

# **LE LANGAGE HTML**

## LE H.T.M.L. Version4

Jusqu'en 1990, les informations disponibles sur Internet exigeaient une bonne connaissance technique pour être lues. C'était d'ailleurs tellement compliqué que même des physiciens rencontraient de grandes difficultés pour échanger de simples données. L'un d'entre eux, Tim Berners-Lee a créé une méthode pour relier avec facilité des documents sur Internet à l'aide de liens hypertextes. L'idée n'était pas nouvelle mais son langage HTML simple a réussi là où des projets hypertextes ambitieux avaient échoué.

**Le terme HTML signifie HyperText Markup Language. Il s'agit d'un langage de création de scripts (contrairement aux langages de programmation qui remplissent des fonctions beaucoup plus étendues). Le langage HTML permet de coder une page à l'aide de commandes de mise en forme. Ces dernières sont interprétées par un navigateur Web et apparaissent ensuite sur votre écran.**

Les pages HTML sont aujourd'hui le système de base d'Internet. Elles offrent une grande variété d'applications allant de la page personnelle au service commercial professionnel. Les sites Web peuvent inclure du texte ainsi que des graphiques animés, du son, de la vidéo ou des programmes interactifs complets. Des millions de pages Web sont disponibles chaque jour sur des centaines de serveurs de par le monde.

Le code HTML ne se rencontre pas exclusivement sur le Web. La plupart des réseaux d'entreprises privés l'utilisent désormais pour fournir des informations professionnelles aux employés et aux clients. Ce langage est également celui des présentations sur CD-ROM et du nouveau format haute capacité DVD (Digital Versatile Disk). Dans la nouvelle version d'Internet Explorer, Microsoft a utilisé le langage HTML pour relier le navigateur avec le système d'exploitation Windows.

En tant que créateur de pages Web, n'oubliez pas les différences fondamentales entre les pages Web et les pages papier. Ce qui apparaît sur votre écran comme une simple page, peut-être constitué de divers éléments situés dans des fichiers d'ordinateurs différents. Il est possible par exemple de créer une page qui associe du texte stocké sur un ordinateur en Australie avec des images de Russie et des sons en provenance d'un ordinateur français.

Il existe trois méthodes pour coder en langage HTML.

1. Utilisation d'un simple éditeur de texte comme NOTEPAD ou WORDPAD ou EMACS, vous devrez tout faire à la main mais c'est une excellente façon de progresser.
2. Utilisation d'un éditeur HTML comme HomeSite de Allaire pour WIN95 ou Pagespinner pour MAC. C'est un bon choix pour les débutants car on peut l'utiliser comme éditeur de texte simple et *seulement* commencer à utiliser les fonctions évoluées lorsque les connaissances de base sont acquises.
3. Utilisation d'un package de logiciel WYSIWIG vous permettant de voir directement ce que vous allez obtenir comme FrontPage de Microsoft, PageMill d'Adobe, Home Page de Claris ou encore Dreamweaver de MacroMedia. Les sociétés de production importantes qui travaillent avec HTML choisissent souvent ce type de produit, c'est pourquoi il est intéressant de les connaître.

## LES CODES DE BASE DE LA PAGE HTML

Pour créer une page Web, plusieurs balises sont indispensables. Une **balise** est une commande codée qui indique comment doit être affichée une partie de votre page Web.

Les balises sont facilement identifiables. Elles sont constituées de mots encadrés entre les signes < (inférieur) et > (supérieur). La majorité des balises sont composées de deux parties : une balise d'ouverture (<XXXX>) qui signale le début d'un morceau de texte, et une balise de fin (</XXXX>) pour en signaler la fin.

Voici les paires de balises correspondantes pour la structure de base :

**<HTML>** et **</HTML>**. Vous rencontrerez ces balises en tout début et à la fin d'une page HTML. Vous ne devez pas les oublier, car elles permettent au navigateur de reconnaître la page HTML.

**<HEAD>** et **</HEAD>**. La balise HEAD va regrouper toutes les balises HTML et les informations associées qui s'appliquent à la page. Dans le cas d'une page HTML simple, cette section n'aura qu'une seule balise :

**<TITLE>** et **</TITLE>**. Cette balise spécifie un nom pour votre page et le titre apparaîtra au-dessus de la barre de menu dans l'interface du navigateur. Il sera également visible dans le fichier des signets si un visiteur choisit d'enregistrer la page dans ses sites favoris.

