

# **Créer un Site Web en Langage HTML**



**Institut Supérieur des  
Sciences Et Techniques**

**Saint-Quentin**

**Formation au langage HTML**

Ralph Davidovits © Novembre 1996  
ralph.davidovits@insset.u-picardie.fr



# Table des matières

<b>1. BIBLIOGRAPHIE:</b> .....	<b>8</b>
<b>2. LOGICIELS RECOMMANDÉS POUR CRÉER DES PAGES WEB:</b> .....	<b>9</b>
2.1 SUR PC/WINDOWS:.....	9
2.2 SUR MAC:.....	9
<b>3. INTRODUCTION</b> .....	<b>11</b>
3.1 À QUOI SERT UN SERVEUR WEB? QUELLES INFORMATIONS Y PUBLIER ? QUELS OUTILS MATÉRIELS ET LOGICIELS (SERVEUR, CRÉATION) UTILISER SUR MAC, PC ET UNIX? LES PRIX?.....	11
3.2 RÉGIME JURIDIQUE DES PAGES, DU DESIGN DU SITE ET DES ÉLÉMENTS ATTACHÉS À CE SERVEUR.....	11
<b>4. LA CONSTRUCTION D'UN SERVEUR WEB</b> .....	<b>12</b>
4.1 INTRODUCTION:.....	12
4.2 L'URGENCE D'ÊTRE SUR LE WEB.....	12
4.3 LA COLLABORATION DES COMPÉTENCES POUR L'ÉDITION DE WEB.....	13
<b>5. LA PHASE PRÉPARATOIRE À LA CONCEPTION</b> .....	<b>14</b>
5.1 ÊTRE FAMILIER AVEC LE WEB.....	14
5.2 CONSULTER LES SITES WEB EXISTANTS.....	14
5.3 LES LIMITES ET CONTRAINTES DE LA PUBLICATION SUR INTERNET.....	15
<b>5.3.1 Comprendre ces contraintes et limitations techniques</b> .....	<b>15</b>
5.3.1.1 La vitesse de connexion.....	15
5.3.1.2 La plate-forme de l'ordinateur.....	16
5.3.1.3 Afficher des images et du texte en haute résolution.....	16
5.3.1.4 Le logiciel de consultation des pages Web.....	17
<b>5.3.2 Limitations et contraintes pour l'utilisateur</b> .....	<b>18</b>
5.3.2.1 Être perdu dans le Cyberespace.....	18
5.3.2.2 Consulter le Web opposé à la lecture de publications imprimées.....	18
<b>6. DÉTERMINER LES OBJECTIFS DU PROJET</b> .....	<b>20</b>
6.1 POURQUOI CRÉER UN SITE WEB?.....	20
6.2 DÉVELOPPER L'OSSATURE D'UNE ORGANISATION.....	20
<b>6.2.1 Le cadre organisationnel du Web</b> .....	<b>21</b>
6.2.1.1 Structurer l'information.....	21
6.2.1.2 Ajouter des liens dans la structure.....	22

6.2.1.3 Intégrer des outils de recherches.....	22
<b>6.2.2 Regrouper le contenu et les activités.....</b>	<b>23</b>
6.2.2.1 Utiliser la hiérarchisation pour organiser les groupes.....	23
6.3 ADAPTER LE SITE AUX BESOINS DE L'INTERNAUTE.....	25
6.4 CRÉER LE PROTOTYPE EN HTML.....	26
<b>6.4.1 La création de pages types.....</b>	<b>26</b>
6.4.1.1 La Page d'accueil.....	26
6.4.1.2 Les pages de navigations intermédiaires.....	27
6.4.1.3 Le contenu des pages.....	27
<b>7. LE LANGAGE HTML.....</b>	<b>28</b>
7.1 LES BASES DE LA PROGRAMMATION.....	28
<b>7.1.1 Faut-il encore apprendre HTML ?.....</b>	<b>28</b>
<b>7.1.2 L'hypertexte.....</b>	<b>29</b>
7.1.2.1 Le livre.....	29
7.1.2.2 Le document technique.....	29
7.1.2.3 Le journal.....	30
<b>7.1.3 Notions d'hypertexte.....</b>	<b>30</b>
7.2 HTML, SON ENVIRONNEMENT.....	31
<b>7.2.1 Qu'est-ce que HTML ?.....</b>	<b>31</b>
<b>7.2.2 Dans quel contexte s'exécute HTML ?.....</b>	<b>31</b>
<b>7.2.3 Le protocole d'adressage des documents: l'URL.....</b>	<b>32</b>
7.3 STRUCTURE.....	33
<b>7.3.1 Les balises.....</b>	<b>33</b>
<b>7.3.2 L'accentuation.....</b>	<b>34</b>
7.4 STRUCTURATION DU DOCUMENT HTML.....	35
<b>7.4.1 &lt;HTML&gt;.....</b>	<b>35</b>
<b>7.4.2 &lt;HEAD&gt;.....</b>	<b>35</b>
<b>7.4.3 &lt;TITLE&gt;.....</b>	<b>35</b>
<b>7.4.4 &lt;BASE&gt;.....</b>	<b>35</b>
<b>7.4.5 &lt;META&gt;.....</b>	<b>36</b>
<b>7.4.6 &lt;LINK&gt;.....</b>	<b>36</b>
<b>7.4.7 &lt;SCRIPT&gt;.....</b>	<b>37</b>
<b>7.4.8 &lt;BODY&gt;.....</b>	<b>37</b>
<b>7.4.9 &lt;ADDRESS&gt;.....</b>	<b>37</b>
<b>7.4.10 &lt;!-- Commentaires --&gt;.....</b>	<b>37</b>
<b>7.4.11 Attributs d'une balise.....</b>	<b>38</b>
<b>7.4.12 Squelette d'un document HTML.....</b>	<b>38</b>
<b>7.4.13 Tester les documents HTML.....</b>	<b>39</b>
7.5 DIVISIONS.....	39
<b>7.5.1 &lt;Hn&gt; &lt;/Hn&gt;.....</b>	<b>39</b>
<b>7.5.2 &lt;BR&gt;.....</b>	<b>40</b>
<b>7.5.3 &lt;P&gt;.....</b>	<b>40</b>
<b>7.5.4 &lt;DIV&gt; &lt;/DIV&gt;.....</b>	<b>41</b>
<b>7.5.5 &lt;CENTER&gt; &lt;/CENTER&gt;.....</b>	<b>41</b>
<b>7.5.6 &lt;HR&gt;.....</b>	<b>41</b>
<b>7.5.7 &lt;PRE&gt; &lt;/PRE&gt;.....</b>	<b>41</b>

7.6 LISTES.....	42
7.6.1 Liste descriptive.....	42
7.6.2 <DL> </DL>.....	42
7.6.2.1 <DT>.....	42
7.6.2.2 <DD>.....	43
7.6.3 Listes régulières.....	43
7.6.3.1 <LI>.....	43
7.6.3.2 <UL> </UL>.....	44
7.6.3.3 <OL> </OL>.....	44
7.7 STYLES.....	45
7.7.1 Style physique.....	45
7.7.1.1 <B> </B>.....	45
7.7.1.2 <I> </I>.....	45
7.7.1.3 <U> </U>.....	45
7.7.1.4 <TT> </TT>.....	45
7.7.1.5 <STRIKE> ou <S> </S>.....	46
7.7.1.6 <BIG> </BIG>.....	46
7.7.1.7 <SMALL> </SMALL>.....	46
7.7.1.8 <SUB> </SUB>.....	46
7.7.1.9 <SUP> </SUP>.....	46
7.7.1.10 <FONT> </FONT>.....	46
7.7.1.10.1 L'attribut SIZE.....	46
7.7.1.10.2 L'attribut COLOR.....	47
7.7.1.10.3 L'attribut BASEFONT.....	47
7.7.1.10.4 L'attribut FACE.....	47
7.7.1.11 La fonction &nbsp;.....	47
7.7.2 Style logique, style de paragraphe.....	48
7.7.2.1 <CITE> </CITE>.....	48
7.7.2.2 <CODE> </CODE>.....	48
7.7.2.3 <DFN> </DFN>.....	48
7.7.2.4 <EM> </EM>.....	48
7.7.2.5 <KBD> </KBD>.....	49
7.7.2.6 <SAMP> </SAMP>.....	49
7.7.2.7 <STRONG> </STRONG>.....	49
7.7.2.8 <VAR> </VAR>.....	49
7.7.2.9 <BLOCKQUOTE> </BLOCKQUOTE>.....	49
7.8 CRÉER DES LIENS.....	50
7.8.1 Définition d'une ancre.....	51
7.8.1.1 L'attribut HREF, ancre de départ vers un lien externe.....	52
7.8.1.2 L'attribut HREF, ancre de départ vers un lien interne.....	52
7.8.1.3 L'attribut HREF, ancre de départ vers un point spécifique d'un lien externe.....	52
7.8.1.4 L'attribut NAME, ancre d'arrivée.....	52
7.8.1.5 L'attribut HREF et NAME sur une même ancre.....	53
7.8.1.6 D'autres URL.....	54
7.8.2 Adressage des URL dans l'espace du serveur.....	54
7.8.2.1 Exemples.....	55
7.9 INCLUSION D'IMAGES.....	56
7.9.1 Les images en ligne.....	56
7.9.2 Les formats des fichiers Web.....	56

<b>7.9.3 Le format JPEG</b> .....	56
7.9.3.1 La compression JPEG.....	57
7.9.3.2 Quand utiliser le format JPEG ?.....	57
<b>7.9.4 Le format GIF</b> .....	57
7.9.4.1 La compression GIF.....	58
7.9.4.2 Les images GIF et le nombre de couleurs.....	58
7.9.4.3 En résumé.....	59
<b>7.9.5 &lt;IMG&gt;</b> .....	60
7.9.5.1 Alignement des images.....	60
7.9.5.2 Les images et les navigateurs non graphiques.....	60
7.9.5.3 Les images externes.....	61
7.9.5.4 Les images en guise d'ancre.....	61
<b>7.10 FOND DE PAGE, COULEUR DE TEXTE</b> .....	62
<b>7.10.1 Les fonds d'écran</b> .....	62
7.10.1.1 L'attribut BGCOLOR.....	62
7.10.1.2 L'attribut BACKGROUND.....	62
<b>7.10.2 La couleur du texte</b> .....	63
7.10.2.1 L'attribut TEXT.....	63
7.10.2.2 Les attributs LINK, VLINK et ALINK.....	63
<b>7.11 IMAGES CLIQUABLES</b> .....	64
<b>7.11.1 Système de coordonnées</b> .....	65
<b>7.11.2 Réalisation de la carte des coordonnées URL (ancienne méthode)</b> .....	65
<b>7.11.3 Description des coordonnées dans le fichier HTML (nouvelle méthode)</b> .....	67
<b>7.12 LES FORMULAIRES</b> .....	68
<b>7.12.1 &lt;FORM&gt; &lt;/FORM&gt;</b> .....	68
7.12.1.1 METHOD,.....	68
7.12.1.2 ACTION,.....	69
<b>7.12.2 L'attribut NAME</b> .....	69
<b>7.12.3 L'attribut VALUE</b> .....	69
<b>7.12.4 &lt;INPUT&gt;</b> .....	69
7.12.4.1 L'attribut TYPE.....	69
<b>7.12.5 &lt;SELECT&gt;</b> .....	70
<b>7.12.6 &lt;TEXTAREA&gt;</b> .....	71
<b>7.12.7 Envoi de fichier</b> .....	71
<b>7.12.8 Exemple de formulaire envoyant le résultat par e-mail:</b> .....	72
<b>7.13 LES TABLEAUX</b> .....	73
<b>7.13.1 &lt;TABLE&gt; &lt;/TABLE&gt;</b> .....	74
7.13.1.1 L'attribut BORDER=n.....	74
7.13.1.2 L'attribut CELLPADDING=n.....	74
7.13.1.3 L'attribut CELLSPACING=n.....	75
7.13.1.4 L'attribut WIDTH=n (pixels) ou n%.....	75
7.13.1.5 L'attribut HEIGHT=n (pixels).....	75
<b>7.13.2 &lt;TR&gt;</b> .....	75
7.13.2.1 L'attribut VALIGN.....	76
7.13.2.2 L'attribut ALIGN.....	76
<b>7.13.3 &lt;TD&gt;</b> .....	76
7.13.3.1 Les attributs COLSPAN et ROWSPAN.....	77
7.13.3.2 L'attribut NOWRAP.....	77

7.13.4 L'attribut <i>BGCOLOR</i> .....	78
7.13.5 <i>&lt;TH&gt;</i> .....	78
7.13.6 <i>&lt;CAPTION&gt;</i> .....	78
7.13.7 Exemple:.....	78
7.14 LES FRAMES OU CADRES.....	79
7.14.1 Introduction.....	79
7.14.2 Création de Frames.....	79
7.14.3 <i>&lt;FRAMESET&gt;</i> .....	80
7.14.3.1 L'attribut ROWS.....	80
7.14.3.2 L'attribut COLS.....	80
7.14.3.3 L'attribut BORDER (nouveau de Netscape 3).....	81
7.14.3.4 L'attribut BORDERCOLOR (nouveau de Netscape 3).....	81
7.14.4 <i>&lt;FRAME&gt;</i> .....	81
7.14.4.1 L'attribut SRC.....	81
7.14.4.2 L'attribut NAME.....	82
7.14.4.3 L'attribut MARGINWIDTH.....	82
7.14.4.4 L'attribut MARGINHEIGHT.....	82
7.14.4.5 L'attribut SCROLLING.....	82
7.14.4.6 L'attribut NORESIZE.....	82
7.14.4.7 L'attribut FRAMEBORDER (nouveau de Netscape 3).....	82
7.14.4.8 L'attribut BORDERCOLOR (nouveau de Netscape 3).....	83
7.14.5 <i>&lt;NOFRAMES&gt;...&lt;/NOFRAMES&gt;</i> .....	83
7.14.6 Exemple: (imbrication de balises <i>&lt;FRAMESET&gt;</i> ).....	83
7.14.7 Utilisation des Frames.....	85
7.14.7.1 L'attribut TARGET.....	85
<b>8. LES EXTENSIONS AUX FONCTIONNALITÉS DU WEB.....</b>	<b>87</b>
8.1 <i>&lt;EMBED&gt;</i> .....	87
8.2 <i>&lt;NOEMBED&gt;</i> <i>&lt;/NOEMBED&gt;</i> .....	88
8.3 EXEMPLES DE PLUG-IN.....	88
8.3.1 <i>LiveAudio</i> .....	88
8.3.2 <i>QuickTime</i> .....	89
<b>9. JAVASCRIPT.....</b>	<b>91</b>
9.1 EXEMPLE 1: MENU DÉROULANT.....	92
9.2 EXEMPLE 2: UN MESSAGE DÉROULANT DANS LA BARRE DE STATUT.....	94
9.3 EXEMPLE 3: AFFICHER UN TEXTE AU LIEU DE L'URL DANS LA BARRE DE STATUT.....	95
9.4 EXEMPLE 4: UN COMPTEUR D'ACCÈS PERSONNEL.....	96
9.5 EXEMPLE 5: AFFICHER UN ÉLÉMENT ALÉATOIREMENT DANS UNE PAGE HTML.....	98
9.6 EXEMPLE 6: AFFICHAGE DES ICONES DYNAMIQUES.....	99
<b>10. CONCLUSION:.....</b>	<b>101</b>

## 1. Bibliographie:

De nombreux livres traitent de la programmation HTML, mais certains sont de qualités inégales. Je ne citerai donc que les livres à jour et complets, dont la méthode pédagogique m'est apparue intéressante.

- HTML et la programmation de serveurs Web par Philippe Chaléat et Daniel Charnay, éd. Eyrolles<sup>1</sup>
- HTML pour les nuls, Ed Tittel et Steve James, éd. Sybex, 130 F
- HTML, JAVA, CGI, VRML, SGML, Création de site, coll. Secrets d'expert, éd. S&SM, 299 F
- Créer un serveur Web en HTML, coll. le programmeur, Laura Lemay, éd. S&SM, 179 F
- Grand Livre HTML, éd. Micro Application, 195 F
- Conception graphique de vos pages Web, coll. Studio Graphique, Lynda Weiman, éd. S&SM, 279 F \*
- CGI/PERL et JavaScript, création de pages Web interactives, Isaac Cohen, éd. Eyrolles, 185 F

---

<sup>1</sup> Certaines illustrations et extraits de textes proviennent de ces ouvrages



## **2. Logiciels recommandés pour créer des pages**

### **Web:**

#### **2.1 Sur PC/Windows:**

- Microsoft FrontPage pour la création en Tel écran - Tel écrit (WYSIWYG)
- PageMill pour la création en Tel écran - Tel écrit (WYSIWYG); logiciel commercial
- HotMetal pour l'édition directe du fichier HTML; shareware et commercial pour la version Pro
- Home Site pour la conversion et la modification des images; shareware
- MapEdit. pour la création des coordonnées des images cliquables; shareware
- Paint Shop Pro pour la conversion et la modification des images; shareware

#### **2.2 Sur Mac:**

- PageMill pour la création en Tel écran - Tel écrit (WYSIWYG); logiciel commercial
- Claris HomePage pour la création en Tel écran - Tel écrit (WYSIWYG); logiciel commercial
- PageSpinner pour l'édition directe du fichier HTML; shareware
- WWWWeaver pour l'édition directe du fichier HTML; shareware
- WebMap pour la création des coordonnées des images cliquables; shareware
- GifBuilder pour la création des GIF animés; freeware
- GraphicConverter pour la conversion et la modification des images; shareware



## 3. Introduction

### 3.1 À quoi sert un serveur Web? Quelles informations y publier? Quels outils matériels et logiciels (serveur, création) utiliser sur Mac, PC et UNIX? Les prix?

C'est à vous de répondre à ces questions pour bien cerner vos besoins. Si vous récupérez une machine (Mac ou PC), le serveur peut-être à faible prix en y installant des logiciels à la fois puissants et gratuits (Système Linux avec Apache comme serveur, Macintosh avec Quid Pro Quo comme serveur), de 5.000 à 30.000 FF sous Windows NT avec un serveur Netscape, et jusqu'à 500.000 FF pour les gros serveurs UNIX professionnels.

Sachez que les solutions gratuites sont très puissantes et peuvent traiter jusqu'à 10 connections par seconde pour environ 70 clients à l'écoute et maintenir une ligne à 1 Mb/s occupée à 80% sur des compatibles Macintosh à processeur PowerPC !! Cela représente près de 10.000 personnes par jour. Un Pentium 166 sous Linux, ou un PowerPC 604 à 200 Mhz sous Mac ou Linux avec 512 Ko de cache, 32 Mo Ram, 2 Go de DD AV vous apportent le meilleur rapport qualité-prix.

### 3.2 Régime juridique des pages, du design du site et des éléments attachés à ce serveur.

Cf. [http://www.insset.u-picardie.fr/WebDavid0/droit\\_des\\_pages\\_web.html](http://www.insset.u-picardie.fr/WebDavid0/droit_des_pages_web.html)

## **4. La construction d'un serveur Web.**

### **4.1 Introduction:**

Comme un livre ou un magazine, un CD-Rom ne peut être mis à jour dynamiquement. C'est un support moins efficace pour traiter des sujets d'actualités. Les éléments publiés sur le Web sont, par contre, disponibles immédiatement. et peuvent être mis à jour très régulièrement.

Publier un CD-Rom est plus cher. Peu de monde peut presser un CD-Rom ou imprimer un livre. Dans le cas du Web, tout le monde peut potentiellement créer une page.

La configuration minimale nécessaire est un ordinateur, un navigateur Web, un serveur, un éditeur de texte et de dessin, et accessoirement un scanner, un modem et une connexion à Internet.

### **4.2 L'urgence d'être sur le Web**

Beaucoup de sites Web apparaissent (du point de vue esthétique et organisationnel) comme ayant été pressés de se mettre sur Internet. Ceci, sans prendre en compte le désir des utilisateurs et leurs besoins.

Souvent, au lieu de créer des pages qui rassembleraient des informations dispersées avant de créer un serveur, ces derniers réutilisent du matériel de communication qui n'étaient pas destinés à l'origine pour le Web (Brochure publicitaire, plaquette d'information, ...).

Il est faux de se fier aux publicités qui disent et font croire que tout le monde peut produire de très beau livre, brochures et journaux avec quelques clics de souris. Ces publicités oublient de mentionner la variété des compétences requises pour créer ces belles pages.

Les auteurs de Web trop ambitieux (et souvent ignorant) doivent résister à la tentation de placer des liens juste pour l'art de cliquer. Dans le milieu commercial et d'affaire, le Web doit remplir des fonctions et un rôle particulier, et donc offrir l'information, le produit ou le service désirés.

### 4.3 La collaboration des compétences pour l'édition de Web

Les membres de l'équipe d'édition de Web doivent avoir ces compétences:

**Programmeur informatique:** familier avec les technologies de l'Internet, l'architecture des bases de données, les scripts CGI et la programmation en Java.

