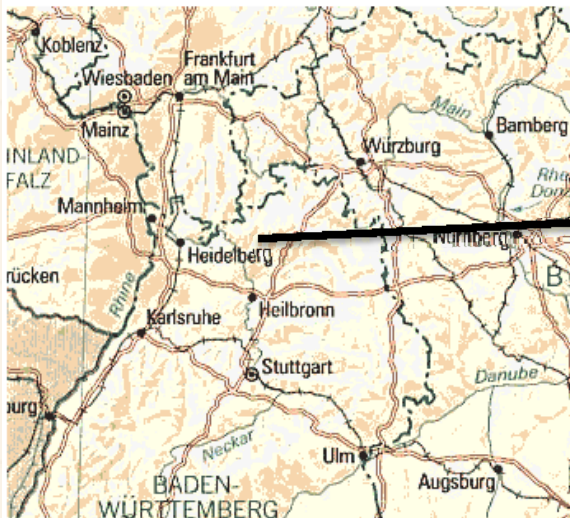
```



# HTML – Les images « mappées » : Exemple

```
<h1>Vite à la ville ou région de votre choix!</h1>
<map name="carte">
  <area shape="rect" coords="11,10,59,29" href="http://www.koblenz.de/" alt="Coblence">
  <area shape="rect" coords="42,36,96,57" href="http://www.wiesbaden.de/" alt="Wiesbaden">
  <area shape="rect" coords="42,59,78,80" href="http://www.mainz.de/" alt="Mayence">
  <area shape="rect" coords="100,26,152,58" href="http://www.frankfurt.de/"
alt="Francfort">
  <area shape="rect" coords="27,113,93,134" href="http://www.mannheim.de/" alt="Mannheim">
  <area shape="rect" coords="100,138,163,159" href="http://www.heidelberg.de/"
alt="Heidelberg">
  <!-- -etc. -->
</map>
<p></p>
```

Vite à la ville ou à la région de votre choix!



Clic

Stadt Heidelberg

Startseite Suchen Inhaltsverzeichnis Hilfe Barrierefrei

Bürgerservice  
Politik und Verwaltung  
Stadtfinos  
Kultur und Freizeit  
Tourismus  
Menschen  
Bildung, Wissenschaft und Forschung  
Planen, Bauen, Wohnen

## Willkommen in Heidelberg.de

### Tag drei der „Heidelberger Woche der Energie“

Am Mittwoch, 10. September, finden wieder viele attraktive Ve  
Rahmen ihrer Kampagne „Klima sucht Schutz in Heidelberg – a  
September, veranstaltet.  
[mehr zum Thema...](#)

Source : [http://fr.selfhtml.org/html/graphiques/compose\\_liens.htm](http://fr.selfhtml.org/html/graphiques/compose_liens.htm)

# HTML – Les balises DIV et SPAN

## ■ Mécanisme général de groupement d'éléments et d'augmentation de la structure

- DIV pour "block-level"
- SPAN pour "inline"

```
<DIV id="section1">
```

```
  <H1>...</H1>
```

```
  <P>...
```

```
</DIV>
```

```
<DIV id="section2">
```

```
  <H1>...</H1>
```

```
  <P>...
```

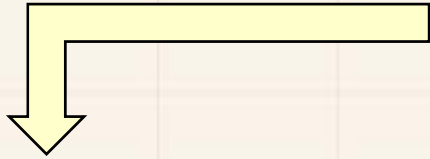
```
</DIV>
```

## ■ Pas de sémantique, utilisable avec les styles

Source: <http://www-adele.imag.fr/~donsez/cours/html.pdf>

➔ Très utilisées avec les feuilles de style (CSS)

# HTML – Les balises DIV et SPAN : Exemples (1/3)



```
<div align="center">  
  <h1>Tout centré</h1>  
  <ul>  
    <li>tout centré</li>  
    <li>tout?</li>  
    <li>tout!</li>  
  </ul>  
</div>
```

```
<div align="right">  
  <h1>Tout contre la marge droite</h1>  
  <ul>  
    <li>tout contre la marge  
droite</li>  
    <li>tout?</li>  
    <li>tout!</li>  
  </ul>  
</div>
```

**Tout centré**

Tout centré  
tout?  
tout!

**Tout contre la marge droite**

tout contre la marge droite  
tout?  
tout!

# HTML – Les balises DIV et SPAN : Exemples (2/3)

Application d'un style élaboré sur la balise SPAN :

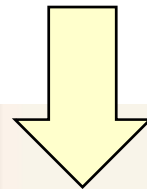
```
<span style="border:solid 1px black; background:lightsteelblue; width:300; display:block;">  
  <table width="95%" cellpadding="0" cellspacing="0">  
    <tr><td align="center">Ligne 1</td></tr>  
    <tr><td align="left">Ligne 2</td></tr>  
    <tr><td align="right">Ligne 3</td></tr>  
    <tr><td align="center">Ligne 4</td></tr>  
  </table>  
</span>
```

	Ligne 1	
Ligne 2		
		Ligne 3
	Ligne 4	

# HTML – Les balises DIV et SPAN : Exemples (3/3)

```
<html>
<body>
<p>C'est un paragraphe SANS SPAN <span style="color:#0000FF;">C'est un paragraphe AVEC SPAN</span> C'est un paragraphe SANS SPAN</p>
<p>
<span style="color:#00DD45;">
C'est un autre pragraphe AVEC SPAN
</span>

<p>C'est un paragraphe SANS DIV <div style="color:#0000FF;">C'est un paragraphe AVEC DIV</div> C'est un paragraphe SANS DIV</p>
<p>
<div style="color:#00DD45;">
C'est un autre paragraphe AVEC DIV
</div>
</p>
</body>
</html>
```



C'est un paragraphe SANS SPAN C'est un paragraphe AVEC SPAN C'est un paragraphe SANS SPAN

C'est un autre pragraphe AVEC SPAN

C'est un paragraphe SANS DIV

C'est un paragraphe AVEC DIV

C'est un paragraphe SANS DIV

C'est un autre paragraphe AVEC DIV

SPAN → Dans le texte

≠

DIV → Saut de ligne

- Clé de toute mise en page (p. ex aussi tabulations)
  - Élément `<table>`
  - Légende de tableau `<caption>`
  - Nouvelle ligne de tableau (table row) `<tr>`
  - Nouvelle cellule de tableau (table data) `<td>`
  - Cellules de titre (table head) `<th>`
- **On préfère maintenant utiliser les feuilles de style CSS pour réaliser des mises en page évoluées**

# HTML – Les tableaux : Exemples