**<BODY>** et **</BODY>**. La balise BODY définit la zone contenant le corps de texte et contrôle divers aspects visuels du corps de la page comme l'arrière-plan, la couleur du texte, la couleur des liens, et même les marges.

EXEMPLE :

```
<HTML>
<HEAD>
<TITLE>Ma première page</TITLE>
</HEAD>
<BODY>
Bienvenue sur ma première page !
```

Elle n'est pas très jolie, mais c'est juste pour illustrer les balises de base.  
Ceci est un nouveau paragraphe, mais il n'apparaîtra pas comme tel car il manque de nouvelles balises.

Merci de votre attention.

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

## LES OPTIONS DE BASE DE LA MISE EN FORME

**<P>**. La balise paragraphe permet le découpage du texte en ajoutant l'équivalent de deux retours chariot sur une machine à écrire ou d'un retour à la ligne suivi d'une ligne vide dans un traitement de texte. Les balises paragraphes s'écartent de la structure commune des balises HTML. En effet on peut ne pas définir les paragraphes avec une balise **<P>** de début et une balise **</P>** de fin, mais en codant avec une balise **<P>** pour terminer un paragraphe. C'est simple et légale. L'autre forme doit être connue pour certaines applications avancées d'HTML.

**<BR>**. La balise break permet de passer à la ligne suivante. Vous pouvez l'utiliser afin de revenir à la ligne sans introduire d'espace dans le texte.

EXEMPLE :

```
<HTML>
<HEAD>
<TITLE>Ma première page</TITLE>
</HEAD>
<BODY>
Bienvenue sur ma première page ! <P>
```

Elle n'est pas très jolie, mais c'est juste pour illustrer les balises de base. **<BR>**  
Ceci est un nouveau paragraphe, mais il n'apparaîtra pas comme tel car il manque de nouvelles balises. **<P>**

Merci de votre attention.  
**</BODY>**  
**</HTML>**

## LES EN-TETES

Elles apparaissent avec une police plus grande et plus vive que le reste du document. Le code est très simple.

**<H1>** et **</H1>**. Ces deux balises doivent encadrer le texte que vous voulez mettre en en-têtes. Ce sont des balises de premier niveau donc les plus grosses.

**<H2>** et **</H2>**. Ces deux balises permettent d'obtenir des en-têtes de taille légèrement inférieure.

**<H3>** et **</H3>**. Ces balises permettent d'obtenir le plus petit niveau.

Je recommande plutôt l'utilisation des balises **<H2>** et **<H3>** pour les titres car les proportions sont plus en rapport avec la taille du texte par défaut, que la balise **<H1>**.

Vous pouvez aussi coder des en-têtes **<H4>**, **<H5>** et **<H6>** pour réduire progressivement la taille des titres, mais elles sont de moins en moins utilisées.

```
<HTML>
<HEAD>
<TITLE>Ma première page</TITLE>
</HEAD>
<BODY>
<H2>Bienvenue sur ma première page ! </H2>
```

**<H3>**Elle n'est pas très jolie, mais c'est juste pour illustrer les balises de base. **</H3>**  
Ceci est un nouveau paragraphe, mais il n'apparaîtra pas comme tel car il manque de nouvelles balises. **<P>**

Merci de votre attention.  
**</BODY>**  
**</HTML>**

Remarquez que les coupures de paragraphes ont été supprimées lorsqu'un en-tête était inséré. Ces derniers ajoutent un retour à la ligne, alors la balise **<P>** devient inutile.

Si vous utilisez NOTEPAD ou WORDPAD pour coder en HTML n'oubliez pas de sauvegarder vos œuvres au format texte (.txt) et non .htm ou .html sinon vous ne pourrez plus les ouvrir avec votre éditeur de texte en cliquant dessus. Mais tout n'est pas perdu si vous avez sauvegardé votre travail en \*.html. Vous pouvez ouvrir vos fichiers en cliquant dessus avec le bouton droit de la souris tout en maintenant la touche "SHIFT" appuyée, cela fait apparaître un menu déroulant dans lequel la fonction "Ouvrir avec..." est disponible. Pour visualiser votre page HTML vous devrez en faire une copie en .htm et cliquer dessus pour lancer votre navigateur.