**L'infographiste et l'interfaceur:** doit connaître les limites et contraintes de l'édition d'applications interactives et capable d'organiser et de structurer les projets de grandes ampleurs.

**L'écrivain-journaliste:** il doit être familier avec l'hypertexte et ses implications pour structurer les documents; familier avec la documentation « en-ligne », les outils de recherche et méthodes de requêtes.

**Le commercial:** doit comprendre la puissance stratégique d'Internet et doit identifier les opportunités et futurs marchés et applications.

De bonnes compétences en marketing et management sont nécessaires pour comprendre les implications économiques, retour sur investissement, rassembler la bonne équipe pour créer le site. Le coût de cette mise en œuvre, des équipements, maintenance, doit être considéré et approuvé.

## **5. La phase préparatoire à la conception du site.**

### **5.1 Être familier avec le Web**

Un mauvais design peut donner les pires sentiments et critiques de la part du public et briser l'image de marque d'une entreprise ou d'une association. Un bon design peut, d'une certaine manière, entraîner un engouement de la part de la concurrence pour imiter ce site. Mais plus important, il donne une image de marque de qualité et de dynamisme contrôlé dans la maîtrise d'une nouvelle technologie.

Le meilleur moyen d'accroître ses connaissances sur le design des pages Web est de consulter souvent le Web.

### **5.2 Consulter les sites Web existants.**

Il y a des chances pour qu'un site Web concurrent du votre existe déjà sur Internet.

Il existe 5 catégories générales de sites Web:

- Les services tournés vers l'information (journaux, magazines, banques de données,...)
- Les institutions de recherche et d'éducation.
- Le divertissement (jeux, sports, loisirs, télé, radio, activités associatives, ...)

- Les entreprises qui communiquent avec l'Internet (industries, entreprises de services, publicité, ...)
- Les sociétés qui font du commerce sur Internet (VPC, catalogue, achat en ligne, ...)

### **5.3 Les limites et contraintes de la publication sur Internet.**

Diverses limitations et contraintes doivent être prises en compte. Ces limitations influencent directement le processus de création des pages Web. Sans cette connaissance, le site n'exploitant pas correctement cette technologie risque d'avoir de mauvaises surprises. Car l'internaute, quelle que soit la plate-forme sur laquelle il consulte votre page, ne veut pas savoir pourquoi une page s'affiche mal.

Il faut donc être au courant de toutes ces contraintes et des avancées technologiques qui permettent de les contourner. Cela s'apprend aussi en consultant beaucoup de sites Web.

#### **5.3.1 Comprendre ces contraintes et limitations techniques.**

##### **5.3.1.1 La vitesse de connexion.**

La plupart des utilisateurs se connectent par Internet via un modem à 14,4 ou 28,8 kps. Ainsi, une image de 100 Ko demande plus d'une minute dans une connexion de moyenne qualité.

Il faut aussi garder à l'esprit que les particuliers payent leur connexion à Internet et leurs factures téléphoniques. Il faut donc éviter qu'ils ne perdent du temps à charger des choses inutiles.

Compte tenu du type de connexion, le créateur de page Web doit utiliser les dessins et photos avec prudence. Il faut réserver les images pour les contenus qui

requièrent une illustration ou une description visuelle et que cette image apportera un réel bénéfice à l'utilisateur.

#### **5.3.1.2 La plate-forme de l'ordinateur.**

La plate-forme de l'ordinateur a une importance pour l'utilisateur car certaines opérations sont faites « localement ». C'est le cas de la mise en forme de la page Web une fois reçue, l'affichage des images, des fonds, des scripts, .... Cela inclut aussi les vidéos, les commentaires sonores, musiques, etc.. Ces pages, au rendu complexe, sont affichées beaucoup plus rapidement sur des ordinateurs puissants que sur ceux qui le sont moins et ceci indépendamment de la vitesse de connexion.

#### **5.3.1.3 Afficher des images et du texte en haute résolution.**

Une autre manière d'afficher les pages plus rapidement est de travailler sur la qualité du rendu des images. Par exemple, une image de 15cm sur 10 aura une taille de 90 Ko à 8 bits ou 256 couleurs. La même image en 5 bits - 32 couleurs prendra 66 Ko sans compression. La meilleure réduction se fait au format JPEG pour donner entre 30 et 10 Ko selon le taux de compression de l'image.

La résolution à l'écran et la profondeur des couleurs ont une énorme importance dans l'aspect visuel d'une page Web.

Une profondeur de 16 ou 24 bits (32000 ou 16,7 millions de couleurs) ne sont pas les configurations minimales de tous les utilisateurs qui souvent, ne peuvent afficher que 256 couleurs à la fois. Pour cela, ils doivent acheter de la mémoire vidéo supplémentaire ou mettre à jour leur carte vidéo.

Dans ce cas, Netscape essaye de corriger cette faiblesse en affichant des images avec les nuages de point pour simuler une image en 16 ou 24 bit sur un écran ne pouvant afficher que 256 couleurs à la fois. On appelle cela « postérisation ».



De plus, la résolution de l'écran varie suivant les systèmes. Le format le plus commun est 640x480 pixels. Pour des questions de lisibilité sur ces écrans, on limitera la largeur des pages Web à 600 pixels maximums.

L'affichage du texte pose un défi au créateur de pages Web. La taille des polices n'existe pas en HTML. Le type de police n'est pas fixé (Palatino, Times, ...), et c'est l'utilisateur qui la définit dans ses préférences.

Il est difficile de fixer les espaces entre les éléments de la page Web car la mise en page s'adapte à la dimension de la fenêtre d'affichage de l'utilisateur. Le meilleur conseil est donc de visualiser le résultat final sur différents ordinateurs avec différentes configurations pour trouver un résultat correct.

#### **5.3.1.4 Le logiciel de consultation des pages Web.**

Les logiciels Navigateurs affichent eux aussi différemment les pages Web selon leur version et le système sur lesquels ils tournent. Certains acceptent d'afficher des balises HTML alors que d'autres non.

De plus, tous peuvent éviter de charger les images d'une page Web afin de naviguer plus vite lorsque la connexion est trop lente. Ce qui dénature profondément le design d'une page.

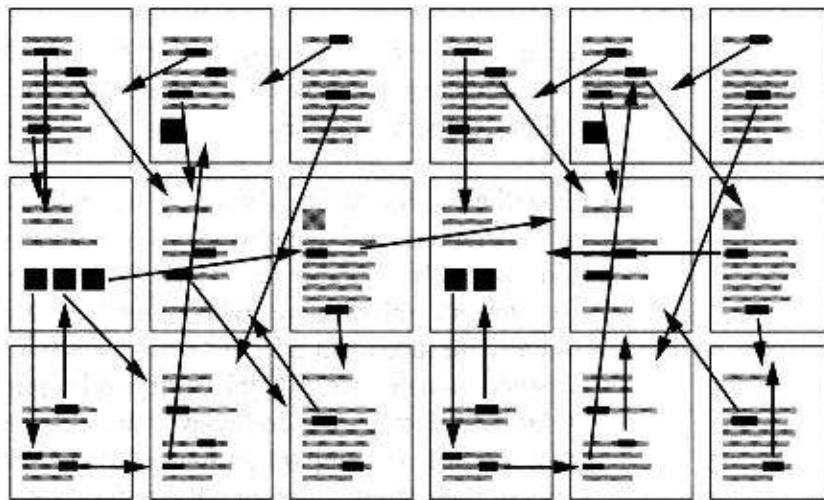
Netscape 3 et MS Internet Explorer 3 offrent maintenant à peu près les mêmes fonctions d'affichages. Mais Netscape reste le standard incontesté actuellement. Il a notamment introduit l'utilisation des tableaux (TABLE) et des cadres (FRAME) permettant de mieux contrôler la mise en page. De plus, on peut modifier les fonds d'écran, les couleurs d'affichages, faire des pages Web dynamiques, multimédia ou en 3 dimensions grâce aux extensions (ou Plug-in) de Netscape.

### 5.3.2 Limitations et contraintes pour l'utilisateur.

#### 5.3.2.1 Être perdu dans le Cyberespace.

L'internaute peut être facilement désorienté et perdu pendant qu'il navigue dans les sites Web précisément parce que la force du Web est la capacité à traverser de manière non linéaire l'information.

La principale cause de confusion est que l'organisation de la structure de l'information n'est pas aérée. Même avec peu de documents, l'internaute peut perdre son sens de l'orientation sur un site Web.

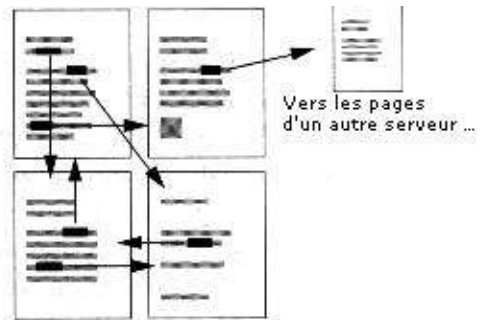
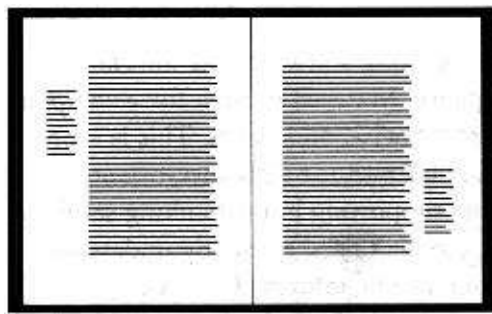


#### 5.3.2.2 Consulter le Web opposé à la lecture de publications imprimées.

Un livre est généralement linéaire dans son organisation, sa structure et sera vu dans différentes positions physiques par le lecteur (assis, debout, couché, ...).

La consultation du Web est totalement différente car tout est fragmenté, lu rapidement, cliqué ou plutôt zappé tout aussi vite. Souvent, l'internaute ne fait que survoler les pages. La capacité d'attention est beaucoup plus courte lorsque le désir d'information est grand. Car l'on sait ce que l'on cherche et l'on ne s'arrêtera que lorsqu'on l'aura trouvé. Ou alors, il agit ainsi car rien n'a attiré son attention, et il navigue aussi vite dans l'espoir de trouver ce qui éveillera sa curiosité.

Une autre simple raison de cette attitude est la difficulté de lire les textes des pages sur l'écran d'un ordinateur à cause du scintillement.



**Livre contre document Hypertexte**

L'information proposée doit donc être franche, directe et concise. C'est d'autant plus important que le site Web propose une information spécialisée. L'internaute doit vite comprendre ce qu'on attend de lui devant n'importe quelle page Web.

## **6. Déterminer les objectifs du projet.**

### **6.1 Pourquoi créer un site Web?**

Déterminer les objectifs et les buts du projet sont les étapes préliminaires obligatoires pour produire un site Web. Un objectif précis détermine la raison fondamentale de la présence sur le Web. Cela permettra de finaliser et d'adapter la conception même des pages.

### **6.2 Développer l'ossature d'une organisation**

Être compréhensible requiert de la part de l'équipe responsable du projet d'organiser et de structurer l'information avant de prendre la décision de déterminer un style visuel et de présentation précis.

Un plan d'ensemble doit d'abord être développé pour tout organiser de façon cohérente et pour que chacun comprenne son rôle dans ce projet.

Toutes les utilisations, activités et situations doivent être anticipées et ajustées par le responsable du projet. Parfois, le plan nécessite d'envisager des possibilités futures d'extensions pour ne fermer aucune possibilité.

### **6.2.1 Le cadre organisationnel du Web.**

Se précipiter sur l'aspect visuel de la page sans planification c'est comme construire une maison sans plan d'architecte. C'est le désordre assuré.

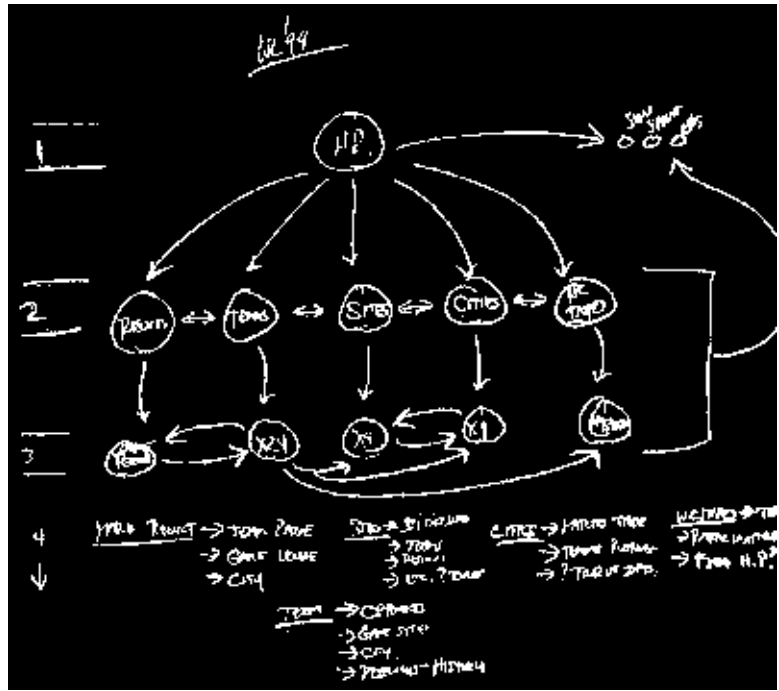
Souvent les avantages du lien hypertexte sont mal compris, c'est-à-dire interprété comme lier une information tout le temps depuis n'importe où vers n'importe quoi.

#### **6.2.1.1 Structurer l'information.**

Pour éviter d'être confrontés à des ambiguïtés, une organisation est nécessaire surtout si le sujet est complexe à traiter ou demande une vaste documentation. La nature même de l'hypermédia avec une hiérarchie que l'on distingue facilement, offre une navigation et une intuitivité qu'un système complexe de liens indifférenciés n'offre pas.

Si l'internaute se pose les questions suivantes: « D'où viens-je? », « Comment y suis-je venu? », « Comment retourner en arrière? », de telles questions sont les symptômes d'un site Web mal structuré à l'organisation floue.

En terme de facilité d'utilisation, le responsable du projet doit développer le sens de la prédictabilité. C'est-à-dire prévoir le comportement de l'internaute face à un site Web.



#### 6.2.1.2 Ajouter des liens dans la structure.

L'information doit être accessible de n'importe quel point du site Web. La meilleure organisation permet d'aller de n'importe quel point en moins de quatre clics de souris quelle que soit la taille du site.

Cependant, offrir un trop grand nombre de choix peut rendre la navigation encore plus confuse et donner un résultat inverse à celui souhaité: être perdu dans le Cyberspace!

#### 6.2.1.3 Intégrer des outils de recherches.

Dans de vastes sites Web, les moteurs de recherche sont complémentaires à l'organisation et devraient être inclus dans le projet si nécessaire. L'avantage principal de ce moteur de recherche est de délivrer à l'internaute une information spécifique et personnalisée.

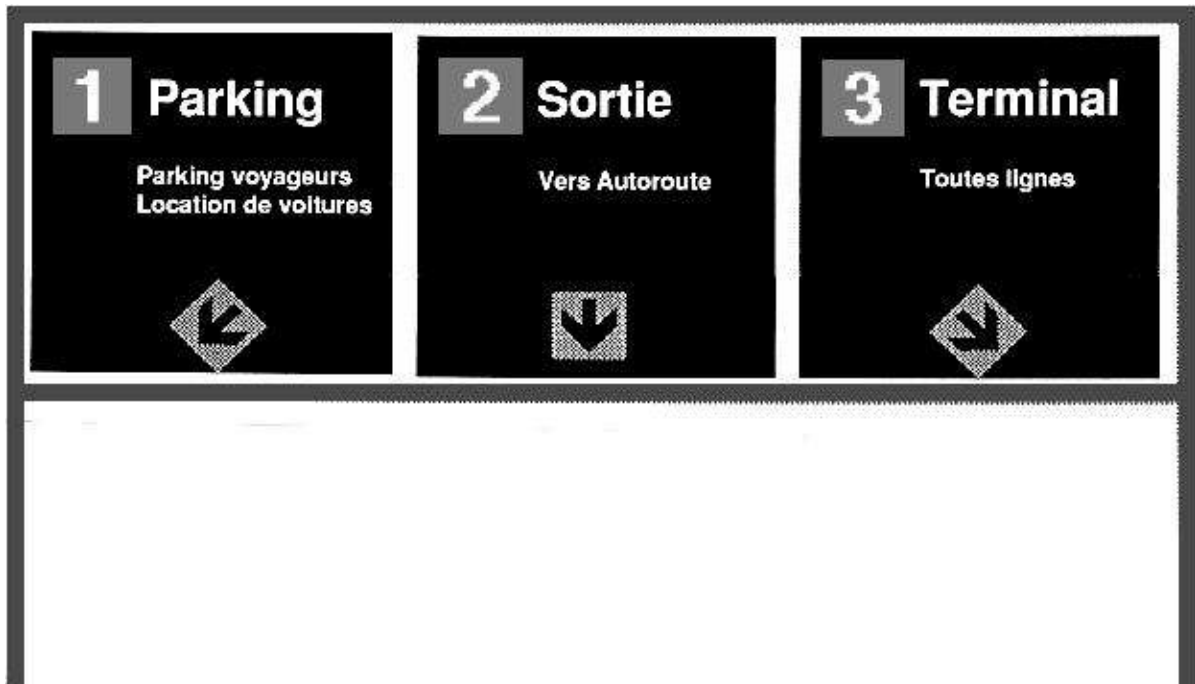
## **6.2.2 Regrouper le contenu et les activités.**

Une fois que le but et les objectifs du projet ont été définis, on évalue la quantité d'information à déposer et les activités à proposer. L'équipe responsable du projet doit rassembler toutes les informations disponibles et la grouper en plusieurs catégories. Ces catégories auront des relations entre elles afin d'offrir des passerelles dans le site. De ces groupes, il faut encore essayer de les synthétiser dans des catégories plus générales.

### **6.2.2.1 Utiliser la hiérarchisation pour organiser les groupes**

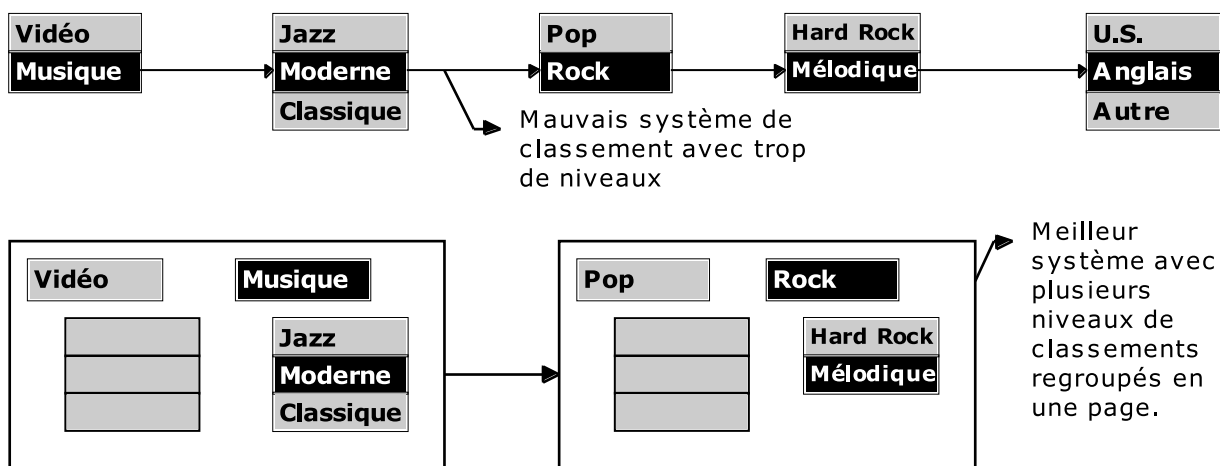
Il faut hiérarchiser ces grandes catégories en fonction de l'objectif et du but fixés. Cette hiérarchie doit surtout être dictée par les raisons pour lesquelles un internaute consultera votre site. Vous devrez répondre à ces questions: Pour quelles raisons un Internaute viendrait-il sur notre site, pour y chercher quoi? Grâce à la prédictabilité, vous connaîtrez les deux ou trois raisons principales des visites et adapterez la hiérarchie en conséquence.

Utiliser la hiérarchisation dans l'aspect visuel d'une page Web facilite la compréhension d'une information. Les panneaux indicateurs utilisent la hiérarchie pour communiquer des informations importantes aux conducteurs dans un temps très court de quelques secondes. L'œil humain navigue de l'élément le plus gros vers le plus petit en allant de préférence de la gauche vers la droite (dans les pays occidentaux). Dans ce cas, l'élément dominant est large et situé au sommet (le chiffre), et a une couleur distincte. Juste après, un texte large indique la destination des voies. En dessous, un texte de caractère plus petit, donne des informations complémentaires. Enfin, la flèche se trouve bien détachée vers le bas pour être claire et confirmer l'information en la signalant autrement afin de lever toute ambiguïté.