```
<TABLE border="1">
<TR><TD>1 <TD>2 <TD>3
<TR><TD colspan="2">4 <TD>6
<TR><TD>7 <TD>8 <TD>9
</TABLE>
```

1	2	3
4	6	
7	8	9

```
<TABLE border="1">
<CAPTION>
<EM>A test table with merged cells</EM>
</CAPTION>
<TR><TH rowspan="2"><TH colspan="2">Average
<TH rowspan="2">Red<BR>eyes
<TR><TH>height<TH>weight
<TR><TH>Males<TD>1.9<TD>0.003<TD>40%
<TR><TH>Females<TD>1.7<TD>0.002<TD>43%
</TABLE>
```

*A test table with merged cells*

	Average		Red eyes
	height	weight	
Males	1.9	0.003	40%
Females	1.7	0.002	43%

```
<table>
<caption>Ventes</caption>
<tr><th>Mois</th><th>CH</th>
<th>UE</th><th>Autres</th></tr>
<tr><th>Janvier</th><td>3,2</td>
<td>4,8</td><td>5,4</td></tr>
...
</table>
```

Ventes  
Mois CH UE Autres  
Janvier 3,2 4,8 5,4

# HTML – L'élément `<table>`

- `border` → taille de bordure tableau en pixels
- `width` → largeur du tableau en pixels ou %
- `align` → alignement du tableau
- `cellspacing` → espace entre cellules, pixels
- `cellpadding` → espace à l'intérieur cellules
- `bgcolor` → couleur de fond



# HTML – Alignement des cellules

- Dans `<tr>` → alignement pour toute la ligne
- Dans `<td>` `<th>` → alignement pour une cellule
- Alignement horizontal `align`
  - `left`, `center`, `right`, `justify`
  - `align="char" char="."` → valeur numérique, alignement sur point
- Alignement vertical `valign`
  - `top`, `middle`, `bottom`
  - `baseline` tous les contenus alignés sur la même ligne de base

# HTML – Fusionner des cellules

- Pratique pour en-têtes
- Attribut `colspan` dans `<td>` ou `<th>` → étendue de la colonne (enlever x cellules dans la ligne pour garder même nombre de colonnes dans tout le tableau)
- Attribut `rowspan` dans `<td>` ou `<th>` → étendue de la ligne (les lignes suivantes doivent avoir x cellules en moins)
- Il est facile de se perdre pour étendre correctement les données (v. exemple suivant)

# HTML – Les tableaux : Exemple récapitulatif

```
<table width="80%" border="1" cellpadding="6"
  align="center">
<caption><h4>Faculté des Lettres</h4></caption>
<tr><th rowspan="2">Cours</th><th
  colspan="2">Enseignant</th></tr>
<tr><th>Nom</th><th>Bureau</th></tr>
<tr><td>NTIC</td><td>Luka
  Nerima</td><td>L705B</td></tr>
<tr><td>IA</td><td>Christian
  Pellegrini</td><td>U4XX</td></tr>
</table>
```


## Faculté des Lettres

Cours	Enseignant	
	Nom	Bureau
NTIC	Luka Nerima	L705B
IA	Christian Pellegrini	U4XX

- Pour interaction avec l'utilisateur
- Programme sur le serveur (ou script JavaScript) → retourne une page de résultats
- Common Gateway Interface CGI → programme sur le serveur qui traite des données
- Formulaires → forum, écrire mail, CQM, moteur de recherche etc.)

- Balise `<form>` qui peut contenir du texte et/ou les champs
  - `action` lien vers l'URI du CGI qui traite du formulaire
  - `method` manière de transmettre l'info
    - `get` → le contenu des formulaires est ajouté à l'URI spécifié dans `action`  
(`chercher.cgi?nom=Toto`)
    - `post` → le contenu est posté dans le corps de la requête http

# HTML – Les champs (ou contrôles)

- A inclure dans `<form>`
- Pratiquement tous `<input />`
  - `type="text"`
  - `type="password"`
  - `type="file"` 
- `name=""` → nom unique qui sera transmis
- `value=""` → valeur par défaut
- `size=""` → largeur du champ
- `maxlength=""` → longueur maximale de l'input

# HTML – Cases à cocher et boutons radio

- `<input name="xx" type="checkbox" value="yy" />`
  - `checked="checked"` optionnel
  - plusieurs contrôles avec même nom possible mais pas même valeur
  - n'est envoyé au CGI que si sélectionné
- `<input name="xx" type="radio" value="yy" />`
  - plusieurs valeurs nécessaires avec même nom
  - `checked="checked"` optionnel mais recommandé (1er de la série par défaut)

# HTML – Les boutons

- **Envoi du formulaire** `<input type="submit" value="Envoyer">`
  - `name` optionnel, envoie le contenu de `value`
- **Effacement** `<input type="reset" value="Effacer ...">`
- **Image équivalent à submit** `<input type="image" src="" alt="">`
- **Autres boutons** `<input type="button" value="Effacer ..." onclick="">`
  - nécessaire de définir action javascript



# HTML – Soumettre des valeurs cachées

- Valeurs à transmettre que l'utilisateur n'a pas besoin/droit de modifier
- `<input type="hidden" name="xx" value="yyy" />`
- ```
<form method="GET"
action="http://www.google.com/search"
target="google">
<input type="text" name="q" size=20 maxlength="255"
value="" />
<input TYPE="hidden" name="hl" value="fr"/>
<input TYPE="hidden" name="as_dt" value="i" />
<input TYPE="hidden" name="as_sitesearch"
value="www.latl.unige.ch" />
<input type="submit" name="btnG" value="Recherche
Google" />
</form>
```

# HTML – Les sélecteurs de liste

- Pour sélectionner valeur (= ComboBox)  
`<select>`
  - `name=""`
  - `size=""` nombre d'éléments visibles en même temps
  - `multiple="multiple"` sélection multiple possible
- `<option>` inclut les options possibles
  - `value=""` valeur à transmettre (si absent, envoie la valeur incluse dans option `<selection`
  - `selected="selected"` → valeur par défaut

Exemple :