Si vous utilisez EMACS rien de tout ça. Vous pouvez visualiser la page directement sous EMACS. N'oubliez pas de modifier les options dans help pour avoir le texte en couleur, ce qui est plus pratique pour différencier les différentes structures du texte. Ce n'est qu'à la fin que vous sauvegarderez sous un .htm pour exploiter votre travail.

## LES LIAISONS INTERPAGES

`<A>` et `</A>`. C'est avec la balise ancrage que sont créés les liens hypertextes. Son emploi est particulier et elle est suivie de `HREF` puis du nom de l'URL de la page vers laquelle vous voulez créer le lien. Par exemple :

```
<A HREF= "http ://www.passeportformation.com/index.html">Passeport Formation</A>
```

Dans cette liaison, les mots **Passeport Formation** apparaîtront en bleu souligné. Lorsqu'un utilisateur cliquera sur ces mots, il verra apparaître la page nommée `index.htm`.

`HREF` est un attribut de la balise `<A>` qui signifie *hypertext référence*.

## METHODES DE LIAISONS

### LIAISONS RELATIVES

Lorsque vous créez un lien entre deux pages du même ordinateur, il n'est pas nécessaire de spécifier l'adresse Internet complète. Si les deux pages sont dans le même répertoire, il suffit d'utiliser leur nom de fichier HTML :

```
<A HREF= "pagedeux.htm">Allez en page 2.</A>
```

Si le nombre de vos pages est important, vous devrez les enregistrer dans plusieurs répertoires. Alors pour créer un lien vers une page se trouvant dans un autre répertoire, il est inutile de préciser l'adresse Internet. Il sera en effet possible d'utiliser une adresse relative étendue qui spécifiera le répertoire en plus du nom de fichier. Par exemple, vous vous trouvez dans le répertoire de travail suivant :

```
C:\Societe\Developpement\Html\Place
```

Le répertoire `Place` contient deux répertoires : `Images` et `Pages`

Le répertoire `Html` contient deux répertoires : `Photos` et `Vidéo`, en plus du répertoire `Place`.

Vous voulez créer un lien vers une page dans un sous dossier :

```
<A HREF= Pages/pages3.htm>Images expliquées</A>
```

Vous voulez créer un lien vers une page dans un autre dossier :

```
<A HREF=../Photos/photoafric.htm>vue de l'Afrique</A>
```

En HTML le slash (/) est utilisé pour séparer les répertoires comme en Unix ou Mac. N'utilisez pas l'anti-slash (\) comme en Dos ou en Windows. Les deux points (..) signifie : Le dossier contenant le dossier courant.

## LIAISONS ABSOLUES

Si vous voulez créer un lien vers une page contenue sur un autre site, vous devrez noter l'adresse Internet complète de cette page. Exemple :

```
<A HREF="http://www.passeportformation.com/index.html">Passeport Formation</A>
```

**Conseil** : N'oubliez pas de placer dans le bas de toutes vos pages Web un lien permettant le retour vers votre page d'accueil. Cela facilitera la visite de votre site.

## LIAISONS E-MAIL

Avec la balise <A> vous pouvez créer un lien vers votre adresse e-mail. C'est la façon la plus simple de communiquer avec les lecteurs de vos pages Web. Par exemple :

```
<A HREF="mailto:aalexand@free.fr">Envoyez-moi un message e-mail.</A>
```

Si vous voulez faire apparaître votre véritable adresse e-mail, saisissez-la dans l'attribut HREF et dans le message situé entre les deux balises <A> et </A>.

```
<A HREF="mailto:aalexand@free.fr">aalexand@free.fr</A>
```

## LIAISONS INTRAPAGES

C'est la même balise <A> qui est utilisée pour accéder à certaine partie d'une page Web. Cette technique permet d'insérer une table des matières au début d'un document long ou de placer dans le bas d'une page un lien vers le début de celle-ci. Cela vous permettra aussi de créer un lien vers une section d'une autre page Web.

### Les ancrages nommés

L'ajout d'un nom à une balise standard <A> peut apporter un plus à votre liaison. La possibilité est ainsi offerte au lecteur de se brancher sur cet emplacement à partir d'une adresse sur le Web ou d'un emplacement quelconque de votre projet HTML. Par exemple :

```
<A NAME="top">Début de la page</A>
```

Maintenant que votre ancrage est nommé, vous pouvez pointer sur cet ancrage à partir d'une autre section de la page Web en faisant appel à un *lien référencé*. Celui-ci utilisera le nom de l'ancrage plutôt qu'un lien relatif ou absolu pour identifier la cible dans le code HTML. Par exemple :

```
<A HREF="#top">Retour au début du document.</A>
```

Le symbole (#) signifie que le mot " top " représente un ancrage nommé dans le document courant. Lorsqu'un navigateur cliquera sur " Retour au début du document ", le navigateur Web affichera la section qui débute avec la balise <A NAME="top">.