Organiser l'information hiérarchiquement est nécessaire pour faciliter la compréhension, mais est aussi très pratique. Tout ne peut pas être placé dans la page d'accueil !

La structure hiérarchique ne doit pas aller trop profondément (3 ou 4 clics maximums). Il faut se rappeler que le but premier d'un site Web est de permettre à l'internaute de satisfaire des besoins précis.





### 6.3 Adapter le site aux besoins de l'internaute.

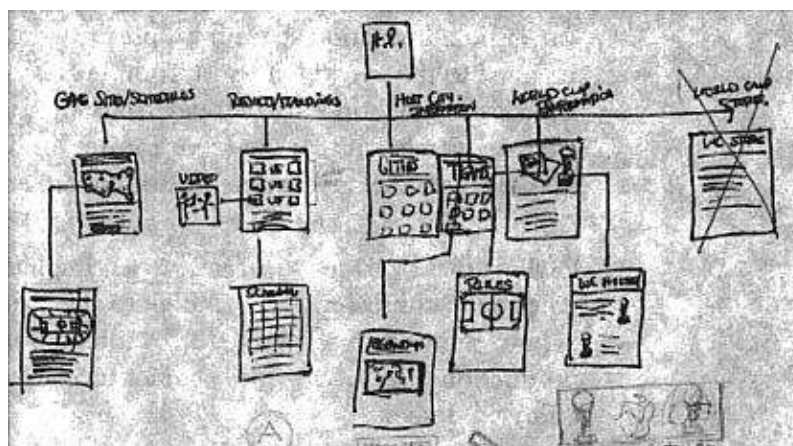
Une fois que l'ossature organisationnelle est mise en place, le comportement type de l'internaute devant remplir une fonction précise est décrit, analysé, disséqué et vérifié. L'internaute visitera un site pour trois raisons principales:

1. Pour avoir une information générale sur le sujet développé,
2. Pour relever une information très précise difficilement trouvable autrement,
3. Pour prendre contact avec l'organisation (contact commercial ou relationnel).

On doit créer un site où la navigation est « intuitive ». Pour cela on doit pouvoir aller directement et simplement vers l'information recherchée.

Mais, il ne faut pas encore commencer à écrire une seule ligne de code HTML. Le meilleur moyen est encore la feuille de papier pour facilement corriger et avoir une vue générale, surtout s'il faut souvent changer les liens.

Créer toute l'organisation au brouillon sur papier est beaucoup plus souple et permet de mieux visualiser le cheminement que fera l'internaute à travers le site.

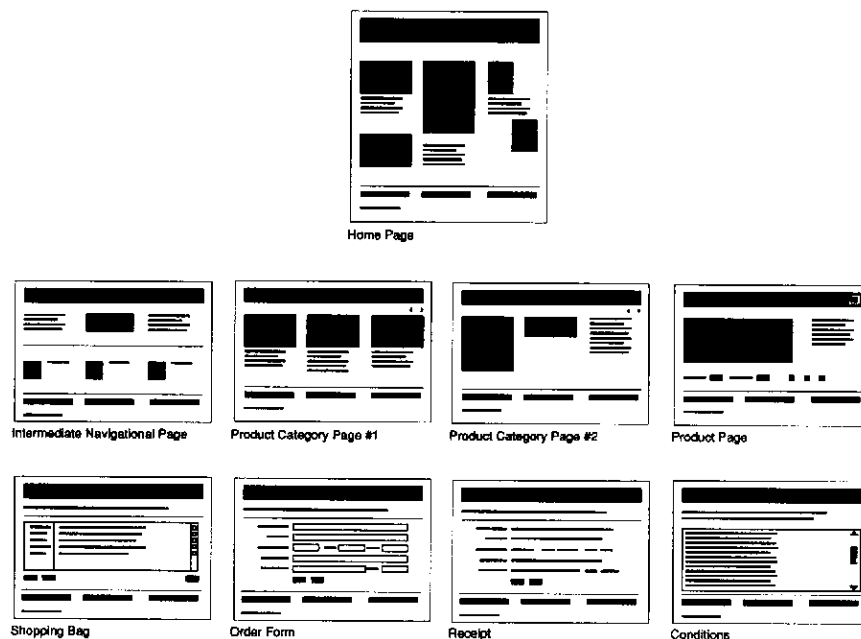


## 6.4 Créer le prototype en HTML

Une fois que l'équipe de développement a donné son accord sur l'organisation, on peut commencer à faire un prototype en HTML. Ce processus sera efficace si le prototype papier a été détaillé (mais pas trop) et souple dans son organisation. Car l'avis des nombreux utilisateurs va faire que vous apporterez de nombreuses modifications.

### 6.4.1 La création de pages types.

Dans un site vaste, la définition de modèles est nécessaire pour des questions de rapidité de réalisation, de simplicité et de clarté de l'interface. Chaque page type aura une fonction précise où tous les éléments de la page Web seront à la même place (texte, images, menu, barre de navigation, ...).



#### 6.4.1.1 La Page d'accueil.

La page d'accueil est la page principale du site et sera souvent la première page dans laquelle l'internaute entrera. Elle doit l'impressionner, générer de l'intérêt et de la

curiosité. Cette première impression compte beaucoup car elle déterminera si l'internaute continuera à naviguer dans votre site ou s'il zappera ailleurs.

Une information minimale est requise. On doit vite pouvoir identifier ce que le site propose à l'utilisateur et qui est l'éditeur. Il doit y avoir une indication claire du contenu du site, les outils de navigations essentiels avec les grands groupes de hiérarchies et surtout ne pas oublier l'adresse E-mail du responsable de ce site pour toutes remarques.

#### **6.4.1.2 Les pages de navigations intermédiaires.**

Elles permettent de présenter des sous catégories si votre site est très vaste. Mais il ne doit pas y en avoir trop. Une seule doit suffire.

#### **6.4.1.3 Le contenu des pages.**

Il faut que dans cette page se trouve les éléments informant l'utilisateur de sa position par rapport à la hiérarchie du site. Il doit aussi avoir la possibilité de revenir en 3 ou 4 clics de souris à n'importe quel élément du site.

## **7. Le langage HTML**

### **7.1 Les bases de la programmation**

#### **7.1.1 Faut-il encore apprendre HTML ?**

On voit souvent dans la presse que tel traitement de texte propose dans sa nouvelle version la possibilité de sortie de documents en langage HTML. D'autres produits se proposent de convertir tel format de texte en HTML. Des éditeurs plus ou moins Tel écran - Tel écrit (WYSIWYG) apparaissent çà et là ; souvent ils n'offrent que la possibilité d'insérer des balises en allant cliquer dans un menu, et tous ne font pas une analyse structurelle du fichier, ce qui serait le plus intéressant. Ils évitent essentiellement de connaître le balisage et les règles de structuration. Tous ces produits n'indiquent pas toujours clairement quel niveau d'HTML ils sont capables de traiter ou de produire. Là est peut-être le problème le plus crucial. Actuellement, nous en sommes à la version HTML 3.2 .

Il reste enfin que toute la partie interactive (les formulaires) n'est pour l'instant accessible qu'en travaillant au niveau du langage.

La connaissance du langage ne sera pas en tout cas un handicap, même si dans un avenir plus ou moins proche on disposera de véritables produits pour générer des pages sur le Web.

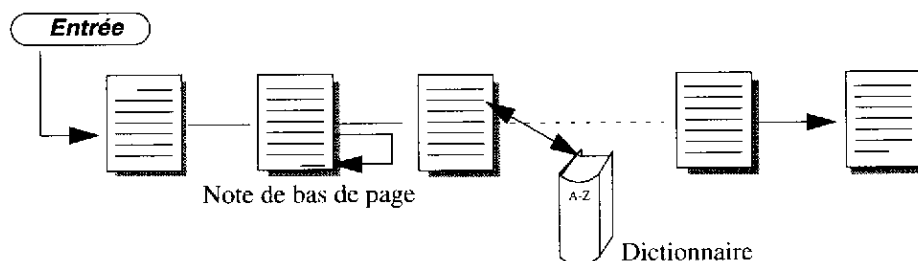
### 7.1.2 L'hypertexte

On pourrait définir l'hypertexte comme un système qui relie par des liens activables des éléments d'informations. Dans le concept du World Wide Web, on devrait préférer au terme hypertexte le terme hypermédia puisque l'on va pouvoir accéder au texte mais aussi à l'image, au film et au son.

Effectuons tout d'abord un rappel sur l'organisation des documents imprimés dans les types ou formats les plus courants :

#### 7.1.2.1 Le livre

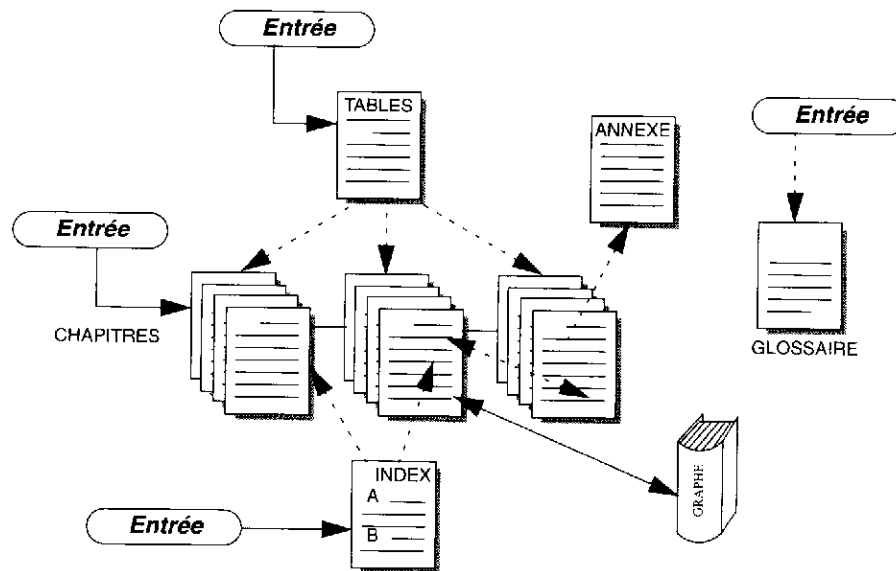
On accède généralement à un tel ouvrage (livre, roman) en commençant par la première page, la lecture se poursuivant dans l'ordre chronologique des pages ; le seul déroulement dans la lecture est celui introduit par la note de bas de page. Le document se suffit à lui-même, et il n'y aura en principe pas d'accès nécessaire à d'autres ouvrages (éventuellement on aura recours à un dictionnaire).



#### 7.1.2.2 Le document technique

L'accès à l'information va se faire en choisissant d'abord le volume, puis on orientera sa recherche en consultant soit la table des matières, soit l'index pour naviguer dans le document. On sera aussi peut-être amené à consulter un glossaire ou diverses annexes afin de compléter la réponse attendue. Une telle structure de document est dite hiérarchique ou arborescente ; sa consultation est plus difficile

lorsqu'on travaille sur une édition "papier" que lorsqu'il s'agit d'un document électronique. Ce type de document sera un très bon candidat à une présentation hypertextuelle.



### 7.1.2.3 Le journal

Cet autre type de document correspond à un besoin différent. Nous allons l'étudier car il est intéressant d'analyser la similitude entre la première page, la "une" d'un journal, et la page d'accueil d'un service sur le Web. La première page d'un journal est conçue pour répondre aux critères suivants :

- identifier le journal (logotype, typographie, mise en page)
- attirer l'oeil (couleur, iconographie)
- donner envie d'acheter le journal pour continuer la lecture que l'on aura pu commencer sur la première page.

### 7.1.3 Notions d'hypertexte

Dans les deux derniers types de documents décrits on voit apparaître le principe de liens entre divers éléments de texte non contigus. Dans une forme "papier" des documents on exécute ces liens en tournant des pages ou en allant chercher sur des rayons des livres au gré des références, notes de bas de page ou autres renvois rencontrés dans la lecture. Il sera chaque fois nécessaire de préserver le retour au point

d'interruption de la lecture en marquant la page par un signet. Dans un document électronique, on ira consulter ces liens en cliquant sur un repère spécial ; la sauvegarde du point de retour devra être assurée par une mémoire du système.

## **7.2 HTML, son environnement**

### **7.2.1 Qu'est-ce que HTML ?**

HTML HyperText Markup Language est un langage simple utilisé pour créer des documents hypertexte pour WWW. Ce n'est pas un langage de description de page tel que PostScript; HTML ne permet pas de définir de façon stricte l'apparence d'une page, même si une utilisation un peu détournée le fait utiliser quelque fois comme un langage de présentation. De plus, la présentation de la page est très dépendante du navigateur utilisé: le même document ne produit pas le même résultat sur l'écran s'il est affiché avec un navigateur en mode ligne, Mosaic ou Netscape.

HTML s'attache à décrire la structure et le contenu d'un document, et non le format de la page et son apparence. Comme tout langage, il est en constante évolution. La version 3.2 est en cours de finalisation. Sa simplicité est telle qu'il n'est pas nécessaire d'utiliser un éditeur particulier. Il demande de la rigueur et de l'attention dans l'écriture des documents afin que ceux-ci s'affichent correctement quels que soient le contexte et le navigateur utilisé.

### **7.2.2 Dans quel contexte s'exécute HTML ?**

- Le client : c'est un logiciel de consultation s'exécutant sur tous les types de plateforme (Macintosh, station Unix, PC). Il dialogue avec un serveur selon un protocole

spécifique (HTTP HyperText Transfert Protocol) qui va lui fournir l'information structurée en code HTML. C'est ce client qui a en charge l'exécution de ce code (on dit aussi du langage source) et qui produira l'affichage du document.

- Le serveur : ce logiciel est en permanence à l'écoute des requêtes que vont formuler les clients. Il en vérifie la validité, s'assure que la demande émane d'un client autorisé à accéder au document, et lui envoie normalement texte et image avant de clore la connexion.

### 7.2.3 Le protocole d'adressage des documents: l'URL

L'adresse d'un document, appelée URL Uniform Resource Locator va être composée à l'aide des éléments suivants:

- le protocole d'échange entre le client et le serveur. A ce niveau du cours le seul protocole que nous ayons rencontré est HTTP; nous verrons plus tard qu'il en existe d'autres;
- l'adresse Internet du serveur qui diffuse les documents. Cette adresse est unique sur tout le réseau, c'est l'adresse TCP/IP de la machine. Elle a la forme d'une suite de nombres telle que 134.158.69.113; comme ces nombres ne sont pas facilement mémorisables, un annuaire (DNS) résout généralement la relation adresse numérique nom symbolique de la machine | nom du domaine (exemple : 193.57.117.6 est l'adresse de la machine www.insset.u-picardie.fr où www représente le nom de la machine et insset.u-picardie.fr le nom du domaine) ;
- l'arborescence des répertoires (le chemin) qui conduit au document ;
- le nom du document qui aura toujours l'extension .html.
- La syntaxe minimum utilisée pour représenter l'URL d'un document est la suivante:

```
protocole://nom_du_serveur/
```

lorsqu'il n'y a pas de nom de fichier précisé on arrivera sur le fichier par défaut du serveur, habituellement la page d'accueil. La syntaxe que l'on rencontre couramment est :

```
protocole://nom_du_serveur/repertoire/sous-repertoire/nom_du_document
```



La syntaxe complète est :

```
protocole://username;password@nom_du_serveur:port/sous-  
repertoire/nom_du_document?arguments
```

Les fichiers ou documents que diffusent le serveur sont stockés dans des répertoires du serveur. Il existe un système de protection des fichiers propres à WWW. Il comporte plusieurs niveaux de protections.

## 7.3 Structure

### 7.3.1 Les balises

Le langage HTML permet de décrire la structure d'un document pour WWW. Cela sera obtenu en encadrant chacune des différentes structures du texte par une paire de balises, une au début et une autre à la fin.

Dans un traitement de texte lorsque l'on veut mettre un mot en gras, on sélectionne les caractères composant ce mot, puis l'on clique généralement sur une petite icône qui a pour effet de faire passer les caractères dans la graisse choisie. De façon interne et sans que cela apparaisse à l'écran, le traitement de texte va inscrire dans le fichier un code indiquant que commence une zone en caractère gras, puis l'on trouve les caractères significatifs, et enfin un nouveau code indiquant le retour aux caractères standards.

Dans HTML les balises sont délimitées par les signes < (inférieur à) et > (supérieur à). Un texte balisé aura donc cette apparence:

...texte courant <balise>texte affecté par la balise</balise> suite du texte ...

On remarquera le caractère / dans la balise de fin. De plus, on peut écrire indifféremment la balise en majuscule ou en minuscule, voire en mélangeant les deux. Le résultat sera le même.

### 7.3.2 L'accentuation

L'enregistrement des fichiers contenant les documents HTML est effectué sur la machine serveur en code ASCII 7 bits, ce qui ne permet pas d'accentuer directement les textes. On utilisera donc des mots clés pour tous les caractères accentués et autres symboles.

Le code des mots-clés pour les lettres accentuées est ISO Latin 1. Ainsi la phrase suivante:

"Le Naïf" a déjà été créé au théâtre français.

sera codée en HTML par :

```
"le Na&iuml;l;f" a d&eacute;j&agrave; &eacute;t&eacute; cr&eacute;e au th&eacute;âtre fran&ccedil;ais.
```

## 7.4 Structuration du document HTML

### 7.4.1 <HTML>

Tous les documents HTML commenceront par la balise <HTML> et se termineront par la balise </HTML>.

### 7.4.2 <HEAD>

Le titre du document sera généralement inclus dans une zone de prologue ou d'en-tête : <HEAD>. Cette zone se termine par </HEAD>.

Entre cette balise, on pourra insérer divers éléments comme la balise <TITLE>, <LINK>, <META>, <BASE>, <SCRIPT>, etc.

### 7.4.3 <TITLE>

Un texte qui apparaît dans la zone titre du navigateur, est défini par <TITLE>. Le titre défini entre les balises <title> et </title> a, outre un caractère informatif.

### 7.4.4 <BASE>

Cette balise donne la racine des URL des liens relatifs du document HTML (cf. Adressage des URL dans l'espace du serveur).

ex: `<BASE HREF="http://www.insset.u-picardie.fr/">`  
...  
`<IMG SRC="images/photo.gif">`

#### 7.4.5 <META>

Cette balise décrit certaines propriétés du document comme l'auteur, sa date d'expiration, des mots-clés, une description de la page, etc.

```
<META NAME="Author" CONTENT="Ralph Davidovits">
<META HTTP-EQUIV="Expires" CONTENT="Tue, 20 Aug 1996 14:25:27 GMT">
<META NAME="Description" CONTENT="Ce site présente une page Web">
<META NAME="Keywords" CONTENT="Web Internet site internaute">
<META HTTP-EQUIV="Refresh" Content="15; URL=http://:www.truc.fr/">
```

Ce dernier exemple est intéressant car il permet de forcer le navigateur Web à charger automatiquement une nouvelle page au bout de tant de secondes. Dans l'exemple ci-dessus, au bout de 15 secondes, la page d'accueil qui se trouve sur le serveur <http://:www.truc.fr/> sera chargée.

#### 7.4.6 <LINK>

Très peu de logiciels navigateurs utilisent cette balise. Elle permet de rajouter des boutons dans la barre de navigation du logiciel comme raccourci vers des pages définies. Netscape ne les reconnaît pas.

```
<LINK REL=(Contents, Index, Glossary, Copyright, Next, Previous, Help, Bookmark)
  HREF="http://...">
<LINK HREF="mailto:ralph.davidovits@insset.u-picardie.fr">
<LINK REL=Bookmark TITLE="WebDavido: Le Village Mondial"
  HREF="http://www.insset.u-picardie.fr/WebDavido/index.html">
<LINK REL=Contents HREF="http://www.insset.u-
  picardie.fr/WebDavido/table_des_matières.html">
```

#### 7.4.7 <SCRIPT>

Cette balise spécifie le script utilisé pour automatiser certaines tâches et créer une page HTML plus dynamique.

```
<SCRIPT="JavaScript">
<!-- on cache le script
...
// fin de la cachette -->
</SCRIPT>
```

#### 7.4.8 <BODY>

Entre <BODY> et </BODY> va résider tout le reste du document. Nous verrons dans la suite du cours que des extensions HTML 3 permettent de compléter l'utilisation de cette balise. Nous parlerons des attributs optionnels BGCOLOR et BACKGROUND un peu plus tard.

#### 7.4.9 <ADDRESS>

Le bloc adresse est un élément prévu pour indiquer toutes informations concernant (l'auteur du document (adresse, téléphone, E-mail, etc...)). Il peut être inséré à n'importe quel endroit du document ; l'habitude dans WWW est de mettre cette zone adresse à la fin du document. Les navigateurs affichent généralement le texte compris entre <address> et </address> en caractère italique.

#### 7.4.10 <!-- Commentaires -->

Tout texte commençant par <!-- et se terminant par --> ne sera pas interprété par le navigateur, donc pas affiché. Ceci sert à l'auteur du document pour commenter son fichier source.