```
<select name="animal">  
  <option value="1">chat</option>  
  <option value="2">chien</option>  
</select>
```

# HTML – Les champs textes multilignes

- Textes longs, multilignes `<textarea>`
- `name=""` nom du champ
- `rows` et `cols` nombre de caractères pour dimension du champ
- on voit attribut `nowrap="nowrap"` ou `wrap= OFF|VIRTUAL| PHYSICAL` mais pas recommandé
- A l'intérieur du tag: texte à afficher par défaut
  - `<textarea name="toto">Ce texte est affiché dans le formulaire</textarea>`

# HTML – Formulaires : Exemples

```
<HTML><HEAD><TITLE>Formulaire</TITLE></HEAD>
<BODY><H1 align=center>Formulaire</H1><hr>
<FORM ENCTYPE="multipart/form-data" METHOD="POST" ACTION="/servlet/formprocess">
Ligne de Texte : <INPUT TYPE=text VALUE=" A Remplir" NAME=lig>
Zone de Texte : <TEXTAREA NAME=zon COLS=40 ROWS=5>
A Remplir
</TEXTAREA>
<INPUT TYPE=checkbox VALUE="Opt1" NAME=opt> Option 1
<INPUT TYPE=checkbox VALUE="Opt2" NAME=opt> Option 2<BR>
<INPUT Type=submit Value="Envoi">
<INPUT Type=reset Value="Reinit">
</FORM>
</BODY></HTML>
```

```
<FORM action="http://somesite.com/prog/adduser" method="post">
<P>
First name: <INPUT type="text" name="firstname"><BR>
Last name: <INPUT type="text" name="lastname"><BR>
email: <INPUT type="text" name="email"><BR>
<INPUT type="radio" name="sex" value="Male"> Male<BR>
<INPUT type="radio" name="sex" value="Female"> Female<BR>
<INPUT type="submit" value="Send"> <INPUT type="reset">
</P>
</FORM>
```

First name:

Last name:

email:

Male

Female

# HTML – Les objets externes (1/2)

## ■ Inclusion de « documents » externes

```
<OBJECT data="TheEarth.mpeg"
        type="application/mpeg">
    Votre browser ne visualise pas les films MPEG.
</OBJECT>
```

## ■ Objets imbriqués : suite d'objets alternatifs

```
<OBJECT title="The Earth as seen from space"
        classid="http://www.observer.mars/TheEarth.py">
    <!-- Else, try the MPEG video -->
    <OBJECT data="TheEarth.mpeg" type="application/mpeg">
        <!-- Else, try the GIF image -->
        <OBJECT data="TheEarth.gif" type="image/gif">
            <!-- Else render the text -->
            The <STRONG>Earth</STRONG> as seen from space.
        </OBJECT>
    </OBJECT>
</OBJECT>
```

# HTML – Les objets externes (2/2)

## ■ Inclusion de contrôles ActiveX et d'applets (appliquette)

```
<OBJECT id="clock1"
  classid="clsid:663C8FEF-1EF9-11CF-A3DB-080036F12502"
  data="data:application/x-oleobject;base64, ...base64 data...">
  A clock.
</OBJECT>
```

```
<OBJECT codetype="application/java"
  classid="AudioItem"
  width="15" height="15">
  <PARAM name="snd" value="Hello.au|Welcome.au">
  Java applet that plays a welcoming sound.
</OBJECT>
```

## ■ Ancienne forme (*non recommandée*)

```
<APPLET code="AudioItem" width="15" height="15">
  <PARAM name="snd" value="Hello.au|Welcome.au">
  Java applet that plays a welcoming sound.
</APPLET>
```

# HTML – Les cadres (frames)

- Visualisation de plusieurs documents dans différents cadres (frames) dans la page du butineur
- Les actions dans un cadre peuvent agir sur un autre cadre

## ■ Exemple

```
<FRAMESET COLS="100, *">  
  <FRAME SRC="docgauche.htm">  
  <FRAMESET ROWS="25%,50%,*">  
    <FRAME SRC="docdroitehaut.htm">  
    <FRAME SRC="docdroitemilieu.htm">  
    <FRAME SRC="docdroitebas.htm">  
  </FRAMESET>  
</FRAMESET>  
<NOFRAMES>  
  Document non supporté par votre navigateur  
</NOFRAMES>
```

# HTML – Les éléments des frames HTML

- FRAMESET

- COLS

- le frameset est composé de colonnes  
donne la largeur de chaque colonne

- ROWS

- le frameset est composé de lignes  
donne la hauteur de chaque ligne

- FRAME

- SRC

- document à afficher dans le cadre

- NAME

- nomme un cadre

- TARGET

- désigne le cadre qui affichera le document demandé

- nomdecadre*, `_BLANK`, `_PARENT`, `_TOP`, `_SELF`

- Remarque : l'élément `BASE` peut avoir un attribut `TARGET`

- `<BASE href="http://www.mycom.com/Slides" target="dynamic">`



# HTML – Définition du jeu de cadres (1/2)

- Un document `<html><head>`, extension `.htm(l)` comme le reste
- Pas `<body>` mais `<frameset>`
- Séparation verticales définies par attribut `col`, horizontales par `rows`
  - `<frameset cols="100,*,20%">`
  - valeurs séparées par virgule, cadres de haut en bas ou gauche à droite
  - soit pixels, soit pourcentage, soit `*` → le reste
- Il faut un message alternatif pour les anciens browsers qui ne peuvent afficher des cadres