#### 7.4.11 Attributs d'une balise

Nous verrons dans l'étude des différentes balises HTML que certaines peuvent admettre des attributs, chacun de ces attributs pouvant avoir une valeur. Au niveau syntaxique, le premier attribut est séparé de la balise par un espace, ainsi que les attributs entre eux. Si l'attribut doit prendre une valeur, elle est spécifiée après le signe = (égal) suivant l'attribut.

La valeur de l'attribut sera écrite entre guillemets (") si cette valeur est alphanumérique.

```
<balise attribut1=numerique attribut2="alphanumerique">.
```

#### 7.4.12 Squelette d'un document HTML

Nous avons maintenant défini suffisamment d'éléments de structure du langage HTML pour lister l'ossature type que nous trouverons toujours dans un document HTML :

```
<HTML>
  <HEAD>
    <TITLE>Essai</TITLE>
  </HEAD>

  <BODY>
    .....
  </BODY>

</HTML>
```

### 7.4.13 Tester les documents HTML

Voici les étapes à suivre pour tester les fichiers HTML avant de les installer dans les répertoires du serveur :

- appelez l'éditeur de texte,
- écrivez le code html,
- enregistrez sous un nom avec le suffixe .html
- appelez le navigateur Netscape par exemple,
- utilisez le menu Open file (Netscape),
- sélectionnez votre document où va s'afficher dans le navigateur,

Si vous n'êtes pas satisfait...

- retournez dans l'éditeur de texte sans quitter le navigateur,
- corrigez ou modifiez votre code html,
- utilisez la commande reload et jugez de vos modifications.

## 7.5 Divisions

Pour être plus compréhensif tout texte est habituellement divisé selon des paragraphes subissant une numérotation hiérarchique. Dans HTML encore une fois tout est à la charge de l'auteur. Celui-ci dispose de six grosseurs de titre qui vont lui permettre de diviser son document et d'un système de liste.

### 7.5.1 <Hn> </Hn>

C'est la balise affectant la taille des caractères où n varie de 1 à 6. Les plus gros ont la valeur 1 et les plus petits la valeur 6. Le texte entre ces balises est traité en caractère gras. La balise de début et la balise de fin génèrent automatiquement un

espace de type nouveau paragraphe (passage à la ligne avec espace correspondant environ au saut d'une ligne). A l'intérieur de cette balise, on peut rajouter un élément optionnel permettant d'aligner le titre à gauche, au centre ou à droite de l'écran (Align= Right, Left ou Center).

EX: `<H1 ALIGN=Center> Titre </H1>`

### 7.5.2 <BR>

Lorsqu'ils exécutent du code HTML les navigateurs traitent le texte au kilomètre dans la largeur de leur fenêtre et ignorent le découpage en ligne tel qu'il existe dans le fichier source. La longueur des lignes sera donc fonction de la taille que fixe le lecteur en dimensionnant la fenêtre de son navigateur. Il appartient à l'auteur du document de provoquer lorsqu'il le souhaite un passage à la ligne. La balise BR qui génère ce passage est une balise vide c'est-à-dire qu'il n'y a pas de balise de fin (</BR> ne s'utilise pas). Quatre fonctions optionnelles s'ajoutent pour empêcher du texte ou des images de déborder à droite, à gauche ou toute la largeur de la fenêtre (Clear=Right, Left ou All).

EX: `<BR CLEAR=All>`

### 7.5.3 <P>

La balise <P> provoque le passage au paragraphe suivant ; tout comme <BR> cette balise est généralement considérée comme une balise vide, bien qu'il soit tout à fait valide de terminer un paragraphe par </P>. P est différente de BR : elle provoque un passage à la ligne et décale la ligne suivante d'un espace d'environ une ligne. Contrairement à la balise <BR>, la balise <P> n'est pas cumulable. Si vous en ajoutez plusieurs l'un à la suite de l'autre, le logiciel Web n'en prendra en compte qu'un seul. Des attributs optionnels à cette balise existent pour aligner les textes et images dépendantes du paragraphe (Align=Right, Center et Left).

EX: `<P ALIGN=Center>`

Seuls les tableaux <TABLE> ne sont pas affectés par cet alignement et restent alignés à gauche.



#### 7.5.4 <DIV> </DIV>

Cette balise est un peu l'équivalent du <P ALIGN=...>. Elle s'accompagne obligatoirement des attributs ALIGN=Right, Left ou Center. Elle permet cette fois d'aligner les tableaux <TABLE>.

EX: <DIV ALIGN=Right>

#### 7.5.5 <CENTER> </CENTER>

Ceci est la première fonction d'alignement qu'introduit Netscape 1. Elle est synonyme à <P ALIGN=Center> et <DIV ALIGN=CENTER>. Elle permet que le centrage soit compatible avec les plus anciens logiciels Web.

#### 7.5.6 <HR>

C'est une balise de division graphique qui dessine un trait sur toute la largeur de la fenêtre du navigateur. Il est recommandé d'utiliser de cette possibilité avec modération. Elle peut être utilisée seule mais plusieurs options sont disponibles: Align (Right, Center et Left), NOSHADE (qui affiche un trait simple sans relief généralement de couleur noire), SIZE (épaisseur du trait en pixel) et WIDTH (largeur du trait en pixel ou en %).

EX: <HR NOSHADE SIZE=5 WIDTH=60%>

#### 7.5.7 <PRE> </PRE>

Cette balise permet de respecter la mise en page précise d'un texte ou d'une portion de texte, de la façon dont il a été défini dans le fichier HTML. Le navigateur

interprète ce texte préformé à l'aide d'une police de caractères non proportionnelle, ce qui autorise des alignements de type tableaux.

La balise PRE admet l'attribut WIDTH qui permet de fixer le nombre de colonnes dans lequel va se situer le texte non formaté (`<pre width=40>` par exemple). Si cet attribut est absent la valeur prise par défaut est de 80 colonnes. Cet attribut n'est pas interprété par tous les navigateurs.

## 7.6 Listes

HTML définit deux types de listes :

- un type de liste descriptive, typiquement un glossaire,
- un type de liste régulière ou énumération avec ou sans numérotation.

### 7.6.1 Liste descriptive

#### 7.6.2 <DL> </DL>

L'élément <DL> ouvre une liste descriptive. Il définit le début de la liste et va englober deux autres balises (DT et DD) dont le rôle est de caractériser, de désigner chacun des éléments, la partie définition et l'élément lui-même. La fin de la liste sera caractérisée par la balise </DL>.

##### 7.6.2.1 <DT>

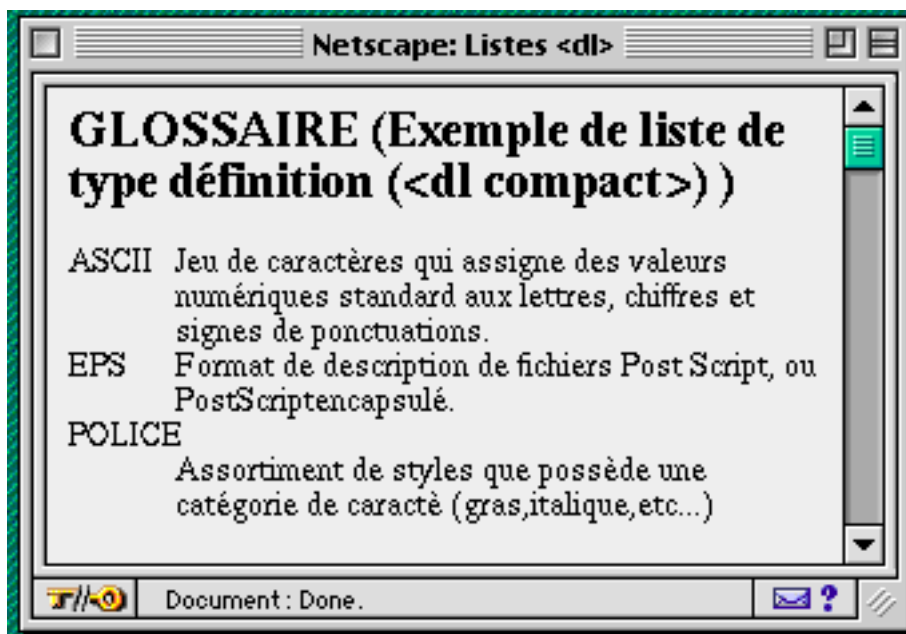
L'élément <DT> précède donc chaque élément de la liste, celui-ci ne devant pas excéder une ligne. Il est à noter que le système de liste ne gère pas automatiquement de mise en relief des caractères de l'élément. Il appartient à l'auteur du document d'utiliser des balises de styles ou de titre (<Hn>) pour effectuer cette opération.

### 7.6.2.2 <DD>

La balise <DD> correspond à la zone de définition de l'élément. La taille de cette zone n'est pas limitée et chacune des lignes sera décalée vers la droite. Cette balise est vide.

La structure de ce modèle de liste sera donc la suivante :

```
<DL>
  <DT>identification de l'élément
    <DD>Description de l'élément
  <DT>...
    <DD>...
</DL>
```



## 7.6.3 Listes régulières

### 7.6.3.1 <LI>

La balise <LI> est vide et commune pour tous les types de listes restant. Elle précédera chaque objet de la liste. La syntaxe générale des listes régulières sera :

```
<balise d'ouverture>
  <li>élément de l'énumération
  <li>..
<balise de fermeture>
```

### 7.6.3.2 <UL> </UL>

La balise <UL> permet de générer des listes non numérotées, chacun des éléments de la liste sera précédé d'une puce, symbole dont le graphisme pourra varier selon le niveau d'imbrication de la liste.

Les balises <LI> ou <UL> dispose d'une option qui permet de choisir la forme de la puce devant la sélection.

<UL TYPE=disc> donnera une puce

<UL TYPE=square> donnera un carré

<UL TYPE=circle> donnera un cercle vide

#### Exemple de liste de type non ordonnée imbriquées (<ul>)

- Fichiers HTML
  - Exemples
    - listes
    - tableaux
    - Styles
      - citation
      - texte en emphase
  - En cours
  - Périmés
- Fichiers images
- ....

### 7.6.3.3 <OL> </OL>

La balise <OL> est pour une liste ordonnée ou numérotée. Chaque balise <LI> va insérer le nombre qui sera affiché devant l'élément de la liste. Des options d'affichage permettent de modifier la numérotation des éléments de la liste.

<OL TYPE=1> donnera la numérotation par défaut de la liste.

<OL TYPE=a> donnera une numérotation alphabétique en minuscule (a, b, c, ...)

<OL TYPE=A> donnera une numérotation alphabétique en majuscule (A, B, C, ...)

<OL TYPE=ii> donnera une numérotation en lettre latine en minuscule (i, ii, iv, xci, ...)

<OL TYPE=II> donnera une numérotation en lettre latine en majuscule (I, II, IV, XCI, ...)

## 7.7 Styles

Les styles de présentation des textes dans HTML ne permettent pas de choisir une police de caractères particulière avec une taille précise pour ceux-ci. C'est toujours l'application ayant en charge l'exécution du code HTML, en l'occurrence le navigateur, qui décidera de la présentation et cela en fonction de ses possibilités et du paramétrage fait par le lecteur. Les éléments de style vont permettre d'attribuer à des mots ou à des paragraphes des attributs prédéfinis.

HTML définit deux méthodes pour la présentation des caractères : le style physique et le style logique.

### 7.7.1 Style physique

Les balises suivantes vont explicitement indiquer le type de caractères à utiliser.

#### 7.7.1.1 **<B>** **</B>**

Cette balise commence un texte en caractères gras.

#### 7.7.1.2 **<I>** **</I>**

Cette balise stylise un texte en caractères italiques.

#### 7.7.1.3 **<U>** **</U>**

Cette balise modifie un texte en caractères soulignés. (elle n'est pas reconnue par tous les navigateurs).

#### 7.7.1.4 **<TT>** **</TT>**

C'est une balise pour mettre le texte en caractères de type machine à écrire (Télétype). La police utilisée est de type non proportionnel.

#### 7.7.1.5 <STRIKE> ou <S> </S>

Cette balise va barrer le texte sélectionné ~~comme ceci~~.

#### 7.7.1.6 <BIG> </BIG>

Cette balise va agrandir la taille de la police affichée. Synonyme de <FONT SIZE=+1> </FONT>

#### 7.7.1.7 <SMALL> </SMALL>

Cette balise va réduire la taille de la police. Synonyme de <FONT SIZE=-1> </FONT>

#### 7.7.1.8 <SUB> </SUB>

Cette balise va placer le texte sélectionné en indice comme <sub>ceci</sub>.

#### 7.7.1.9 <SUP> </SUP>

Cette balise va placer le texte sélectionné en exposant comme <sup>cela</sup>.

#### 7.7.1.10 <FONT> </FONT>

Cette balise permet d'agir sur des blocs distincts de caractères situés sur une même ligne.

##### 7.7.1.10.1 L'attribut SIZE

Cet attribut permet de régler la hauteur des caractères à l'aide d'une échelle relative comprise entre 1 et 7 ou d'une valeur comprise entre -4 (équivalent à la taille 1) et +4 (équivalent à la taille 7).

#### 7.7.1.10.2 L'attribut COLOR

Cet attribut spécifie la couleur des caractères. On peut utiliser une définition selon le modèle RGB.

#### 7.7.1.10.3 L'attribut BASEFONT

Cette option détermine pour le document entier les attributs en couleur, taille, caractère de la police affichée par défaut dans le document.

#### 7.7.1.10.4 L'attribut FACE

Cette option permet de spécifier le nom d'une police autre que les deux polices de caractères autorisées jusque là. Il suffit de donner le nom de la police ou son substitut dans le cas où cette police ne serait pas présente sur l'ordinateur de l'internaute. Seules Netscape 3 et MS Internet Explorer 3 ou des versions supérieures reconnaissent cette option.

Ex: `<FONT COLOR="#F8C9AA" SIZE=4 FACE="ARIAL, HELVETICA, GENEVA" BASEFONT> texte  
</FONT>`

#### 7.7.1.11 La fonction **&nbsp;**;

Cette fonction insère dans le texte un espace insécable. En en ajoutant quatre à la suite, on obtient un alinéa.

### 7.7.2 **Style logique, style de paragraphe**

Comme précédemment l'interprétation par le navigateur des consignes concernant le style des paragraphes est tout à fait libre et les huit styles logiques définis dans HTML pour caractériser des portions de texte ne produiront pas forcément autant de représentations différentes. Il est néanmoins préférable d'utiliser ce balisage en choisissant le mode le mieux adapté au genre du texte: balise de citation pour une citation, balise de code pour un exemple de programme informatique, etc. Se conformer à cette règle est une garantie de bonne exploitation des fichiers HTML par les futurs navigateurs.

#### 7.7.2.1 **<CITE> </CITE>**

Cette balise est souvent utilisée pour citer, plus un ouvrage ou une référence, qu'une citation pour laquelle on préférera la balise `<BLOCKQUOTE>` ; des caractères italiques sont le plus souvent choisis par le navigateur.

#### 7.7.2.2 **<CODE> </CODE>**

La balise `<CODE>` est utilisée pour un exemple de programme informatique ; des caractères du type machine à écrire avec police non proportionnelle (typiquement une police courier) seront sélectionnés.

#### 7.7.2.3 **<DFN> </DFN>**

On utilise `<DFN>` pour la définition d'un terme. Selon le navigateur, des caractères standards, gras ou gras et italiques seront utilisés.

#### 7.7.2.4 **<EM> </EM>**

La balise `<EM>` permet la mise en valeur d'un texte ; des caractères italiques seront les plus souvent employés.



#### 7.7.2.5 <KBD> </KBD>

Pour illustrer une saisie au clavier ; des caractères de type machine à écrire avec police non proportionnelle (comme une police courier) seront utilisées.

#### 7.7.2.6 <SAMP> </SAMP>

Pour un exemple de texte on utilise <SAMP> ; des caractères type machine à écrire avec police non proportionnelle (comme la police courier) seront généralement employées.

#### 7.7.2.7 <STRONG> </STRONG>

Pour mettre un texte très fortement en valeur on utilise la balise <STRONG> ; utilisation de caractères gras.

#### 7.7.2.8 <VAR> </VAR>

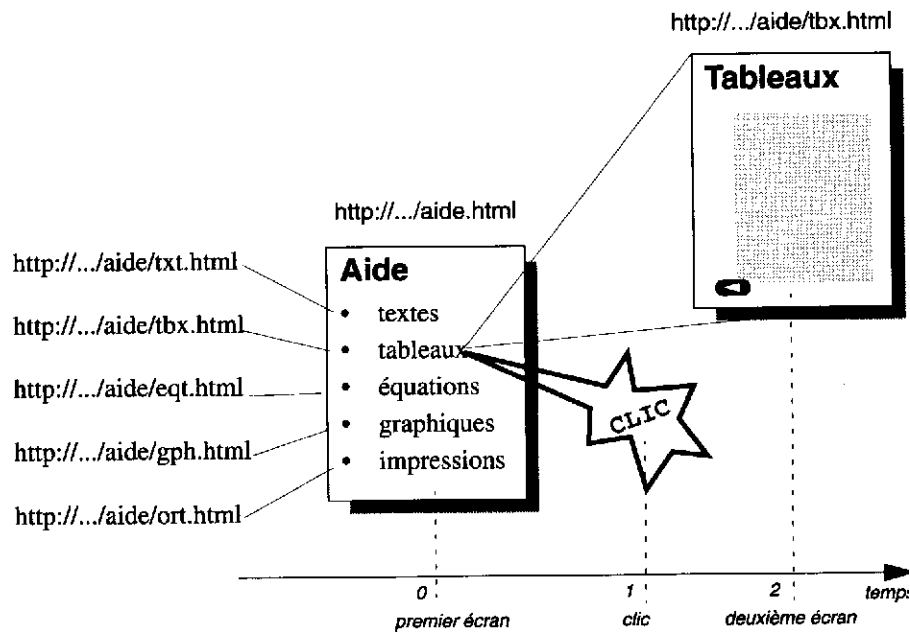
Cette balise est utilisée pour représenter une variable dans un contexte informatique par exemple ; utilisation de caractères italiques.

#### 7.7.2.9 <BLOCKQUOTE> </BLOCKQUOTE>

<BLOCKQUOTE> peut être utilisé dans le même esprit que la balise <CITE> mais des caractères standards seront utilisés. Le bloc de texte sera entièrement décalé vers la droite et des espaces de type paragraphe seront générés de part et d'autre du bloc. C'est d'ailleurs le seul attribut de cette liste véritablement intéressant. En effet, les autres styles logiques trouvent des équivalents dans les styles physiques (gras, italique, souligné, ...), tandis qu'ici, cette balise décale vers la droite tout un bloc de paragraphe contenant aussi bien du texte que des images. Elle est donc utilisée au-delà de sa définition officielle

## 7.8 Créer des liens

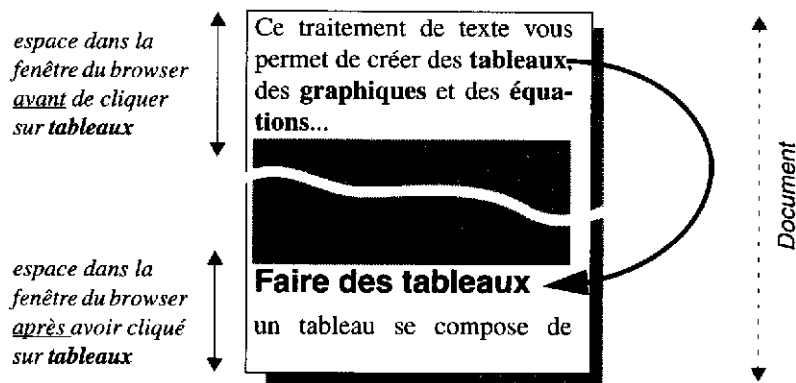
Le lecteur explore un document sur le Web en cliquant sur des zones actives, faisant ainsi apparaître de nouveaux documents. Dans HTML une zone active peut correspondre à un caractère, un mot ou un groupe de mots, une image ou une portion d'image. Le principe sera toujours identique : associer à cette zone active l'URL (voir le protocole d'adressage des documents l'URL) du document qui se substituera au document affiché lorsque l'on cliquera sur cette zone.



Le langage HTML définit la notion importante d'ancrage pour spécifier le départ et l'arrivée d'un lien hypertexte.

- l'ancrage de départ est cette zone active sur laquelle va cliquer le lecteur pour appeler une nouvelle page.
- l'ancrage d'arrivée est une zone inactive spécifiant le point d'arrivée d'un lien hypertexte. C'est donc une adresse de destination.

Il existe un autre type d'ancre d'arrivée lorsque la zone cliquable d'un document provoque un déplacement dans le même fichier. Cette ancre d'arrivée ne comporte pas d'attributs visuels et n'est évidemment pas une zone cliquable, d'où son nom d'ancre inactive. Les liens internes sont très intéressants, lorsqu'un document est trop long pour tenir totalement dans la fenêtre du navigateur. Cela permet de se déplacer d'autant plus rapidement dans le document que celui-ci, lu en une seule fois, se trouve en mémoire de l'ordinateur.



### 7.8.1 Définition d'une ancre

<A> </A>

Cette balise ne sera jamais utilisée seule.

### 7.8.1.1 L'attribut HREF, ancre de départ vers un lien externe

Il s'agit ici de créer un lien vers un serveur situé quelque part sur l'Internet, ou sur un document proposé par ce serveur. Par exemple dans une page du serveur de la ville d'Arles on trouvera un lien sur le fichier du serveur du ministère de la culture proposant les œuvres de Van Gogh.

La balise <A> spécifie l'attribut HREF dont la valeur précise l'URL du document à atteindre :

```
<A HREF="url-de-destination">zone_cliquable_avec_attributs_visuels</A>
```

### 7.8.1.2 L'attribut HREF, ancre de départ vers un lien interne

Dans ce cas on se propose de mettre un lien hypertexte vers un point précis du fichier en cours d'exécution : par exemple à la ligne 15 du fichier on propose un lien pour sauter directement à la ligne 95. Il faut donc placer une ancre active (ancre de départ) à la ligne 15 et une ancre inactive (ancre d'arrivée) à la ligne 95.

```
<A HREF="#étiquette">zone-cliquable-avec-attributs-visuels</A>
```

définit l'ancre de départ.

### 7.8.1.3 L'attribut HREF, ancre de départ vers un point spécifique d'un lien externe

On référence dans ce cas une entrée particulière d'un document externe :

```
<A HREF="url-destination#étiquette">zone-cliquable-avec-attributs-visuels</A>
```

### 7.8.1.4 L'attribut NAME, ancre d'arrivée

```
<A NAME="label">zone-NON-cliquable-SANS-attributs-visuels</A>
```

définit l'ancre d'arrivée, lieu que l'on pourra atteindre en cliquant sur une ancre de départ. On notera dans l'ancre de départ, le caractère # (dièse) indiquant que la référence hypertexte n'est pas une URL mais une étiquette.

### 7.8.1.5 L'attribut HREF et NAME sur une même ancre

Application à la note de bas de page

Une ancre peut admettre simultanément plusieurs attributs. Nous allons utiliser l'analogie avec une note de bas de page dans un texte imprimé pour étudier l'utilisation des attributs HREF et NAME spécifiés sur la même ancre. La note de bas de page est indiquée au lecteur par un petit chiffre en exposant <sup>1</sup>, comme ceci. Cette méthode fonctionnera très bien et le lecteur, lorsqu'il cliquera sur l'ancre active provoquera bien le saut vers la note.

Il reste maintenant à lui assurer un retour aisé vers le point d'interruption de sa lecture, car ce point n'est plus affiché sur l'écran si la page HTML est longue.

- dans le texte courant on place une ancre active et passive de la forme :

```
...<A HREF="#note1" NAME="retour1"> (1) </A>...
```

- de même en-tête du bloc de texte composant la note :

```
<A HREF="#retour1" NAME="note1"> 1 </A> Texte de la note...
```

Ainsi le lecteur clique sur l'ancre (1), va lire l'annotation, puis il clique sur l'ancre 1 et retrouve le fil de sa lecture.

L'extrémité d'un lien peut être n'importe quel type de document :

- un fichier HTML
- une image GIF, JPEG, un film, animation,
- un texte

---

<sup>1</sup> Vous lisez une note de bas de page.

### 7.8.1.6 D'autres URL

La découverte de l'Internet pour le grand public a été favorisée par le WEB et son interface graphique. Si HTTP est le protocole du WEB, d'autres services sont utilisés sur l'Internet depuis sa création :

- FTP pour le transfert de fichiers entre ordinateurs connectés à l'Internet
- SMTP pour l'échange de courrier électronique
- NNTP pour les forums d'Internet

etc.

La spécification d'accès direct à un fichier sera :

```
ftp://username;password@nom-du-serveur/repertoire/nom-du-fichier
```

```
news:nom.du.groupe.de.news
```

```
mailto:nom@site.fr
```

On peut ajouter des attributs supplémentaires à mailto comme adjoindre un sujet prédéfini en suivant cet exemple:

```
mailto:ralph.davidovits@insset.u-picardie.fr?Subject=Titre du message
```

### 7.8.2 Adressage des URL dans l'espace du serveur

Dans la conception d'une page HTML si l'on ne référence que des URL de documents résidant dans le même espace, alors il est possible de décrire chaque appel vers un nouvel URL par rapport à l'URL où l'on se situe. Ce mode d'adressage est nommé chemin relatif à l'opposé du chemin absolu qui commence par `http://www...`

Si on décrit un lien vers un fichier résidant dans le même répertoire, il est suffisant d'indiquer le nom du fichier. Par contre si dans un fichier on décrit un lien vers un fichier d'un autre répertoire, il faudra décrire l'arborescence vers ces répertoires. La navigation vers les fichiers et les répertoires se fait comme sous le DOS sauf qu'il faut mettre un slash / normal et pas un anti-slash \.

### 7.8.2.1 Exemples:

Pour aller charger le fichier `essai.html` qui se trouve dans le même répertoire que le fichier affiché actuellement dans le serveur Web, comme URL le nom du fichier est suffisant: `essai.html`

En adresse absolue cela donne par exemple:

```
http://www.domaine.fr/boite/repertoire/essai.html
```

Nous garderons cette adresse comme point de départ pour illustrer la navigation dans les exemples ci-après.

Pour aller charger le fichier `texte.html` qui se trouve dans le répertoire « dossier » en aval, il faut faire comme suit: `dossier/texte.html`

Par rapport à l'exemple, en adresse absolue cela donne:

```
http://www.domaine.fr/boite/repertoire/dossier/texte.html
```

Pour aller charger le fichier `fichier.html` qui se trouve dans un dossier plus en amont dans l'arborescence il faut le faire précéder de 2 points (`..`): `../fichier.html`

Par rapport au premier exemple, en adresse absolue cela donne:

```
http://www.domaine.fr/boite/fichier.html
```

Pour aller chercher un fichier ou un dossier en commençant par la « racine » c'est-à-dire le sommet de l'arborescence, il faut le faire précéder d'un simple `/`: `/source/feuille.html`

Par rapport au premier exemple, en adresse absolue cela donne:

```
http://www.domaine.fr/source/feuille.html
```

## **7.9 Inclusion d'images**

### **7.9.1 Les images en ligne**

On entend par image en ligne une image dont l'affichage se fait automatiquement dans la fenêtre du navigateur lors du transfert de la page HTML. Cet affichage se fait de façon quasi simultanément avec l'apparition du texte de la page. L'illustration trouve sa place dans le document HTML dans les mêmes conditions qu'une photo sur un page imprimée.

### **7.9.2 Les formats des fichiers Web**

La compression est essentielle pour réaliser des images Web aussi "légères, que possible". La compression est généralement associée à une perte de qualité. Les images affichées ne seront donc pas l'exacte réplique de ce qu'elles étaient lors de leurs conceptions.

Les deux formats d'images les plus couramment acceptés par les navigateurs sont GIF et JPEG. Une grande différence existe entre ces deux formats. Les images JPEG peuvent être codées sur 24 bits et peuvent donc contenir jusqu'à 16 millions de couleurs. Au contraire, les images GIF sont toujours codées sur 8 bits et contiennent donc un maximum de 256 couleurs.

La conversion d'un autre format d'image (tel que PICT, TIFF ou EPS) vers JPEG et GIF est aisée si vous possédez le logiciel de conversion approprié.

### **7.9.3 Le format JPEG**

Le format JPEG est remarquablement bien adapté au Web. En effet, tout en produisant des fichiers de faible poids, il permet de travailler en 24 bits, c'est-à-dire en



16,7 millions de couleurs. Pour ce faire, le format JPEG utilise un procédé de compression avec perte de qualité capable de réduire jusqu'à cent fois la taille des images.

### **7.9.3.1 La compression JPEG**

Les images JPEG sont compressées lors de leur sauvegarde. Inversement, elles sont décompressées juste avant leur affichage dans le navigateur.

Un dérivé de ce format affiche les images en plusieurs passes qui correspondent à différents niveaux de qualité. Il est baptisé JPEG Progressif. L'utilisateur peut donc avoir un aperçu de l'image avant qu'elle ne soit totalement chargée.

### **7.9.3.2 Quand utiliser le format JPEG ?**

Le format JPEG est particulièrement bien adapté aux images photographiques qui, par nature, possèdent de nombreux dégradés de couleurs. La compression JPEG donne de bons résultats lorsqu'elle est appliquée sur des images pour lesquelles les changements de couleurs sont très progressifs, comme par exemple, de subtiles variations de lumière. Elle n'est, par conséquent, pas bien adaptée aux images constituées de multiples zones monochromatiques, telles les illustrations que l'on rencontre dans des bandes dessinées et dans des clip-arts. Aussi, la compression JPEG n'est pas adaptée aux images qui contiennent du texte ou des formes non lissées.

En utilisant un logiciel de dessin adapté, vous pouvez choisir le niveau de compression (faible, moyen, haut, maximum) et décider ainsi de visu quelle dégradation maximale vous êtes prêt à accepter.

### **7.9.4 Le format GIF**

Par définition, les fichiers au format GIF contiennent un maximum de 256 couleurs. Le format GIF utilise la compression LZW qui permet de réduire la taille des fichiers sans perte de qualité. Les fichiers GIF sont de deux types: GIF87a et GIF89a.

Le format GIF89a fait partie du format GIF87a. Il permet d'attacher une information complémentaire au fichier, comme par exemple la couleur de transparence. Dans ces images, certaines régions peuvent être masquées pour devenir transparentes. Vous entendrez également parler d'images GIF entrelacées. Elles apparaissent en plusieurs passes, par augmentation progressive de la résolution. Cela donne le sentiment que l'image apparaît plus vite (mais cela reste une impression !).

#### **7.9.4.1 La compression GIF**

Le procédé de compression GIF est intéressant. Il consiste à mémoriser les transitions des pixels à l'intérieur de l'image. Chaque changement de couleur sur une même ligne élémentaire est mémorisé et augmente donc la taille du fichier.

#### **7.9.4.2 Les images GIF et le nombre de couleurs**

Avant de pouvoir sauvegarder une image au format GIF, vous devez, le cas échéant, diminuer à 256 le nombre de couleurs qui la composent pour ne pas dépasser les 8 bits impartis au codage de chaque couleur. Mais un problème survient. En effet, la palette de 256 couleurs n'est pas la même entre les Macintosh et les PC sous Windows. Seules 216 couleurs sont communes. Pour éviter un effet de postérisation (une couleur sera remplacée par une autre plus proche) dans la fenêtre d'affichage du navigateur Web, il est nécessaire de créer cette palette spéciale afin de ne pas avoir de mauvaise surprise. Cette palette porte le nom de CLUT et la plupart des logiciels de dessin savent la gérer (Photoshop, GraphicConverter, etc.). Sur Internet, cette palette au format Photoshop est disponible pour Mac et PC, et même en image avec ses correspondances RGB pour la créer à la main, s'il le fallait, ou pour voir à quoi cela correspondrait.

Le fichier CLUT pour PC: <ftp://luna.bearnet.com/pub/lynda/BCLUT2.ZIP>

Le fichier CLUT pour MAC: <ftp://luna.bearnet.com/pub/lynda/newCLUT.hqx>

L'image du fichier CLUT: <ftp://luna.bearnet.com/pub/lynda/nhue.gif>

Afin d'abaisser encore la taille de l'image en Ko, on peut réduire le nombre de couleurs à afficher. Le document sera donc chargé plus rapidement. Certains logiciels de dessin vous proposent d'optimiser automatiquement cette palette de couleur. D'autres ont besoin que l'on intervienne manuellement en descendant le fichier à 7 bits (128 couleurs), 6 bits (64 couleurs), 5 bits (32 couleurs), 4 bits (16 couleurs), 3 bits (8 couleurs), 2 bits (4 couleurs) voire 1 bit (noir et blanc). En descendant progressivement le nombre des couleurs, on remarquera que la qualité du fichier est parfois peu altérée et le gain en Ko non négligeable. Ce procédé intéresse surtout les grandes images.

La méthode de compression GIF donne de bons résultats sur des images comportant peu de changements de couleurs. Au contraire, elle ne convient pas à la compression d'image ayant subi un lissage, car ce dernier introduit du bruit qui augmente la taille de l'image dans de larges proportions.

#### **7.9.4.3 En résumé**

- Le format JPEG est bien adapté aux images photographiques, aux images en niveaux de gris, aux images scannées et autres images contenant des dégradés de couleurs.
- Le format GIF est bien adapté aux simples graphiques dessinés sur l'ordinateur, aux dessins au trait, et autres images qui ne comportent pas de trop nombreux changements de couleurs.

Enfin, il est plus pratique d'enregistrer ces images dans les deux formats et de comparer leurs tailles respectives en Kilo-octets pour choisir la meilleure. Pour le format JPEG, la niveau de compression « moyen » est un bon compromis entre taille du fichier et altération de la qualité de l'image.

### 7.9.5 <IMG>

C'est la balise permettant d'inclure une image. Cette balise sera toujours complétée par l'attribut SRC qui permet de donner l'adresse du fichier graphique contenant l'image :

```
<IMG SRC="/repertoire/sous-repertoire/nom-du-fichier-graphique">
```

#### 7.9.5.1 Alignement des images

La balise IMG admet un attribut de présentation permettant de situer l'image par rapport à la ligne de texte courante. Il s'agit de l'attribut ALIGN dont les valeurs peuvent être :

- ALIGN=TOP pour aligner le haut de l'image sur la ligne courante
- ALIGN=MIDDLE pour aligner le milieu de l'image sur la ligne courante
- ALIGN=BOTTOM pour aligner le bas de l'image sur la ligne courante
- ALIGN= LEFT et RIGHT font flotter l'image à gauche ou à droite du texte.
- WIDTH=l et HEIGHT=H où l et H représentent la largeur et la hauteur de l'image en pixels.
- BORDER=n où n représente l'épaisseur du trait bordant l'image. La valeur 0 supprime la bordure.
- HSPACE=n où n représente le nombre de pixels séparant horizontalement l'image du texte ou d'une autre image qui lui serait accolé.
- VSPACE=n ; même chose que précédemment mais pour la verticale.

#### 7.9.5.2 Les images et les navigateurs non graphiques

L'attribut ALT accompagné du texte de substitution remplit cette fonction :

```
<IMG SRC="photo.gif" ALT="Photo du cheval Camarguais">
```

En lieu et place de la photo, le lecteur trouvera ce texte qui pourra aider à une meilleure compréhension du document.

### 7.9.5.3 Les images externes

Ce type d'image n'apparaît pas à l'écran lors du chargement de la page. Elle est proposée par un lien hypertexte dont l'extrémité pourra être une image GIF, JPEG ou autre.

Vous pouvez `<a href="venus.jpg"> cliquer ici </a>` pour voir le tableau "la naissance de V&eacute;nus de Botticelli".

### 7.9.5.4 Les images en guise d'ancre

On peut tout à fait remplacer le texte d'une ancre par une balise définissant une image.

```
<a href="venus.jpg"></A>
```

Dans ce cas l'image est encadrée par un filet de couleur pour indiquer qu'il s'agit d'un lien hypermédia sur lequel on peut cliquer. C'est cette méthode qui est utilisée pour mettre des boutons de navigation dans les pages HTML.

## 7.10 Fond de page, couleur de texte

Un certain nombre d'attributs de la balise <BODY> permettent de contrôler la couleur du fond de la fenêtre du navigateur, la couleur des caractères du texte, et enfin la couleur des liens.

### 7.10.1 Les fonds d'écran

#### 7.10.1.1 L'attribut BGCOLOR

Cet attribut va permettre de choisir une couleur pour le fond de la page. Il peut être utilisé largement car à l'inverse de l'attribut BACKGROUND, il n'est pas consommateur de réseaux.

```
<BODY BGCOLOR="#rrggbb">
```

“rr”, “gg”, “bb” sont des valeurs hexadécimales comprises entre 00 et FF qui spécifient le degré de saturation des couleurs, rouge, verte et bleu dont le mélange produit la couleur souhaitée.

00 représente le minimum et FF le maximum pour une des composantes. Ainsi la valeur FF0000 donne un rouge avec une intensité maximum, FFFFFFFF du blanc, 000000 du noir, etc.

```
<BODY BGCOLOR="#FFFFFF">
```

va donner une page HTML sur un fond blanc alors que par défaut le fond de page est gris.

#### 7.10.1.2 L'attribut BACKGROUND

Avec cet attribut nous allons spécifier non plus une couleur mais une image ; celle-ci résidant sur le serveur, on va augmenter le temps de chargement de la page vers le navigateur.

```
<BODY BACKGROUND="fichier-graphique.gif">
```

fichier-graphique.gif représente l'URL de l'image sur le serveur

## 7.10.2 La couleur du texte

### 7.10.2.1 L'attribut TEXT

Cet attribut permet de contrôler la couleur du texte standard, c'est-à-dire tout texte ne spécifiant pas un lien. Le codage de la couleur se fait de la même façon que pour l'attribut BGCOLOR

```
<BODY TEXT="#rrggbb">
```

### 7.10.2.2 Les attributs LINK, VLINK et ALINK

Ces trois attributs permettent de contrôler la couleur des liens (soit du texte, soit du cadre autour d'une image).

couleur des liens non visités : 

```
<BODY LINK="#rrggbb">
```

couleur des liens visités : 

```
<BODY VLINK="#rrggbb">
```

couleur du lien au moment du clique : 

```
<BODY ALINK="#rrggbb">
```

Exemple de couleurs :

azur	00FFFF
orange	FFBE1D
rose	FF86B9
or	FFD833
gris	939393
violet	AE61FF
lavande	A4A4FF
jaune	FFFF00

D'autre part, certaines couleurs peuvent être écrites avec leur nom anglais car les navigateurs Web les ont intégrés. Ces 16 couleurs ont pour nom: Black, Silver, Gray, White, Maroon, Red, Purple, Fuschia, Green, Lime, Olive, Yellow, Navy, Blue, Teal et Aqua.

Par exemple: `<BODY BGCOLOR="White">`

## 7.11 Images cliquables

A plusieurs reprises nous avons dit que l'on pouvait associer une URL à une portion d'images et qu'en conséquence cliquer sur cette zone de l'image pouvait permettre la connexion vers cette URL. La correspondance entre différentes zones d'une image et différentes URL est nommée image cliquable ou encore image réactive.

Pour réaliser une image cliquable, on commencera par mettre l'image à la taille souhaitée. En effet la seconde opération va consister à faire un relevé des coordonnées zone par zone des surfaces cliquables.

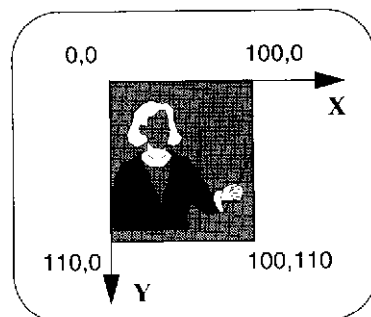
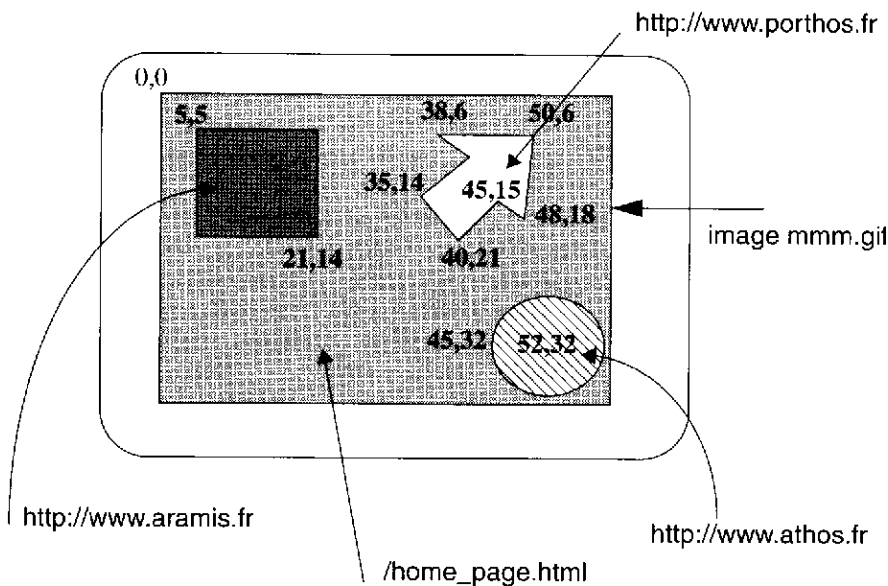


Image de largeur 100 pixels  
et de hauteur 110 pixels

Coordonnées d'une image





### 7.11.1 Système de coordonnées

Une image se mesure en points ou pixels et l'origine de ces coordonnées se situe en haut et à gauche de l'image.

#### 7.11.2 Réalisation de la carte des coordonnées URL (ancienne méthode)

Le premier travail va donc consister à créer un fichier dont chaque ligne va définir une forme géométrique pour la zone cliquable et son URL associée.