```
<noframe><body>Votre navigateur  
ne supporte pas les  
frames</body></noframe>
```

## HTML – Définition du jeu de cadres (2/2)

- Contenu par défaut au chargement de la page
- A l'intérieur de `<frameset>`, dans l'ordre des différents cadres (haut/bas ou gauche/droite)
  - Soit un second `<frameset>` pour séparer dans l'autre sens
  - Soit lien vers document avec `<frame name="" src="">`
  - `frameborder = 1` ou `0` → ligne séparation du cadre
  - `noresize` empêche modification de la taille
- Le nom du cadre sert à définir destination d'un lien
- La page liée est soit un doc html normal, soit une nouvelle définition de cadres

# HTML – Les cibles (*target*) dans les frames (1/2)

- Si rien n'est spécifié, nouvelle page affichée dans son cadre d'origine
- Attribut `target` pour spécifier la cible (cadre)
  - `target="right"` → affiche dans le cadre qui a été défini par `<frame ... name="right">`
  - Cas particuliers
    - `target="_blank"` → nouvelle fenêtre navigateur
    - `target="_self"` → même fenêtre, même cadre
    - `target="_parent"` → cadre niveau supérieur
    - `target="_top"` → même fenêtre dans la totalité
- `<base target="">` dans `<head>` du document → donne cible par défaut

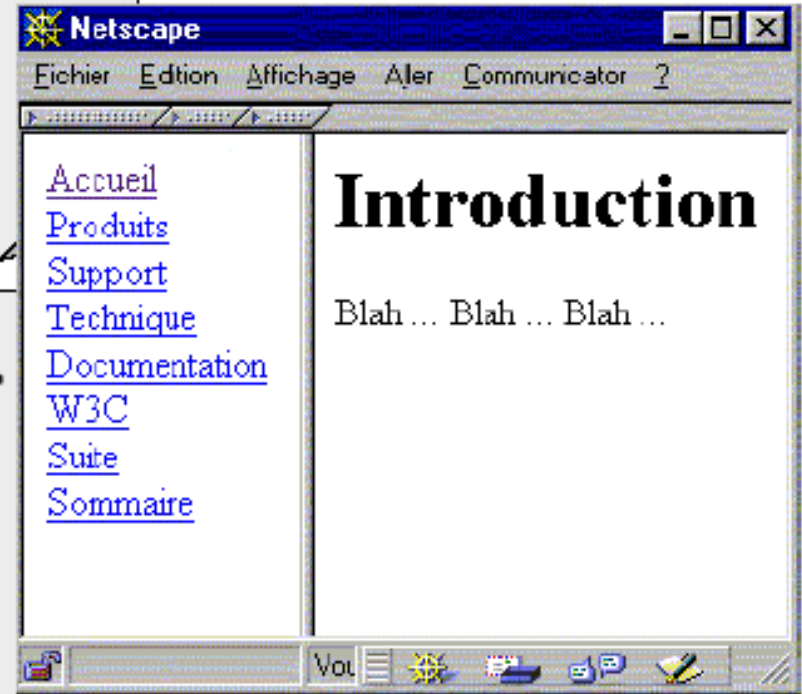
# HTML – Les cibles (target) dans les frames (2/2)

*indexframe.htm*

```
<FRAMESET COLS="100, *">  
  <FRAME SRC="som1.htm" NAME="fr1">  
  <FRAME SRC="intro.htm" NAME="fr2">  
</FRAMESET>
```

*som1.htm*

```
<HTML><HEAD><BASE TARGET="fr2"></HEAD>  
<BODY>  
  <A HREF="intro.htm">  
    Accueil</A><BR>  
  <A HREF="produit.htm" TARGET="fr2">  
    Produits</A><BR>  
  <A HREF="sommaire.htm" TARGET="_top">  
    Support Technique</A><BR>  
  <A HREF="http://www.w3c.org/TR" TARGET="_self">  
    Documentation W3C</A><BR>  
  <A HREF="som2.htm" TARGET="fr1">  
    Suite Sommaire</A><BR>  
</BODY></HTML>
```



# HTML – Les IFrames (Inline Frame)

*texteinclus.html*

```
<HTML><BODY>  
  <H1>Texte inclus</H1>  
</BODY></HTML>
```

*doc.html*

```
<HTML><BODY>  
...  
<IFRAME src="texteinclus.html" width="400" height="500"  
  scrolling="auto" frameborder="1">  
  [Votre visualisateur ne supporte pas les frames.  
  Cependant, vous pouvez visiter  
  <A href= "texteinclus.html"> le document relatif.</A>]  
</IFRAME>  
...  
</BODY></HTML>
```

# HTML – Les IFrames : Exemple

## Regarder par le carreau

Voyez donc SELFHTML par la fenêtre:



```
<html>
<head>
  <title>Cadres incorporés</title>
</head>
<body>
  <h1>Regarder par le carreau</h1>
  <p>Voyez donc SELFHTML par la fenêtre:</p>
  <iframe src="../../../index.htm" name="SELFHTML_in_a_box"
width="90%"height="400">
    <p>Votre navigateur ne peut malheureusement pas afficher de cadre
incorporé: Vous pouvez appeler la page incorporée par ce lien:
<a href="../../../index.htm">SELFHTML</a></p>
  </iframe>
</body>
</html>
```

# ***CSS – Cascading Style Sheets***

# Cascading Style Sheets (CSS) (1/2)

## ■ Feuilles de style en cascade

- ✓ Regrouper dans un même document des caractéristiques de mise en forme associées à des groupes d'éléments.
- ✓ Définition de plusieurs styles possibles et héritage des styles en cascade (Cascading).
- ✓ Fournir une plus grande richesse d'éléments de style graphique afin d'améliorer l'apparence des documents HTML.

## ■ Les feuilles de style permettent :

- ✓ d'obtenir une présentation homogène sur tout un site en faisant appel sur toutes les pages à **une même définition de style** ;
- ✓ de permettre le changement de l'aspect d'un site complet entier par **la seule modification de quelques lignes** ;
- ✓ une **plus grande lisibilité du HTML**, car les styles sont définis à part ;
- ✓ des **chargements de page plus rapides**, pour les mêmes raisons que précédemment ;
- ✓ un **positionnement plus rigoureux des éléments**.



# Cascading Style Sheets (CSS) (2/2)

- Trois niveaux de CSS
  - ✓ CSS Level 1, CSS Positionning, CSS 2
- Cascading Style Sheets CSS 2.1 pour HTML et XML
  - ✓ <http://www.w3.org/TR/CSS21/>
  - ✓ Nouvelle spécification en cours : CSS 3
- Support de CSS :
  - ✓ Microsoft Internet Explorer 4.x, 5.x, 6.x, etc.
  - ✓ Firefox x.x
  - ✓ Netscape Navigator 4.x, 6.x, 7.x
  - ✓ Mozilla x.x
  - ✓ Opera 5.x, 6.x, 7.x
  - ✓ Camino 0.8x
  - ✓ Safari 1.xx

# CSS – Incorporation de CSS dans HTML (1/2)

## ■ Feuille de style interne (ou style de document) :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <STYLE type="text/css">
      <!-- Définition des styles; -->
    </STYLE>
  </HEAD>
  <BODY></BODY>
</HTML>
```

## ■ Feuille de style externe :

```
<HTML>
  <HEAD>
    <LINK rel="stylesheet" type="text/css"
href="style.css">
  </HEAD>
  ...
```

# CSS – Incorporation de CSS dans HTML (2/2)

## ■ Feuille de style en ligne

```
<HTML>
  <HEAD> ... </HEAD>
  <BODY> ...
    <H1 style="Font: 18px Verdana; font-weight:bold;"> Titre
  </H1>
  ...
</BODY>
</HTML>
```

### ✓ Balises non concernées par l'application de styles

- *BASE, BASEFONT, HEAD, HTML, META, PARAM, SCRIPT, STYLE, TITLE*

## ■ Feuille de style importé

```
<HTML>
  <HEAD>
    <STYLE type="text/css">
      <!--
        @IMPORT URL(url de la feuille à importer);
      -->
    </STYLE>
  </HEAD>
  ...
```

# CSS – Définition de styles externes et cascadés

*styl1.css*

```
H1.cs { color : blue; text-align: center }
```

*styl2.css*

```
@import "styl1.css";  
P.cs { color : green; border: solid red; }
```

*spec1.htm*

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"  
    "http://www.w3.org/TR/REC-html40">  
<HTML>  
  <HEAD>  
    <LINK href="styl2.css" rel="stylesheet" type="text/css">  
  </HEAD>  
  <BODY>  
    <H1 class="cs">Titre en bleu</H1>  
    <P class="cs">Paragraphe en vert.  
    <P>Paragraphe Normal.  
  </BODY>  
</HTML>
```

# CSS – SPAN et DIV : Rappel

- Appliquer des styles à des paragraphes ou des morceaux de paragraphes.

- ✓ La balise SPAN

- Permet de modifier des morceaux de paragraphes

```
<SPAN class=Nom_de_la_classe> Texte </SPAN>
```

- ✓ La balise DIV

- Permet de modifier des blocs de paragraphes

```
<DIV class=important> paragraphes </DIV>
```

- Exemple

```
<span style= "color: green ; font-family: Times, sans-serif ; font-size: 14p  
; " >Vous constatez </span> que ce texte n'est pas totalement <span style=  
"color: green ; font-family: Times, sans-serif ; font-size: 14pt ; ">en vert  
. </span>
```

Vous constatez que ce texte n'est pas totalement en vert .

# CSS – *Styles en cascade*

- Plusieurs feuilles peuvent être associées à un même document
- Plusieurs définitions de style peuvent affecter le même tag dans la même feuille
  - `h1{color:red;} p{...} ... h1{font-size:20pt}`  
équivalent à `h1{color:red; font-size:20pt}`
- La dernière valeur spécifiée écrase les précédentes

# CSS – Styles en cascade : Exemple

```
<HTML>
  <HEAD>
    <STYLE type="text/css">
      <!--
        h1 {font-family: verdana, arial, sans-serif ;
          background-color: #FFFFFF ;
          color: green ; font-size:12pt ; }
      -->
    </STYLE>
  </HEAD>
  <BODY>
    <DIV style = "font-family: verdana, arial, sans-serif ; color:
maroon ; font-size:10pt ; font-size:16pt ;background-color: #FFFFFF ;
">
      <H1>Ceci est un test en cascade</H1>
    </DIV>
    <H1>
      <DIV style = " font-family: verdana, arial, sans-serif ; color:
maroon ; font-size:10pt ; font-size:16pt ;background-color: #FFFFFF ;
">
Ceci est un test en cascade</DIV>
    </H1>
  </BODY>
</HTML>
```

Ceci est un test en cascade

**Ceci est un test en cascade**

# CSS – Plusieurs feuilles de style

- Plusieurs feuilles associées au document
  - La dernière définition d'un tag l'emporte sur les autres
    - ```
<link rel=stylesheet type="text/css" href="style1.css">  
<link rel=stylesheet type="text/css" href="style2.css">  
<link rel=stylesheet type="text/css" href="style3.css">
```
- Styles alternatifs → sélectionnables suivant le browser
  - ```
<LINK rel="alternate stylesheet" type="text/css"  
href="style1.css" title="style1">  
  <LINK rel="alternate stylesheet" type="text/css"  
href="style2.css" title="style2">  
  <LINK rel=stylesheet type="text/css"  
href="stylepardefaut.css">
```



# CSS – Notion d'héritage

- Un élément X contenu dans un élément Y va hériter des propriétés de cet élément Y !!
- Exemple :

## Héritage : CSS

```
.niveau1 {
    color:#0033FF; /* bleu */
    text-align:right;
    font-weight:bold;
    font-family:Verdana, Arial, Helvetica, sans-serif;
}
.niveau2 {
    text-decoration:underline;
    color:#CC3366; /* violet */
}
```