Une image cliquable sera lancée par l'ensemble de balises suivant :

```
<A HREF="/cgi-bin/imagemap/nom_symbolique">
<IMG SRC="url-du-fichier.gif" ISMAP>
</A>
```

/cgi-bin/imagemap/ est un programme devant être préalablement installé dans le serveur Web pour pouvoir proposer des images cliquables.

On crée alors un fichier texte avec le suffixe « .map » dans lequel on placera toutes les coordonnées et les URL correspondants. Il faudra placer ce logiciel à un endroit approprié du serveur . Ce fichier « .map » ressemblera à cela:

```

# Created by WebMap 1.0.1
# Vendredi 10 Mai 1996 at 1:41
# Format: NCSA

default http://www.insset.u-picardie.fr/geopolymer/index.html
rect http://www.insset.u-picardie.fr/geopolymer/home.html 73,0 283,29
rect http://www.insset.u-picardie.fr/geopolymer/directory.html 0,30 164,74

```

Ce fichier de relations s'écrit de la façon suivante:

FORME URL X1, Y1 X2, Y2 ...

où le couple  $X_n, Y_n$  définit les coordonnées d'un point de la forme utilisée, X représente l'abscisse et Y l'ordonnée.

Il existe trois types de formes géométriques pour définir une zone cliquable:

circle \_url\_ centre-X, perimetre-Y

...le point X spécifie le centre du cercle et Y un point quelconque situé sur le périmètre.

rect \_url\_ gauche-X, sommet-Y droit-X, base-Y

...le premier couple spécifie le sommet supérieur gauche le deuxième couple spécifie le sommet inférieur droit.

poly \_url\_ X1,Y1 X2,Y2 ...Xn,Yn

...chaque point définit l'extrémité d'un vecteur composant le polygone.

default \_url\_

...indique l'URL par défaut qui sera appliqué si l'on clique en dehors des zones indiquées. Le plus souvent se sera l'URL de la page dans laquelle est insérée cette image cliquable.

L'unité de mesure utilisée est le pixel.

Mais cette **méthode devient obsolète** car il faut créer un fichier « .map » distinct de la page HTML et c'est le serveur qui manipule l'ensemble.

### 7.11.3 Description des coordonnées dans le fichier HTML (nouvelle méthode)

Pour une question de rapidité, les images cliquables sont désormais exécutables directement depuis le logiciel Web, d'où un gain de temps et de facilité de création.

Donc, HTML 3 permet la réalisation de cartes cliquables où seul le client intervient :

```
<IMG SRC=url-du_fichier.gif USEMAP=#nom>
```

USEMAP définit l'adresse d'une zone de code HTML dans laquelle sont décrites les surfaces cliquables et les URL associées. Cette zone débute par la balise <MAP NAME="nom" pour se terminer par la balise </MAP>.

Entre ces deux balises on définit les zones cliquables par des balises <AREA SHAPE="figure" HREF="url" COORDS="coordonnées">

Les coordonnées des figures (RECT, POLY, CIRCLE) sont toujours exprimées en pixel et reprennent la même méthode de définition par abscisses et ordonnées vue plus haut. A la différence du fichier « .map », toutes les coordonnées sont séparées par une virgule; il n'y a plus d'espace pour séparer un couple de coordonnées.

La balise <AREA SHAPE="default" HREF="url"> indique l'URL par défaut qui sera appliqué si l'on clique en dehors des zones indiquées. Le plus souvent se sera l'URL de la page dans laquelle est insérée cette image cliquable.

L'exemple de tout à l'heure ressemble maintenant à cela:

```
<IMG ALIGN=Middle SRC="../geopolymer/images/arborescence.gif" BORDER=0  
WIDTH="547" HEIGHT="171" USEMAP="#arborescence">
```

```
<MAP NAME=arborescence>  
  <AREA SHAPE=rect COORDS="73,0, 283,29" HREF=home.html>  
  <AREA SHAPE=rect COORDS="0,30, 164,74" HREF=directory.html>  
<AREA SHAPE=default HREF=index.html>  
</MAP>
```

## 7.12 Les formulaires

Les formulaires ou Forms sont la réponse apportée par le langage HTML pour générer sur l'écran des zones de dialogue avec le lecteur. Comme dans un formulaire papier on va pouvoir réaliser des zones dans lesquelles on entrera un texte, des cases à cocher, des listes de choix, etc... On peut aussi concevoir une recherche dans une base de données, un dictionnaire. Par exemple: le lecteur frappe dans une page HTML le mot à rechercher, puis reçoit dans la page suivante la définition du mot.

Mais il faudra toujours un élément approprié pour traiter l'information qui y est saisie. Cet élément sera presque toujours un programme s'exécutant dans un espace du serveur réservé à cet effet : le CGI ou Common Gateway Interface.

### **Important:**

La création des formulaires et de leurs attributs en langage HTML dépend du programme CGI qui va traiter les informations envoyées. C'est en lisant les instructions attachées à ce logiciel que l'on saura quoi mettre.

### 7.12.1 <FORM> </FORM>

La balise <FORM> sera complétée au minimum des 2 attributs suivants :

#### 7.12.1.1 METHOD,

auquel peut être donné la valeur POST ou la valeur GET qui définit le mode de transfert des données vers le script,

#### 7.12.1.2 **ACTION,**

qui définit l'URL d'un programme (script) chargé de traiter les données acquises depuis le formulaire.

#### 7.12.2 **L'attribut NAME**

Il permet de définir le nom qui, lors de l'envoi des données vers le script, va permettre à celui-ci d'identifier la provenance des données. Le nom attribué doit être unique dans le formulaire, sauf dans le cas d'un ensemble composé de plusieurs éléments (checkbox ou bouton radio).

#### 7.12.3 **L'attribut VALUE**

- spécifie un texte à écrire sur les boutons reset et submit (information visuelle).
- pré-remplit un champ textarea (information visuelle).
- attribue une valeur à l'option pour les boutons radio et checkbox (information pour le script). La valeur spécifiée par NAME identifie le bloc de boutons.

#### 7.12.4 **<INPUT>**

La balise <INPUT> va servir à définir des champs pour entrer un texte et des boutons permettant de choisir des options. Sans attribut autre que NAME, la balise va définir une zone d'entrée de texte simple.

##### 7.12.4.1 **L'attribut TYPE**

L'attribut TYPE associé à la balise <INPUT> permet le choix de l'élément d'entrée.

- le type TEXT permet de rentrer du texte. Comme l'attribut PASSWORD, on peut

ajouter deux attributs supplémentaires. `SIZE="n"` où `n` est le nombre de caractères donnant la longueur de la zone de saisie et `MAXLENGTH="n"` donnant le nombre maximal de caractères à saisir.

- le type `PASSWORD` permet de saisir un mot de passe de façon confidentielle,
- le type `SUBMIT` déclenche l'envoi du formulaire vers le script; pas besoin de `NAME`,
- le type `RESET` permet d'effacer les données déjà entrées; pas besoin de `NAME`,
- le type `CHECKBOX` permet de faire un bloc de boutons permettant un choix multiple d'options (fonction logique et exclusif),
- le type `RADIO` permet de faire un bloc de bouton permettant un choix exclusif parmi plusieurs options (fonction logique ou exclusive). Cliquer sur une option désélectionne automatiquement les autres. L'attribut `CHECKED` a pour effet de positionner le bouton radio en mode enfoncé (ou coché pour l'attribut `CHECKBOX`).
- le type `HIDDEN` sert pour passer des données acquises dans un formulaire à un autre formulaire et cela sans que rien n'apparaisse à l'écran.

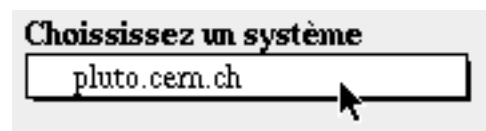
#### 7.12.5 <SELECT>

La balise `<SELECT>` permet de générer des listes à choix simple (ou exclusif) ou à choix multiple (et exclusif). Elle se programme comme une liste où chacun des éléments est spécifié par la balise `<OPTION>`. De la présence de l'attribut `SIZE` va dépendre la présentation de la liste. Si la valeur donnée à l'attribut `size` est inférieure à 2 ou si l'attribut `size` est omis, la liste est interprétée en menu déroulant (pop-list). Dans le cas contraire la liste s'affiche dans une fenêtre à ascenseurs. La valeur donnée à l'attribut `size` donne alors le nombre de lignes visibles dans la fenêtre. L'option à choix multiple découle de la présence de l'attribut `MULTIPLE`.

Le code HTML ci-après:

```
<select name="site">
  <option>pluto.cern.ch
  <option>mickey.ibp.fr
  <option>donald.u-picardie.fr
</select>
```

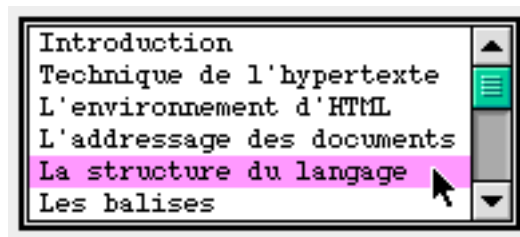
donne l'affichage suivant :



Programmation d'une liste avec sélection unique sans sélection par défaut :

```
<select name="menu" size=6>
  <option>Introduction
  <option>Technique de l'hypertexte
  <option>L'environnement HTML
  <option>L'adressage des documents
  <option>La structure du langage
  <option>Les balises
</select>
```

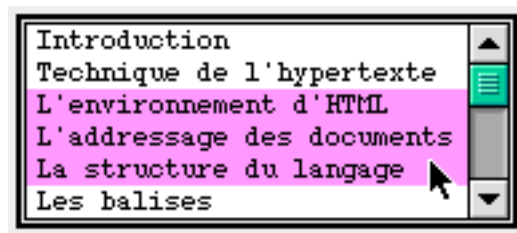
L'affichage produit :



Si on modifie la première ligne avec l'introduction de l'attribut *multiple* :

```
<select name="menu" size=6 multiple>
```

cela permettra ce type de sélection :



### 7.12.6 <TEXTAREA>

La balise <TEXTAREA> permet de créer une fenêtre avec ascenseurs horizontaux et verticaux dans laquelle on pourra saisir du texte. La valeur donnée aux attributs ROWS=n (lignes) et COLS=n (colonnes) délimite la taille de cette fenêtre.

### 7.12.7 Envoi de fichier

Enfin, on peut envoyer vers le serveur un fichier qui se trouve sur le l'ordinateur du client. Le programme CGI devra être capable de traiter ce fichier car il sera encodé d'une certaine manière. Pour cela, il faut créer un nouveau formulaire qui ressemblera à cela:

```
<FORM ENCTYPE="multipart/form-data" ACTION="URL-du-programme-CGI" METHOD=POST>
  Envoyer ce fichier <INPUT NAME="fichier_client" TYPE="file">
<INPUT TYPE="submit"> </FORM>
```

L'encodage par défaut (ENCTYPE="application/x-www-form-urlencoded") des données du formulaire envoyées vers le programme CGI ne convient pas ici.

### 7.12.8 Exemple de formulaire envoyant le résultat par e-mail:

```
<FORM ACTION="mailto:code@domaine.fr" METHOD="POST" ENCTYPE="text/plain">
Identifiant:
  <INPUT NAME="identite" TYPE=Text SIZE="25" MAXLENGTH="35">
<BR>
Mot de passe:
  <INPUT NAME="mot_de_passe" TYPE>Password SIZE="10" MAXLENGTH="15">
<P>
  <INPUT NAME="media" TYPE=Radio VALUE="cd" CHECKED>CD-ROM<BR>
  <INPUT NAME="media" TYPE=Radio VALUE="dk">Disquette<BR>
<P>
  <INPUT NAME="micro" TYPE=Checkbox VALUE="mac" CHECKED>Macintosh<BR>
  <INPUT NAME="micro" TYPE=Checkbox VALUE="pc">PC<BR>
  <INPUT NAME="micro" TYPE=Checkbox VALUE="unix">UNIX<BR>
  <INPUT NAME="cache" TYPE=Hidden VALUE="renseignement">
  <TEXTAREA NAME="commentaire" ROWS=10 COLS=50 WRAP=Virtual>Entrez ici vos
commentaires et remarques</TEXTAREA>
<P>
  <SELECT NAME="diplome" SIZE=1>
    <OPTION>Bac
    <OPTION>Deug
    <OPTION>Licence
    <OPTION>Ma&icirc;trise
    <OPTION>3&egrave;me Cycle
  </SELECT>
  <SELECT NAME="diplome2" SIZE=5 MULTIPLE>
    <OPTION>Bac
    <OPTION>Deug
    <OPTION>Licence
    <OPTION>Ma&icirc;trise
    <OPTION>3&egrave;me Cycle
    <OPTION>DEA
    <OPTION>DESS
    <OPTION>Doctorat
    <OPTION>Autre
  </SELECT>
<P>
  <INPUT TYPE=Submit VALUE="Valider !">
  <INPUT TYPE=Reset VALUE="Effacer">
</FORM>
```



## 7.13 Les Tableaux

HTML 3 permet de réaliser des tableaux avec réglage de l'encadrement, de la taille et de l'espacement des cellules. Ce chapitre décrit les principales balises permettant de présenter un tableau.

La structure d'un tableau est tout à fait comparable à la définition des listes. On définit une balise de début de tableau, puis on décrit le tableau ligne par ligne et enfin on indique la balise de fin de tableau. Cette structure est suffisamment simple pour que le tableau soit facile à maintenir (modification du nombre de lignes et de colonnes).

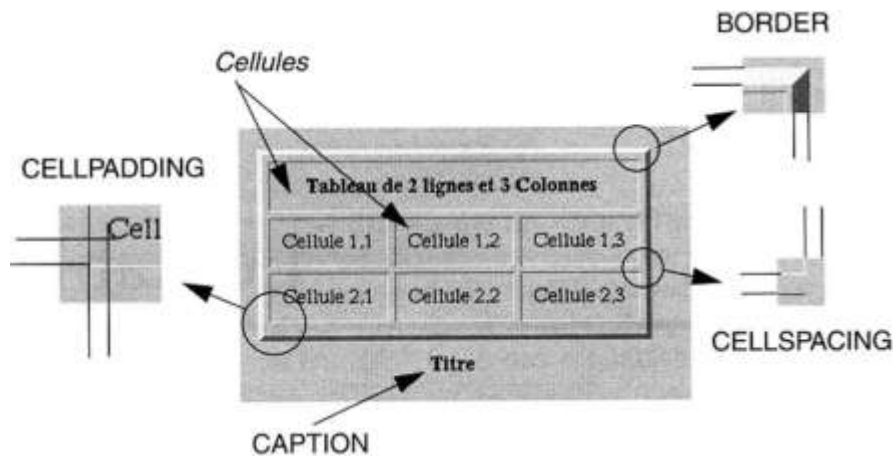
Mais dans le langage HTML, le tableau permet d'être utilisé de 3 manières:

1. Pour afficher des données dans un tableau classique,
2. Pour encadrer un texte, une image, etc.
3. Pour placer des éléments de manière précise dans l'espace d'une page Web.

Une cellule peut contenir les éléments suivants :

- texte
- listes
- images
- liens hypertextes
- élément de formulaire
- un autre tableau

La représentation du tableau est actuellement très dépendante du navigateur utilisé.



Dénomination des éléments dans un tableau HTML

### 7.13.1 <TABLE> </TABLE>

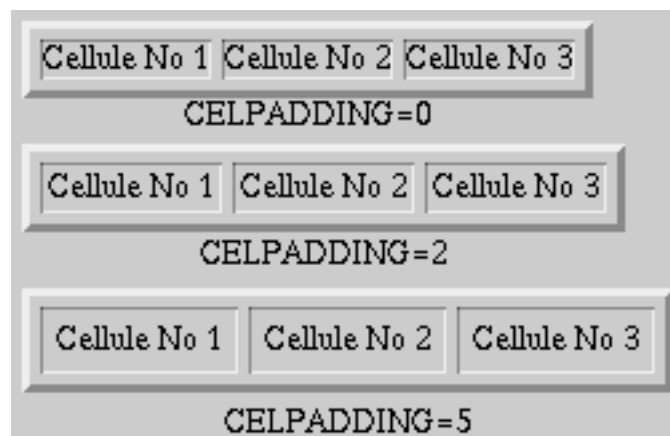
La balise <TABLE> permet l'ouverture d'un tableau ; la fin de tableau est spécifiée par </TABLE>. On peut indiquer les attributs suivants :

#### 7.13.1.1 L'attribut **BORDER=n**

Il définit l'épaisseur du cadre du tableau en pixel. Si BORDER=0, le cadre du tableau est invisible.

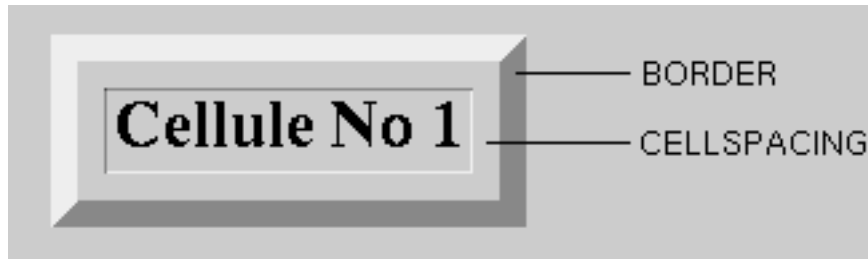
#### 7.13.1.2 L'attribut **CELLPADDING=n**

Cet attribut définit l'espace autour du texte d'une cellule en pixel.



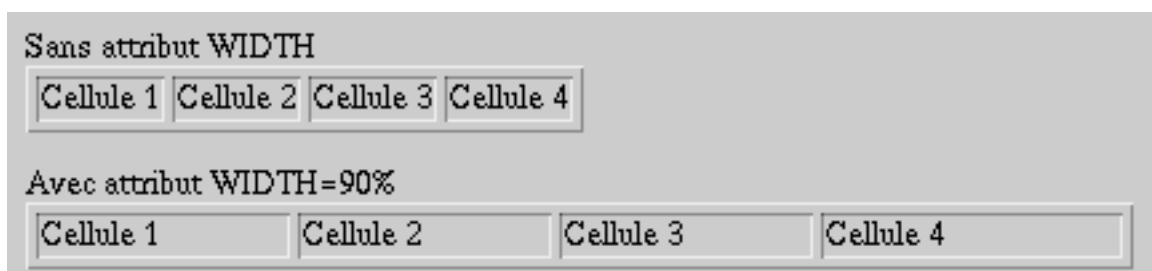
### 7.13.1.3 L'attribut **CELLSPACING=n**

Il définit l'espace inter-cellule en pixel.



### 7.13.1.4 L'attribut **WIDTH=n (pixels) ou n%**

Cet attribut permet de contraindre le tableau à occuper un certain pourcentage de la largeur de la fenêtre du navigateur ou un chiffre en pixel. Cet attribut peut se placer soit dans la balise <TABLE> soit à l'intérieur de la balise <TD>.



### 7.13.1.5 L'attribut **HEIGHT=n (pixels)**

Cet attribut détermine la hauteur d'une cellule <TD> ou du tableau entier dans la balise <TABLE>. Attention, les cellules d'une même ligne auront la même hauteur. Cet attribut n'accepte que la définition en pixel et pas en pourcentage.

## 7.13.2 <TR>

Le tableau est ensuite décrit ligne après ligne, l'élément définissant une nouvelle ligne est <TR> qui admet les attributs d'alignement du texte à l'intérieur de toutes les cellules de la ligne :

### 7.13.2.1 L'attribut VALIGN

VALIGN=x (alignement vertical) qui peut prendre les valeurs :

- . TOP place le texte en haut de la cellule,
- . BOTTOM en bas de la cellule,
- . MIDDLE au centre de la cellule.

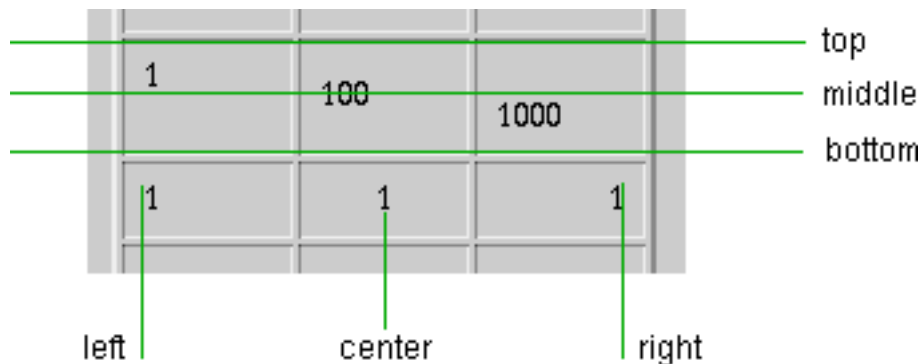
### 7.13.2.2 L'attribut ALIGN

ALIGN=x (alignement horizontal) qui peut prendre les valeurs :

- . RIGHT place le texte à droite de la cellule,
- . LEFT à gauche de la cellule,
- . CENTER centre le texte dans la cellule.