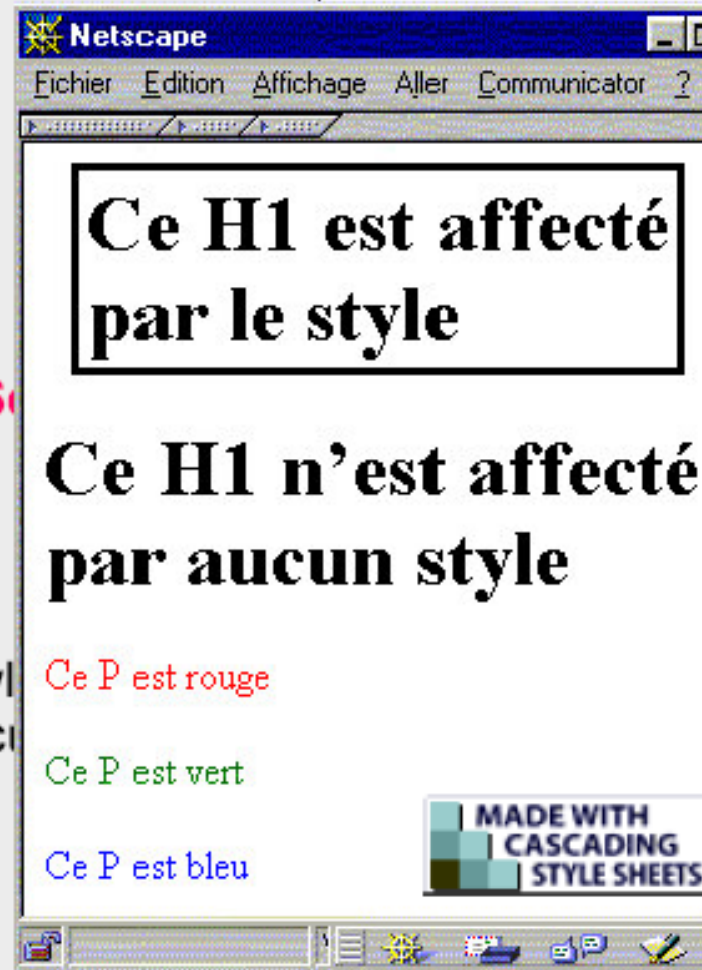
## Héritage : HTML

```
<div class="niveau1">
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
    <div class="niveau2">
        In tempor quam nec enim sollicitudin vehicula.
    </div>
    Mauris lacus. Aenean odio ligula, mollis imperdiet, elementum non, gravida et, leo.
</div>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
 In tempor quam nec enim sollicitudin vehicula.  
 Mauris lacus. Aenean odio ligula, mollis imperdiet, elementum non, gravida et, leo.

# CSS – L'élément STYLE

```
<HEAD>
  <STYLE type="text/css">
    H1.myclass { border-width: 1; border: solid;
                text-align: center}
    P.special { color : green; }
  </STYLE>
  <STYLE TYPE="text/javascript">
    classes.myclass.P.color="red"; /* exprimé en JavaS
    document.tags.P.color = "blue";
  </STYLE>
</HEAD>
<BODY>
  <H1 class="myclass"> Ce H1 est affecté par le styl
  <H1> Ce H1 n'est affecté par auc
  <P class="myclass"> Ce P est rouge </P>
  <P class="special"> Ce P est vert </P>
  <P> Ce P est bleu </P>
</BODY>
```





## ■ Sélecteur de classe

- `H1.nouveau { color: #FF0000 }`
- `<H1 class="nouveau">Tout Nouveau !</H1>`
- `.nouveau { color: #FF0000 }`
  - Sélection de tous les éléments de même classe:

## ■ Sélecteur ID

- `#intro { letter-spacing: 0.3em }`
- `<H1 ID="z98y">Introduction</P>`
- `H1#intro { letter-spacing: 0.3em }`

## ■ Sélecteur contextuel

- `H1 EM { color: red }`
  - Affecte uniquement les éléments EM dans un H1
- `UL LI { font-size: small }`
- `UL UL LI { font-size: x-small }`

## ■ Mixage de sélecteurs

- `.reddish H1 { color: red }`
- `#x78y CODE { background: blue }`
- `DIV.sidenote H1 { font-size: large }`
- `H1 B, H2 B, H1 EM, H2 EM { color: red }`

# CSS – Les classes

- Définition de classes pour appliquer des styles différents aux mêmes balises

- Une classe

- ✓ Définie après le nom d'une balise suivie d'un point (".") :

```
B.rouge {font: Verdana 12px; color: #FF0000; }
```

- ✓ Appel d'une classe

```
<B class="Rouge"> Texte à mettre en rouge et en gras </B>
```

- La classe universelle

- ✓ Définie pour être utilisée dans n'importe quelle balise.

```
.important {font-type: arial; color: red; font-weight: bold}
```

- ✓ Appel de la classe :

```
<h1 class="important">Attention ceci est un  
avertissement</h1>
```

```
<i class="important">(prière d'en tenir compte)</i>
```

# CSS – Les pseudo-classes (1/3)

## ■ Les pseudo-classes

- ✓ Affiner le style appliqué à un certain nombre de balises en définissant une réaction à un événement ou bien à la position relative de la balise au sein des autres balises.
- ✓ Ces pseudo-classes sont prédéfinies, contrairement aux classes

## ■ Plusieurs types de pseudo-classes

- ✓ Les pseudo-classes dynamiques
- ✓ Les pseudo-classes de lien
- ✓ Les pseudo-classes de langue
- ✓ Les pseudo-classes first-child
- ✓ Les pseudo-classes de page
- ✓ Les pseudo-classes de texte

# CSS – Les pseudo-classes (2/3)

## ■ Les pseudo-classes dynamiques

- ✓ Modifier le style d'une balise en fonction d'un événement comme un clic souris
- ✓ 3 pseudo-classes : **:hover** (survol avec souris), **:focus** (i.e. clic dans un formulaire), et **:active** (entre le début d'un clic souris et le moment où on relâche)
- ✓ Exemple : **A:hover** {font-decoration: underline;}

## ■ Les pseudo-classes de lien

- ✓ Modifier le style d'un lien (balise <A>)
- ✓ 2 pseudo-classes : **:link** - style non-consulté-, **:visited** - style consulté-.

## ■ Les pseudo-classes de langue

- ✓ Définir un style en fonction de la langue d'un document ou d'un élément HTML (spécifié grâce à l'attribut optionnel LANG).
- ✓ 1 pseudo-classe : **:lang(Langue)**
- ✓ Exemple: **HTML:lang(fr)** { quotes: '« ' ' »' }



# CSS – Les pseudo-classes (3/3)

## ■ La pseudo-classe descendante

- ✓ Appliquer un style à la première balise au sein d'un élément en utilisant le symbole ">" et la pseudo-classe **:first-child**
- ✓ Exemple: `P > A:first-child {color: #00FF00;}`

## ■ Les pseudo-classes de page

- ✓ Le sélecteur **@page** permet de modifier la mise en page d'une page HTML.
- ✓ 3 pseudo-classes : **:left** (les pages de gauche), **:right** (les pages et de droite), et **:first** (la première page).
- ✓ Exemple:

```
@page:right { size:landscape; margin-left: 2.5cm; }
```

## ■ Les pseudo-classes de texte

- ✓ Appliquer un style à une partie d'un texte délimité par des balises (<P>).
- ✓ 2 pseudo-classes: **:first-line** (la première ligne du texte) et **:first-letter** (la première lettre du texte)
- ✓ Exemple:

```
P:first-letter { font-size: 200%; font-weight: bold; }
```

## ■ Relatives

- em → taille proportionnelle à la taille de la police utilisée
- ex → taille relative à la minuscule de la police
- px → pixels, taille relative à l'appareil (écran, projecteur, imprimante...)

## ■ Absolues

- in, cm, mm → pouces, centimètres, millimètres
- pt → point, 1/72 pouce
- pc → pica, 12 points

# CSS – Les couleurs

## ■ Différentes façons pour définir les couleurs par :

### ✓ Un **nom**.

- 17 couleurs
- Aqua, green, orange, white, black, lime, purple, yellow, blue, maroon, red, fuschia, navy, silver, gray, olive, teal.
- `h1 {color: blue}`

### ✓ Un **code RGB** (ou RVG) en notation **hexadécimal**.

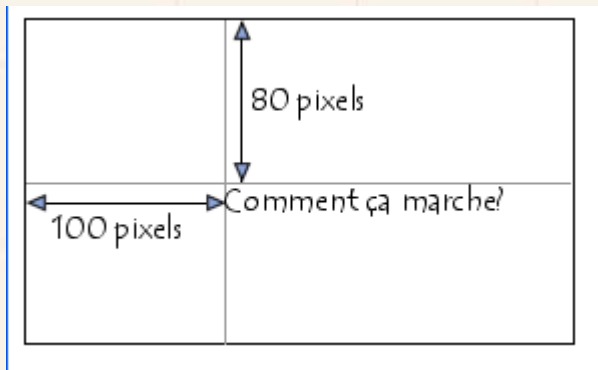
- 16 millions de couleurs avec RGB.
- #RRGGBB : RR, GG, BB = chacun est égal à un nombre hexadécimal entre 00 et FF pour le rouge (Red), le vert (Green) et le bleu (Blue)
- `h1 {color: #0000FF}`

### ✓ Un **code RGB** en notation **décimal**.

- Proportion relative en pourcentage (%) ou en notation entière absolue (entre 0 et 255)
- `rgb(100%,100%,100%)`
- `rgb(24,125,255)`

# CSS – Le positionnement (1/3)

- Avec les balises SPAN et DIV du langage HTML, il est possible de positionner au pixel près du texte et des images.
  - ✓ Positionnement relatif (par rapport à d'autres éléments)
  - ✓ Positionnement absolu (coin supérieur gauche de la fenêtre du navigateur)
  - ✓ Exemple 1 : positionnement d'un texte



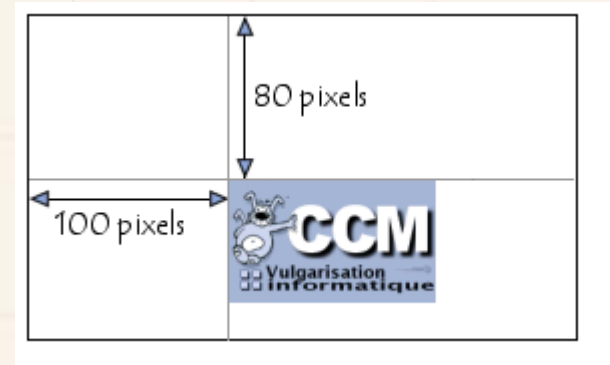
```
<HTML>
  <BODY>
    <SPAN style="position: absolute;
top: 80 px; left: 100 px;">
Comment ça marche? </SPAN>
  </BODY>
</HTML>
```

Source: <http://www.commentcamarche.net/css/>

# CSS – Le positionnement (2/3)

## ✓ Exemple 2 : positionnement d'une image

```
<HTML>
<HEAD>
  <STYLE>
    <!--
      .test{position: absolute;
top: 80px; left: 100px; width:
103px; height: 61px"}
    -->
  </STYLE>
</HEAD>
<BODY>
  <DIV class=test>
    <IMG SRC="test.jpg" >
  </DIV>
</BODY>
</HTML>
```

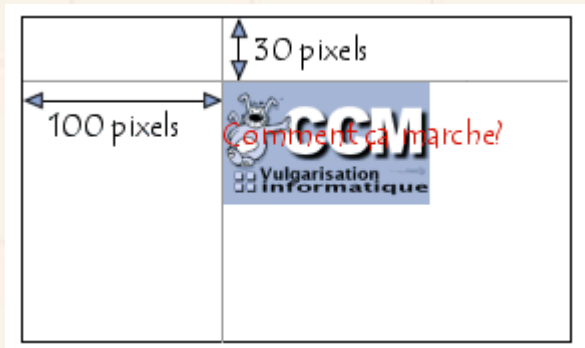


Source: <http://www.commentcamarche.net/css/>