</TR> termine la définition d'une ligne mais il n'est pas obligatoire.

```
<tr align=left><td valign=top>1<td valign=middle>100<td valign=bottom>1000
```



```
<tr><td align=left>1<td align=center>1<td align=right>1
```

### 7.13.3 <TD>

<TD> est l'élément de départ d'une colonne. Il peut être complété par des attributs VALIGN et ALIGN qui seront alors prioritaires sur les mêmes valeurs définies dans la balise <TR>. Comme lui, on n'est pas obligé de le clore par </TD>.

### 7.13.3.1 Les attributs COLSPAN et ROWSPAN

Ils permettent de générer des cellules dont la surface est un multiple de la cellule élémentaire:

COLSPAN=n colonnes, pour définir une cellule dont la largeur est un multiple de la colonne de base.

ROWSPAN=n lignes, pour définir une cellule dont la hauteur est un multiple de la ligne.

The diagram shows two examples of table cells using colspan and rowspan attributes. The first example shows a 2x3 grid of cells labeled 'Cellule 12c1' through 'Cellule 13c3'. A cell at the top is labeled 'COLSPAN="3"' and spans all three columns. The second example shows a 3x3 grid of cells. A cell on the left is labeled 'ROWSPAN="3"' and spans all three rows.

La matrice du tableau définissant le nombre de cellules élémentaires est calculée par le nombre de lignes du tableau (nombre d'instructions <TR>), multiplié par le nombre de cellules (nombre d'instructions <TD>) de la ligne définissant le plus de cellules (plus grand nombre de <TD>)

Le nombre de cellules par ligne du tableau est calculé sur la ligne définissant le plus de cellules.

### 7.13.3.2 L'attribut NOWRAP

L'attribut NOWRAP de la balise <TD> force le navigateur à inscrire tout le texte de la cellule sur une seule ligne.

The diagram shows two examples of table rows. The first example shows a row with three columns. The first column contains the text 'Mise à Jour pour les Now Utilities 6.5. Il s'agit là de la mise à jour de Novembre 96 qui transforme Now Startup Manager 7.02 en 7.03 et Now ScrapBooks 6.52 en 6.53 (fourni)'. The second column contains '553 Ko' and the third column contains '02/12/96'. The second example shows a row with three columns. The first column contains the text 'Mise à Jour pour les Now Utilities 6.5' followed by a long dashed line representing truncated text. The second column contains '553 Ko' and the third column contains '02/12/96'. The first example uses the attribute <TR ALIGN=Left VALIGN=Middle> and the second example uses <TR ALIGN=Left VALIGN=Middle> and <TD NOWRAP >.

#### 7.13.4 L'attribut BGCOLOR

L'attribut BGCOLOR="#rrggb" permet de donner une couleur aux cellules selon la même méthode que dans la balise <BODY> et <FONT>. L'attribut BGCOLOR se place aussi bien à l'intérieur des balises <TABLE> que <TR> et <TD>.

#### 7.13.5 <TH>

Enfin la balise <TH> qui est une balise fonctionnant de façon similaire à la balise <TD> et admettant les mêmes attributs mais considérée comme balise de titre d'une cellule. Le centrage du texte et l'utilisation de caractères gras sont automatiques.

#### 7.13.6 <CAPTION>

Un tableau peut recevoir un titre défini entre les balises <CAPTION> et </CAPTION>. Ce titre peut se situer soit au-dessus en utilisant l'attribut ALIGN=TOP ou en dessous en utilisant ALIGN=BOTTOM.

#### 7.13.7 Exemple:

```
<TABLE BORDER=3 CELLSPACING=5 CELLPADDING=10 BGCOLOR="#55C9FF" WIDTH="300">  
<CAPTION ALIGN=Bottom>Titre du tableau</CAPTION>
```

```
<TR ALIGN=Center VALIGN=Middle>  
  <TD COLSPAN="2" ROWSPAN="2" ALIGN=Right VALIGN=Bottom>l1c1  
  <TD NOWRAP>l1c3  
  <TD BGCOLOR="#FF5454">l1c4  
<TR ALIGN=Center VALIGN=Middle>  
  <TD WIDTH="20">l2c3  
  <TD>l2c4  
<TR ALIGN=Center VALIGN=Middle>  
  <TD>l3c1  
  <TD>l3c2  
  <TD>l3c3  
  <TD>l3c4
```

```
</TABLE>
```

## 7.14 Les Frames ou cadres

### 7.14.1 Introduction

La notion de Frame supprime une des limitations du Web. Auparavant, chaque nouveau document venait écraser le précédent, et obliger bien souvent l'utilisateur à user et abuser de la commande back des navigateurs. Désormais, il est possible de dédier certaines zones du navigateur à des tâches bien précises. On peut imaginer, par exemple, une zone qui contienne une barre de menu et qui n'est quasiment jamais rechargée. Ou encore une autre zone utilisée pour afficher une aide contextuelle, sans que l'utilisateur perde la page sur laquelle il travaille.

### 7.14.2 Création de Frames

Trois nouvelles balises ont été ajoutées au langage HTML afin d'apporter cette nouvelle fonctionnalité. Cependant, la structure générale d'un document divisé en plusieurs Frames diffère de celle d'un document classique. Dans un document classique, le corps est inséré dans les balises `<BODY> ... </BODY>`. Dans le cas des Frames, le corps du document est systématiquement inséré dans les balises `<FRAMESET>... </FRAMESET>` et de plus il ne peut être composé qu'à partir des trois nouvelles balises `<FRAMESET>`, `<FRAME>` et `<NOFRAMES>`.

### 7.14.3 <FRAMESET>

<FRAMESET> est la balise d'ouverture permettant de diviser une zone en sous-zone, soit verticalement, soit horizontalement. Si aucune zone n'est encore définie, les divisions s'appliquent à la zone initiale formée par l'ensemble de la fenêtre du navigateur.

#### 7.14.3.1 L'attribut ROWS

ROWS="hauteur1, hauteur2,..., hauteurN" définit la hauteur des différentes Frames en cas d'un découpage horizontal.

Chacune des valeurs de la liste peut être donnée selon l'un des trois formats suivants, où n est un entier :

- n : la valeur de n indique la hauteur de la sous-zones en nombre de pixels ;
- n % : n indique ici la hauteur de la sous-zone exprimée en pourcentage de la taille de la zone mère ; .
- n\* : ici, n est optionnel. Le caractère \* indique au navigateur qu'il doit donner à la sous-zone toute la place encore disponible. Lorsque plusieurs éléments de la liste utilisent ce format, la place encore disponible est répartie entre les sous-zones correspondantes. n pourra alors être utilisée pour indiquer au navigateur qu'une sous-zone doit recevoir une proportion n fois supérieure de la place encore disponible.

#### 7.14.3.2 L'attribut COLS

COLS est utilisé pour diviser la zone en sous-zones verticales. La syntaxe est en tous points identiques à celle de l'attribut ROWS :

COLS="largeur\_zone1, largeur\_zone\_2,...,largeur\_zone\_n"



#### 7.14.3.3 L'attribut **BORDER** (nouveau de Netscape 3)

**BORDER=n** permet de jouer sur la grosseur des bordures. Si la valeur est **BORDER=0**, la bordure ne s'affiche pas. Cet attribut ne se place que dans la balise **<FRAMESET>**.

#### 7.14.3.4 L'attribut **BORDERCOLOR** (nouveau de Netscape 3)

**BORDERCOLOR="#rrggb"** définit la couleur de la bordure selon le modèle RGB déjà connu dans la partie sur la balise **<BODY>**.

Inséré entre les balises **<FRAMESET>...</FRAMESET>**, on ne peut trouver que les trois balises suivantes :

**<FRAMESET>...</FRAMESET>**, qui permet de rediviser une sous-zone ;

**<FRAME>** qui est utilisée pour caractériser une sous-zone ;

et enfin **<NOFRAME>...</NOFRAME>** qui permet d'afficher un texte à la place des Frames si le navigateur n'est pas capable de gérer les Frames.

**Aucune** autre balise HTML ne peut être acceptée.

### 7.14.4 **<FRAME>**

**<FRAME>** est la balise utilisée pour caractériser les sous-zones définies à l'aide de la balise **<FRAMESET>**. Cette caractérisation est réalisée à l'aide de six attributs.

#### 7.14.4.1 L'attribut **SRC**

**SRC="URL"** indique le document à afficher dans la Frame.

#### **7.14.4.2 L'attribut NAME**

NAME est utilisé pour nommer la zone afin qu'elle puisse devenir la cible de n'importe quel lien hypertexte. Les noms prédéfinis, `_blank`, `_top`, `_parent`, `_self` ainsi que tous les noms commençant par le caractère `'_'` ne peuvent pas être utilisés pour renseigner cet attribut.

La syntaxe de l'attribut NAME est : `NAME="nom_de_la_zone"`

#### **7.14.4.3 L'attribut MARGINWIDTH**

`MARGINWIDTH="n"` définit l'espace entre le document et les frontières verticales de la Frame ;

#### **7.14.4.4 L'attribut MARGINHEIGHT**

`MARGINHEIGHT="n"` définit l'espace entre le document et les frontières horizontales de la Frame ;

#### **7.14.4.5 L'attribut SCROLLING**

`SCROLLING="Yes / no / auto"` indique si la Frame doit ou non posséder une barre de défilement ;

#### **7.14.4.6 L'attribut NORESIZE**

`NORESIZ` indique qu'il est impossible à l'utilisateur de redimensionner la Frame.

#### **7.14.4.7 L'attribut FRAMEBORDER** (nouveau de Netscape 3)

`FRAMEBORDER=YES` ou `NO`. La valeur par défaut est `Yes`. Permet de définir s'il faut afficher les bordures ou pas. Peut-être placé dans les balises `<FRAMESET>` et `<FRAME>`.

#### 7.14.4.8 L'attribut **BORDERCOLOR** (nouveau de Netscape 3)

**BORDERCOLOR**="#rrggbb" définit la couleur de la bordure selon le modèle RGB déjà connu dans la partie sur la balise <BODY>.

#### 7.14.5 <NOFRAMES>...</NOFRAMES>

Cette balise est utilisée pour indiquer le texte que doivent afficher les navigateurs incapables de gérer les Frames. Ce texte est bien sûr ignoré par un navigateur capable de gérer les Frames.

#### 7.14.6 **Exemple:** (imbrication de balises <FRAMESET>)

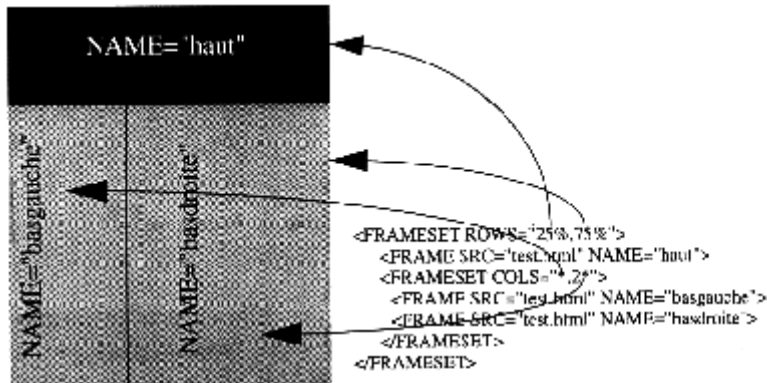
```
<HTML>
<FRAMESET Rows="50,*" BORDERCOLOR="#BED5FF" BORDER=3>
  <FRAME Name="logo" SRC="logo.html" Scrolling="No" NORESIZE
MARGINHEIGHT="5" MARGINHEIGHT="10" FRAMEBORDER=NO>

  <FRAMESET Cols="25%,75%">
    <FRAME Name="sommaire" SRC="sommaire.html" Scrolling="Yes">
    <FRAME Name="intro" SRC="introduction.html" Scrolling="Yes">
  </FRAMESET>

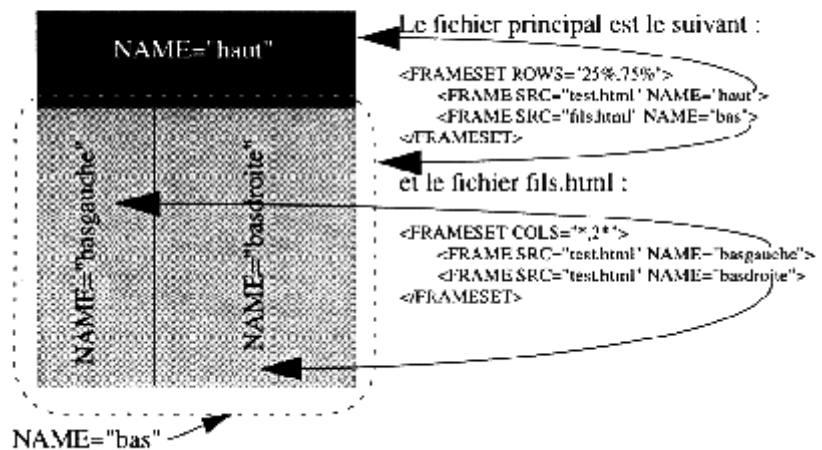
</NOFRAMES>
<BODY>
Mettez &agrave; jour votre logiciel de navigation.
</BODY>
</NOFRAMES>
</FRAMESET>
</HTML>
```

Voici désormais deux exemples qui illustrent la facilité d'utilisation des Frames. Dans les exemples qui suivent, on suppose que le fichier test.html utilisé pour renseigner l'attribut SRC des balises <FRAME> est un fichier HTML classique.

Dans le premier exemple, nous avons imbriqué plusieurs niveaux de Frames. En effet, une première balise <FRAMESET> divise la fenêtre du navigateur en zones horizontales, puis une seconde balise <FRAMESET>, insérée dans la première, divise à nouveau une des Frames obtenues en zones verticales.



Imbrication de balises



Imbrication de documents

Dans le second exemple, on utilise l'imbrication de documents. C'est-à-dire qu'au lieu d'imbriquer une deuxième balise <FRAMESET> à l'intérieur de la première, on crée un second fichier à part contenant cette seconde <FRAMESET>. Cette façon de faire peut sembler moins pratique mais elle permet de nommer la FRAME qui contient les Frames « basgauche » et « basdroite ». Cela est impossible lorsqu'on imbrique des balises <FRAMESET>, dans la mesure où cette balise ne possède pas d'attribut NAME. Dans la pratique, on aura souvent avantage à imbriquer les documents plutôt que les balises <FRAMESET> en prévision de la manière dont on va naviguer dans le site Web.

### 7.14.7 Utilisation des Frames

Nous allons voir comment créer des liens hypertextes utilisant ces Frames. Un nouvel attribut a été créé. Il concerne les balises caractérisant des liens hypertextes, en particulier les balises <A> et <FORM>.

#### 7.14.7.1 L'attribut TARGET

Cet attribut permet de préciser le nom de la zone qui doit recevoir le document correspondant au lien.

Ceci donnera par exemple :

```
<A HREF="URL" TARGET="nom_de_Frame">...</A>  
<FORM ACTION="URL" TARGET="nom_de_Frame">...</FORM>
```

On peut isoler trois cas pour décrire les valeurs possibles de l'attribut TARGET.

Dans le premier cas, l'attribut TARGET a une valeur qui correspond au nom d'une Frame existante. Le document associé au lien hypertexte est alors affiché dans la Frame en question.

Dans le second cas, l'attribut TARGET a une valeur qui ne correspond à aucune des Frames existantes. Le navigateur crée alors une nouvelle fenêtre (un nouveau navigateur) et y affiche le document. Cette nouvelle fenêtre est considérée comme une Frame portant le nom de l'attribut TARGET à l'origine de sa création.

Enfin, dans le dernier cas, l'attribut TARGET prend une des quatre valeurs prédéfinies suivantes :

- `_blank`, qui est utilisé pour indiquer au navigateur qu'il doit créer une nouvelle fenêtre (c'est-à-dire un nouveau navigateur) afin d'y afficher le document. Le nouveau navigateur est une Frame qui ne porte pas de nom et ne peut donc pas être la cible d'un autre lien hypertexte ;
- `_self`, qui indique que le document sera chargé dans la même zone que celle dans laquelle se trouve le lien hypertexte. Il s'agit de la valeur par défaut lorsque l'attribut

TARGET n'est pas renseigné ;

- `_top`, qui indique au navigateur de supprimer toutes les Frames existantes et d'afficher le document de façon classique en occupant toute la surface du navigateur ;
- `_parent`, qui indique au navigateur d'afficher le document en occupant toute la surface de la zone dans laquelle a été affichée le document contenant le lien hypertexte et effaçant du même coup son fichier de description de Frame. Cela n'a de sens que dans le cas de l'imbrication de document et non l'imbrication de balise `<FRAMESET>`, sinon la page s'affiche dans la fenêtre entière du navigateur (comme le `_top`). Dans l'exemple d'imbrication de documents vu plus haut, le résultat sera que le fichier « `fil.html` » n'affichera plus une sous-zone de `FRAMESET` mais sera remplacé entièrement par le document appelé avec l'attribut « `_parent` » tout en laissant la `FRAME Name="haut"`.

## 8. Les extensions aux fonctionnalités du Web (ou Plug-in)

Netscape a mis au point pour la version 2 de son Navigateur un procédé qui permet d'augmenter les capacités d'affichages du Web. En effet, une page web est principalement composée de texte et d'image fixe. Netscape a donc laissé à d'autres sociétés le soin de créer des fonctionnalités supplémentaires. Grâce à cette technologie, le Web est devenu réellement multimédia permettant de lire des fichiers sons, animations, vidéo et autres de manière simple.

Concrètement, il faut télécharger un petit logiciel spécialisé dans la lecture du fichier particulier. Il faut le placer dans le dossier "Plug-in" du navigateur. Si la page Web demande à lire un fichier particulier et que vous n'avez pas le Plug-in, un icône de Plug-in brisé signale qu'il n'est pas possible de le lire car l'extension adéquate n'est pas installée. Les Plug-ins les plus courants sont: RealAudio, Adobe Acrobat, ShockWave pour Director et QuickTime. Nous allons étudier deux types d'extensions (ou Plug-ins) intégrés directement à Netscape, à savoir le son et la vidéo QuickTime.

### 8.1 <EMBED>

Cette balise indique à Netscape qu'il faut jouer un fichier qui sera lu par cette extension. Il doit être complété par des attributs qui dépendent du Plug-in. Il faudra consulter la documentation des concepteurs du Plug-in avant de l'intégrer.

## 8.2 <NOEMBED> </NOEMBED>

Cette balise est faite pour que les navigateurs qui ne reconnaissent pas la technologie des Plug-ins (très rares aujourd'hui) affichent un texte ou autre chose comme alternative.

## 8.3 Exemples de Plug-in

### 8.3.1 LiveAudio

Ce Plug-in permet à Netscape de jouer un fichier son au format AIFF, WAV AU ou MIDI directement dans la page Web.

A l'intérieur de la balise <EMBED> vous pouvez ajouter les attributs suivants:

**SRC=" \_URL\_ "** est obligatoire pour donner l'URL du fichier

**AUTOSTART=TRUE ou FALSE** qui permet au Plug-in de démarrer automatiquement lorsque le fichier a été téléchargé.

**LOOP=TRUE ou FALSE ou un chiffre**, permet de faire jouer un fichier son en boucle. Si vous mettez un chiffre, il sera joué autant de fois.

**VOLUME=0 à 100** joue le son à une certaine puissance.

**ALIGN=TOP, MIDDLE, BOTTOM, LEFT, CENTER ou RIGHT** aligne la fenêtre du Plug-in de la même manière que la balise <IMG> pour les images.

**CONTROLS=CONSOLE, SMALLCONSOLE, PLAYBUTTON, PAUSEBUTTON, STOPBUTTON, VOLUMELEVER** détermine les commandes à afficher. On peut en afficher plusieurs en les séparant par une virgule et en les encadrant par des guillemets.



**HIDDEN=TRUE** cache la fenêtre de contrôle.

etc.

Pour plus d'efficacité et des temps de téléchargement plus court, il faut enregistrer les fichiers sons d'une certaine manière quel que soit le format utilisé. Un échantillonnage compris entre 11 Khz pour la musique et 7 Khz pour la voix sont corrects. Un fichier 8 bits prend beaucoup moins de place qu'un son en 16 bits. De plus, il est recommandé de l'enregistrer avec une méthode de compression adaptée afin qu'il prenne encore moins de Kilo-octets.

### 8.3.2 QuickTime

Ce Plug-in permet de faire jouer un fichier vidéo à l'intérieur d'une page QuickTime.

Ces fichiers vidéo prennent énormément de place et sont donc très long à télécharger. Pour l'intégration des fichiers sons au format QuickTime, il faudra le compresser en 16bits avec la méthode IMA 1:4 pour permettre une compatibilité totale entre Mac et PC.