```
<HTML>
<BODY>
  <SPAN style="position:
absolute; top: 80 px; left: 100
px; width: 103px; height: 61px">
    <IMG SRC="test.jpg" >
  </SPAN>
</BODY>
</HTML>
```

# CSS – Le positionnement (3/3)

## ✓ Exemple 3 : superposition d'éléments



Source: <http://www.commentcamarche.net/css/>

```
<HTML>
  <BODY>
    <SPAN style="position: absolute;
top: 30 px; left: 100px;width: 103px;
height: 61px">
      <IMG SRC="test.jpg" >
    </SPAN>
    <SPAN style="position: absolute;
top: 50 px; left: 100 px;"> Comment ça
marche? </SPAN>
  </BODY>
</HTML>
```

# CSS – Quelques propriétés

## ■ Quelques propriétés

- ✓ pour les polices de caractères
  - font-family: Arial ; font-size: x-large; font-style: bold; ...
- ✓ pour les textes et les paragraphes
  - text-align: center; color: red; text-indent: 5px, width: 640; height: 480; ...
- ✓ pour les couleurs et l'arrière-plan
  - background-color: yellow; background-position: left;...
- ✓ pour les bordures
  - border-left-color: #0000FF; border-top-width: 3pt; ...
- ✓ pour les espaces intérieurs
  - padding-top: 3px; padding-right: 0.25em; ...
- ✓ pour les tableaux
  - border-spacing: 4px; caption-side: top; table-layout: fixed; ...
- ✓ pour les listes
  - list-style-type: circle; list-style-position: inside; ...
- ✓ pour la mise en page
  - @page(size: portrait); margin-left: 2 cm; page-break-before: always, ...

# ***DOM et Script côté client***

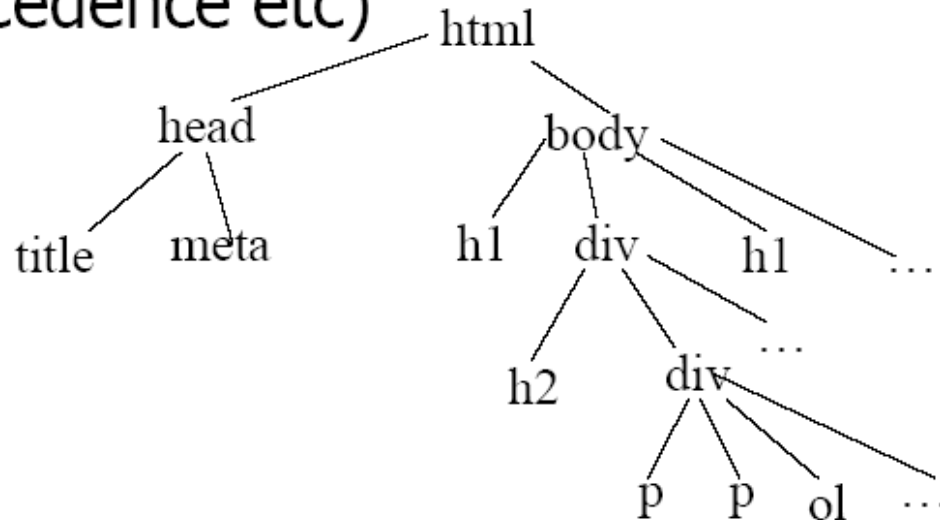


# Document Object Model (DOM) (1/2)

- Description d'un document XML (en donc HTML) sous la forme d'un objet composite
  - chaque élément (textes, graphiques, headlines, styles) est un sous objet lui-même composite.
  - un script (*VBScript*, *JavaScript*, *ECMAScript* ...) embarqué dans le document peut manipuler cet objet composite.
- API standard d'accès à un document et son style
  - le document est vu comme une arbre
  - chaque objet est un nœud de l'arbre et possède un interface définition avec l'IDL CORBA2.2
    - « binding » avec Java et ECMAScript (JavaScript, JScript)

# Document Object Model (DOM) (2/2)

- Document représenté sous forme arbre  
→ notion de hiérarchie (mère/enfant, précédence etc)



## ■ DOM Level 1

- Accès et manipulation de la structure d'un document XML ou HTML
  - <http://www.w3.org/TR/REC-DOM-Level-1>

## ■ DOM Level 2

- CSS, Range, Filtres et Iterateurs, Evénements utilisateur et de transformation, Namespaces
  - <http://www.w3.org/TR/WD-DOM-Level-2>

# Script côté client (1/2)

## ■ But : Moins d'intervention du Serveur WWW

- Génération dynamique de documents HTML par le butineur
- Contrôle des formulaires HTML par le butineur
- Contrôle des ressources du butineur

## ■ Scripts embarqués dans un document HTML

- Mécanisme ouvert

```
<META http-equiv="Content-Script-Type" content="text/tcl">
```

```
<SCRIPT> ... </SCRIPT>
```

```
<SCRIPT LANGUAGE="JavaScript"> ... </SCRIPT>
```

```
<SCRIPT LANGUAGE="JavaScript" SRC="scrpt.js"> ... </SCRIPT>
```

# Script côté client (2/2)

## ■ Fonctionnalités

- Représentation objet des éléments du document
  - layer, form, input, img, ...
  - Manipulation des caractéristiques de ces éléments
  - REMARQUE : précurseur à DOM
- Programmation événementielle
  - rendre le document « dynamique »

## ■ Langages

- JavaScript (NS), JScript (MS), ECMAScript
  - rien à voir avec Java
- VBScript (MS)
  - syntaxe Visual Basic, dialogue avec les contrôles ActiveX

# Prochaines séances

- Prochain cours, le 22/09 :

  - Notions de Javascript

- Mercredi 17/09 :

  - 1<sup>er</sup> laboratoire – « HTML & CSS »