Les attributs à intégrer dans la balise <EMBED> sont les suivants:

**SRC="\_URL\_"** indique l'URL fichier vidéo. Cet attribut est obligatoire.

**PLUGINSPAGE="\_URL\_"** est un paramètre optionnel qui permet d'indiquer au navigateur qui ne dispose pas d'un Plug-in QuickTime de donner l'URL où aller le chercher. L'URL à intégrer de préférence est = <http://quicktime.apple.com/>

**WIDTH=taille en pixel** , donne la largeur de la fenêtre vidéo. Cet attribut est obligatoire.

**HEIGHT=taille en pixel**, spécifie la hauteur en pixel de la vidéo. Il faut impérativement ajouter 24 pixels à ce total pour inclure la taille des commandes de contrôle. Cet attribut est obligatoire.

**AUTOPLAY=TRUE ou FALSE**, permet de faire jouer automatiquement le fichier vidéo dès que son téléchargement est terminé ou dès qu'il peut le jouer.

**CONTROLLER=TRUE ou FALSE**, affiche ou efface les commandes de contrôle.

**LOOP=TRUE, FALSE ou PALINDROME**, permet de faire jouer une vidéo en boucle. L'attribut PALINDROME joue le fichier vidéo en avant et en arrière.

**HREF="\_URL\_"**, permet de transformer la vidéo en lien hypermédia. Il peut être associé à l'attribut TARGET dans le cas où la vidéo s'affiche dans une FRAME.

## 9. JavaScript

Le JavaScript est un langage de programmation mis au point par Netscape pour la version 2 de son navigateur. MS Internet Explorer 3 l'a aussi intégré, si bien que près de 70% des navigateurs Web actuels le reconnaissent. Le JavaScript n'a rien à voir avec le langage Java développé par Sun pour créer des logiciels multi-plateformes et exécutables par Internet. Ici, c'est un langage de script qui permet de donner des pages plus dynamiques qu'avec le HTML. Le code (ou source) est directement écrit dans la page HTML. Il est donc facile de les retrouver pour les modifier ou s'en inspirer. Malheureusement, son exploitation est difficile car elle requiert des notions de programmation en langage C afin de pouvoir l'exploiter correctement. Nous ne traiterons donc pas la programmation en JavaScript mais nous prendrons plusieurs exemples de scripts qui peuvent être intégrés à la page HTML.

Ce script se placera le plus souvent entre les balises <HEAD> en utilisant la balise <SCRIPT> comme dans cet exemple:

```
<HEAD>
  <SCRIPT Language="JavaScript">
    .....
  </SCRIPT>
</HEAD>
```

Une commande (ou fonction) sera placée dans la page HTML pour exécuter le script:

```
<SCRIPT Language="JavaScript">
  fonction.a.executer ()
</SCRIPT>
```

Pour avoir plus d'information sur le JavaScript avec des exemples à adapter, vous pouvez consulter ces pages Web:

**Netscape:**

[http://home.netscape.com/comprod/products/navigator/versions\\_3.0/building\\_blocks/jscript/index.HTML](http://home.netscape.com/comprod/products/navigator/versions_3.0/building_blocks/jscript/index.HTML)

**The JavaScript index:**

<http://www.c2.org/~andreww/javascript/>

**JavaScript Tip of the week:**

<http://www.webreference.com/javascript/>

## 9.1 Exemple 1: Menu déroulant.

Ce menu permet de sélectionner un élément et d'aller directement vers la page HTML correspondante. Dans l'exemple ci-dessous, le menu est placé dans une FRAMESET qui ne bougera pas pour garder constamment ce menu en place et seule la page à charger dans une deuxième FRAMESET bougera. S'il faut faire charger la page qui se trouve dans la même FRAMESET mais dans une autre FRAME il faut remplacer:

```
else if (v == "4") {  
    parent.Frames[2].location = "http://www.insset.u-picardie.fr/WebDavido/";  
}
```

**Par :**

```
else if (v == "4") {  
    parent.Frames[1].location = "http://www.insset.u-picardie.fr/WebDavido/";  
}
```

Noter le chiffre 1 à mettre après « parent.Frames[1] »

Pour que ce menu soit remplacé dans la même page il suffit de mettre ceci:

```
else if (v == "4") {  
    location = "http://www.insset.u-picardie.fr/WebDavido/";  
}
```

Remarquez la disparition de parent.Frames[2].

L'exemple ci-dessous montre comment le placer dans la page HTML, il est suivi de son équivalent en script CGI.

```
<HTML><HEAD>

<TITLE>Barre de Navigation</TITLE>
</HEAD>

<SCRIPT LANGUAGE="LiveScript">
function change(o)
{
  v = o.options[o.selectedIndex].value;
  if (v == "0") {
    alert("Veuillez selectionner une page...");
  } else if (v == "1") {
    parent.Frames[2].location = "http://www.insset.u-picardie.fr/index_Frames.shtml";
  } else if (v == "2") {
    parent.Frames[2].location = "http://www.insset.u-picardie.fr/Pages_INSSET/index.HTML";
  } else if (v == "3") {
    parent.Frames[2].location = "http://www.insset.u-picardie.fr/Pages_INSSET/Recherche.HTML";
  } else if (v == "4") {
    parent.Frames[2].location = "http://www.insset.u-picardie.fr/WebDavido/";
  } else if (v == "5") {
    parent.Frames[2].location = "http://www.insset.u-picardie.fr/entreprises/";
  } else if (v == "6") {
    parent.Frames[2].location = "http://www.insset.u-picardie.fr/saint-quentin/";
  } else if (v == "7") {
    parent.Frames[2].location = "http://www.insset.u-picardie.fr/saint-quentin/musee_lecuyer/";
  } else if (v == "8") {
    parent.Frames[2].location = "http://www.insset.u-picardie.fr/geopolymer/";
  }
}
}
</SCRIPT>

<BODY BGCOLOR="#FFFFFF">

<FORM>

<SELECT onChange="change(this)">
<OPTION VALUE="0">Sélectionnez un site...
<OPTION VALUE="1">Page d'accueil
<OPTION VALUE="2">l'IN.S.S.E.T.
<OPTION VALUE="3">Centres de recherche de l'IN.S.S.E.T.
<OPTION VALUE="4">WebDavido (aide à la navigation sur Internet)
<OPTION VALUE="5">Entreprises
<OPTION VALUE="6">Ville de Saint-Quentin
<OPTION VALUE="7">Musée Maurice-Quentin de La Tour
<OPTION VALUE="8">Geopolymer Institute
</SELECT>
<A HREF="/10Km/" TARGET="page_web"><IMG SRC="images/nouveaute.gif" WIDTH=59 HEIGHT=12 BORDER=0
HSPACE=10 ALT="nouveauté;"></A>
</FORM>

<P><BR><BR><BR><BR>
Si votre navigateur OuaiB n'est pas compatible avec le JavaScript, voici une autre version.<P>
<FORM ACTION="http://www.insset.u-picardie.fr/cgi-bin/redirect" METHOD="POST" TARGET="web_page">
<SELECT NAME="url">
```

```

<OPTION VALUE="">Sélectionnez un site...
<OPTION VALUE="http://www.insset.u-picardie.fr/index.HTML">Page d'accueil
<OPTION VALUE="http://www.insset.u-picardie.fr/Pages_INSSET/index.HTML">l'IN.S.S.E.T.
<OPTION VALUE="http://www.insset.u-picardie.fr/Pages_INSSET/Recherche.HTML">Centres de recherche
de l'IN.S.S.E.T.
<OPTION VALUE="http://www.insset.u-picardie.fr/WebDavido/">WebDavido (aide &grave; la navigation
sur Internet)
<OPTION VALUE="http://www.insset.u-picardie.fr/entreprises/">Entreprises
<OPTION VALUE="http://www.insset.u-picardie.fr/saint-quentin/sommaire.HTML">Ville de Saint-
Quentin
<OPTION VALUE="http://www.insset.u-picardie.fr/saint-quentin/musee_lecuyer/">Musée
Maurice-Quentin de La Tour
<OPTION VALUE="http://www.insset.u-picardie.fr/geopolymer/">Geopolymer Institute
</SELECT>
<INPUT TYPE=Submit VALUE="Lancement !">
</FORM>

</BODY>
</HTML>

```

## 9.2 Exemple 2: Un message déroulant dans la barre de statut.

Les lettres accentuées du message devront être codées en ISO 8859-1 (le é doit s'écrire È) pour Netscape 2 ou en ISO Latin 1 à partir de Netscape 3 (le é s'écrit &eacute;). Il ne faut pas ajouter de guillemets. Pour les apostrophes, il faut les faire précéder d'un anti-slash \ (sur Mac: Option-Maj-/)

```

<HTML>
<HEAD>
<TITLE>JavaScript Scrolling Text</TITLE>

<SCRIPT LANGUAGE="JavaScript">
<!-- on le cache
// Chris Skinner, January 30th, 1996.

function scrollit(seed)
{
    var m1 = "Ceci es un exemple de texte déroulant avec JavaScript";
    var m2 = " . . . vous pouvez mettre votre message ici pour attirer l\'attention";
    var m3 = " . . . Cela peut-être très pratique ! ";

```

```

var msg=m1+m2+m3;
var out = " ";
var c = 1;

if (seed > 100) {
    seed--;
    var cmd="scrollit(" + seed + ")";
    timerTwo=window.setTimeout(cmd,100);
}
else if (seed <= 100 && seed > 0) {
    for (c=0 ; c < seed ; c++) {
        out+=" ";
    }
    out+=msg;
    seed--;
    var cmd="scrollit(" + seed + ")";
    window.status=out;
    timerTwo=window.setTimeout(cmd,100);
}
else if (seed <= 0) {
    if (-seed < msg.length) {
        out+=msg.substring(-seed,msg.length);
        seed--;
        var cmd="scrollit(" + seed + ")";
        window.status=out;
        timerTwo=window.setTimeout(cmd,100);
    }
    else {
        window.status=" ";
        timerTwo=window.setTimeout("scrollit(100)",75);
    }
}
}
// fin de la cachette -->
</SCRIPT>

</HEAD>

<BODY onLoad="timerONE=window.setTimeout('scrollit(100)',500);">
</BODY></HTML>

```

### 9.3 Exemple 3: Afficher un texte au lieu de l'URL dans la barre de statut.

Quand votre souris passe sur un lien hypermédia, vous remarquez dans la barre de statut en bas de la fenêtre l'URL correspondant à ce lien. Avec le JavaScript il est

possible d'afficher un texte à la place pour donner plus de renseignements. Ce script ne doit pas être couplé avec celui qui fait défiler du texte car on y verrait plus rien !

Un lien hypertexte normal s'écrit comme cela:

```
<A HREF="index.HTML"> Page d'accueil </A>
```

Modifié avec JavaScript

```
<A HREF="index.HTML" onMouseOver = "self.status = ' Vous retournez &agrave; la  
page d\'accueil pour voir le sommaire du site ';return true"> Page d'accueil  
</A>
```

Ainsi, lorsque la souris passe sur le lien, dans la barre de statut s'affiche le texte: « Vous retournez à la page d'accueil pour voir le sommaire du site » au lieu de l'URL de cette page.

## 9.4 Exemple 4: Un compteur d'accès personnel

Ce compteur permet de dire combien de fois une même personne a visité cette page. C'est un compteur plus personnalisé que les compteurs qui disent combien de personnes au total ont visité une page.

Ce Script utilise le Cookie du Navigateur Web. Un Cookie est un petit fichier créé par Netscape ou MIE qui garde sur l'ordinateur du consultant des informations précises.

Pour voir ce script fonctionner, une démonstration existe sur le WebDavido (<http://www.insset.u-picardie.fr/WebDavido/>)



En JavaScript le texte qui suit deux // est un commentaire.

Les guillemets ", apostrophes ' et parenthèses () doivent être précédés d'un anti-slash:

\ " \' \ ( \ )

Le cookie de ce script s'appelle WebDavido-compteur . Pour éviter toute interférence, je vous conseille de modifier ce nom.

Un commentaire précédé de deux // dans ce script vous dit où vous devez modifier cela.

Ce script est à placer entre les deux balises <HEAD> et </HEAD>

```
<SCRIPT language="JavaScript">
<!-- Cachons le Script des navigateurs qui ne savent pas le lire

// ici est le compteur
function show_count() {
    var expdate = new Date();
    var num;
    expdate.setTime(expdate.getTime() + (24 * 60 * 60 * 1000 * 700)); // remise a zero du
compteur dans 700 jours
    if(!(num = GetCookie("WebDavido-compteur"))) // changez le nom du cookie ici
    num = 0;
    num++;
    SetCookie ("WebDavido-compteur", num, expdate); // changez le nom du cookie ici
    if (num == 1) {return ("Est-ce la premi&egrave;re fois que vous venez ici ?");}
    else {return ("Vous &ecirc;tes d&eacute;j&agrave; venu(e) " + num + " fois.");}
}

// Fin de la cachette -->
</SCRIPT>
```

Pour afficher le compteur dans la page HTML, à l'endroit où vous voulez l'insérer faites ceci:

```
<SCRIPT language="JavaScript">
<!-- Cachons le Script

    show_count()

// fin de la cachette -->
</SCRIPT>
```

## 9.5 Exemple 5: Afficher un élément aléatoirement dans une page HTML

Grâce au JavaScript, vous pouvez insérer aléatoirement du texte, image, lien, etc. dans une page HTML. Cet élément sera pris au hasard dans une liste prédéfinie. Vous pouvez mettre autant d'éléments que vous le désirez. En JavaScript le texte qui suit deux // est un commentaire. Les guillemets ", apostrophes ' et parenthèses () doivent être précédés d'un anti-slash: \" \' \(\ \)

### **Important:**

La fonction qui commence par « var Idea = new MakeArray(" ... ") » ne doit jamais avoir de retour chariot. Sinon, un bogue survient et le script ne fonctionnera pas.

Ce script est à placer entre les deux balises <HEAD> et </HEAD>

```
<SCRIPT language="JavaScript">
<!-- Cachons le Script des navigateurs qui ne savent pas le lire

// ici se trouve l'affichage aleatoire
// Vous pouvez mettre autant d'elements que vous voulez. Il faudra les separer par ","
// Les guillemets ", apostrophes ' et parentheses () doivent etre precedes
// d'un anti-slash: \" \' \(\ \)

var Idea = new MakeArray(" élément ", " Texte ", " Image ", " Lien hypermédia")

function MakeArray(){
    this.length = MakeArray.arguments.length
    for (var i = 0; i < this.length; i++)
        this[i+1] = MakeArray.arguments[i]
    }

function randomize(the_number)
    {
    now = new Date();
    rnd = now.getSeconds() * now.getMinutes(); // secondes x minutes
    tmp = Math.cos(rnd);
        if (tmp < 0) tmp = 0 tmp; // un petit cosinus
    rnd = tmp * the_number;
    rnd = Math.round(rnd); // on arrondie le chiffre
    if (rnd == 0) rnd = 1;
    }

// Fin de la cachette -->
</SCRIPT>
```

Pour afficher l'élément pris aléatoirement dans la page HTML, à l'endroit où vous voulez l'insérer faites ceci:

```
<SCRIPT LANGUAGE = "JavaScript">
<!-- cachons le script
// affichage aleatoire

    randomize(Idea.length)
    txt = Idea[rndl;
    window.document.write(txt)
// fin de la cachette -->
</SCRIPT>
```

## 9.6 Exemple 6: Affichage des icones dynamiques.

Vous savez qu'en HTML une image peut-être un lien hypermédia. Comme pour les logiciels, il serait plus clair d'animer ces icones en les grisants ou en faisant en sorte que lorsqu'on appuie dessus, il s'enfonce comme un bouton en 3D. Grâce au JavaScript, on peut obtenir quelque chose d'équivalent. Dans notre exemple, deux boutons sont des raccourcis vers des pages. L'un s'appelle Programme, l'autre Internet. Lorsque la souris passera dessus, l'icone va changer d'aspect pour spécifier que cet icone est un lien hypermédia. Ceci n'est uniquement compatible qu'avec Netscape 3 car une commande JavaScript a été ajoutée à cette version.

Dans cet exemple, lorsque la souris passera sur l'icone, celui-ci va s'éclaircir. Lorsque la souris quittera cet icone, ce dernier reprendra son aspect premier normal. Pour cela il faut créer deux images. Une image normale et une autre plus claire ayant surtout la même hauteur et largeur.

L'icône prog\_nor.gif sera le nom de l'icône normal du raccourci « Programme » et l'icône prog\_clair.gif sera le nom de cette même icône en plus clair. Ce jeu d'icônes doit être identifié. Nous l'appellerons « image1 ». Il faudra le préciser à l'intérieur du lien et dans la balise IMG comme le montre l'exemple. La même chose est faite avec le raccourci « Internet ». Attention à ne pas mélanger et à bien respecter les apostrophes ( ' ) et guillemets ( " ). L'image présentée dans la balise <IMG> est la première image qui sera vue même par les logiciels qui ne reconnaissent pas le JavaScript. C'est donc l'image normale.

```
<HTML><HEAD>
<TITLE>Icône dynamique</TITLE>
</HEAD>
<BODY>

<CENTER>
<BR>
<!-- le lien doit tenir sur une seule ligne sans retour chariot. Modifier les liens et noms de
l'image en fonction de votre page -->

<A HREF="programme.HTML" onMouseOver= 'image1.src="prog_clair.gif" '
onMouseOut= 'image1.src="prog_nor.gif" '><IMG NAME="image1" SRC="prog_nor.gif" WIDTH=150
HEIGHT=40 BORDER=0 ALT="PROGRAMME"></A>

<A HREF="internet.HTML" onMouseOver= 'image2.src="internet_clair.gif" '
onMouseOut= 'image2.src="internet_nor.gif" '><IMG NAME="image2" SRC="internet_nor.gif" WIDTH=150
HEIGHT=40 BORDER=0 ALT="INTERNET"></A>

</CENTER>
</BODY></HTML>
```

## 10. Conclusion:

Comme vous l'avez vu, les principes de la programmation HTML sont relativement simples. Les deux problèmes qui vont surgir seront d'assimiler correctement toutes les facettes et subtilités de ce langage (grâce à la pratique et l'expérience) et de suivre les constantes évolutions du HTML afin de ne pas être perdu d'ici un an ou deux.

Pour cela, je vous conseille de suivre les travaux du Consortium WWW ( <http://www.w3.org/> ), les pages de Netscape ( <http://www.netscape.com/fr/> ) et Microsoft Internet Explorer ( <http://www.microsoft.com/france/> ) afin de lire les fonctionnalités supplémentaires qu'apportent les nouvelles versions de leurs logiciels de navigation.

Une fois vos pages HTML terminées, vous devrez suivre les étapes suivantes:

- 1) Vérifier la compatibilité d'affichage de vos pages avec les anciennes versions des navigateurs et les différents modèles de logiciels (Netscape, MS Internet Explorer, Mosaic, etc.). Il faudra surtout vérifier que le design de vos pages ne sera pas bouleversé par ces incompatibilités. Dans ce cours, je me suis efforcé de vous expliquer les possibilités qui correspondent parfaitement au standard actuel à savoir Netscape 2 et MS Internet Explorer 3 qui représentent à eux deux près de 70% des navigateurs

Web utilisés à la fin de l'année 1996.

2) Assurer la promotion de votre site en le référencant dans les annuaires d'Internet. Il existe des serveurs qui rassemblent divers formulaires en une fois. Je vous conseille ces deux-là:

**Francité:** <http://www.i3d.qc.ca/i3dpro/francite.exe/ajout>

**Submit-It:** <http://www.submit-it.com/>

Et ne pas oublier les serveurs plus professionnels ainsi que les serveurs français (une liste complète est disponible sur le WebDavido: <http://www.insset.u-picardie.fr/WebDavido/>) dans lesquels votre site pourrait être enregistré.

Mais il faudra surtout s'assurer que ces sites ont bien référencé votre site. Cela peut être long et ne pas se faire du premier coup.

3) Préparer de constantes mises à jours et améliorations afin de donner une image dynamique de votre site.

Étant donné que l'Internet et le Web sont des mondes en constante évolution et en croissance exponentielle, je vous conseille de lire la presse spécialisée. Vous pourrez suivre l'actualité des technologies et des services proposés afin d'être toujours à jour dans vos connaissances. Vous pouvez d'abord commencer par les journaux en ligne qui parlent de cela en français:

**Branchez-vous:** <http://www.branchez-vous.com/>

**Chroniques de Cybérie:** <http://www.cyberie.qc.ca/chronik/>

**Cahier multimédia de Libération:** <http://www.liberation.fr/multi/index.HTML>

**Supplément multimédia du journal Le Monde:** <http://www.lemonde.fr/>

**Chronique Cyber de France2:** <http://www.france2.fr/>

**CyberFlash de Canal + :** <http://www.cplus.fr/>

Et les magazines en vente dans les maisons de la presse:

**Planète Internet:** <http://www.netpress.fr/>

**Netsurf** pour son découpage entre l'Internet personnel et professionnel

Enfin, en cas de réelle difficulté, vous pouvez m'écrire un message en décrivant précisément votre problème:

[ralph.davidovits@insset.u-picardie.fr](mailto:ralph.davidovits@insset.u-picardie.fr)