Vous devez bien différencier les deux types de balises <A> :

<A HREF> représente ce sur quoi vous cliquez.

<A NAME> est l'emplacement sur lequel vous êtes dirigé.

Vous pouvez créer un lien vers un ancrage nommé dans une autre page en insérant l'adresse ou le nom de cette page suivi du signe # et du nom de l'ancrage. Par exemple :

```
<A HREF=mapage.htm#un>Vers ma page section un</A>
```

```
<A HREF=./mapage.htm#deux>Vers ma page section deux</A>
```

```
<A HREF=../Photos/photoafric.htm#top>vue de l'Afrique</A>
```

# FORMATAGE ET ALIGNEMENT DE TEXTE

## ENRICHISSEMENT GRAS ET ITALIQUE

**<B>** et **</B>**. Lorsque le texte est entre ces deux balises, il apparaît en gras. C'est la balise bold.

**<I>** et **</I>**. Lorsque le texte est entre ces deux balises, il apparaît en italique. C'est la balise italic.

Vous pouvez cumuler ces deux attribut en encadrant le texte par **<B><I>** puis **</I></B>**. Les en-têtes étant déjà en caractères gras, la balise **<B>** n'a aucun effet sur cette partie du document, mais vous pouvez appliquer l'italique.

## CADRAGE DU TEXTE

Lorsque vous débutez un paragraphe avec la balise **<P>**, par exemple, il est possible de préciser un alignement avec la marge gauche, droite, ou centré.

**<P ALIGN="right">**. Pour aligner à droite le paragraphe.

**<P ALIGN="center">**. Pour centrer le paragraphe.

**<P ALIGN="left">**. Pour aligner à gauche le paragraphe.

Ne pas oublier que si vous débutez le paragraphe avec une balise **<P>** vous devez le finir avec la balise fermante **</P>**.

Vous pouvez définir l'alignement de plusieurs paragraphes ou en-têtes en une seule opération. Il suffit pour cela de déterminer la zone sur laquelle l'attribut devra être appliqué avec les balises :

**<DIV>** et **</DIV>**. Cette balise utilisée seule reste sans effet. Si vous lui associez l'attribut **ALIGN**, tous les paragraphes situés entre :

**<DIV ALIGN="center">** et **</DIV>** seront centrés.

Par exemple :

```
<HTML>
<HEAD>
<TITLE>Ma première page</TITLE>
</HEAD>
<BODY>
<DIV ALIGN="center">
<H2>Bienvenue sur ma première page !</H2>

<H3>Elle n'est pas très jolie, mais c'est juste pour illustrer les balises de base.</H3>
</DIV>
<P ALIGN="left">
Ceci est un nouveau paragraphe, mais il n'apparaîtra pas comme tel car il manque de nouvelles balises.
</P><BR>
<P ALIGN="center">
Merci de votre attention.
</P><BR>
<P ALIGN="right">
<A HREF="http://www.passeportformation.com/index.html">Passeport Formation</A>
</P>
</BODY>
</HTML>
```

Note : La balise **<DIV>** permet grâce aux styles de faire de la mise en page rapide.

## LES MARGES

Bien qu'il existe plusieurs façons de créer des marges, nous ne verrons que la plus simple.



<**BLOCKQUOTE**> et </**BLOCKQUOTE**>. C'est avec cette balise placée au plus près de la balise <BODY> que vous pourrez créer des marges. Par exemple :

```
<HTML>
<HEAD>
<TITLE>Ma première page</TITLE>
</HEAD>
<BODY>
<BLOCKQUOTE>
Bienvenue sur ma première page !<P>
```

Elle n'est pas très jolie, mais c'est juste pour illustrer les balises de base.<BR>  
Ceci est un nouveau paragraphe, mais il n'apparaîtra pas comme tel car il manque de nouvelles balises.<P>

Merci de votre attention.

```
</BLOCKQUOTE>
</BODY>
</HTML>
```

## FORMATAGE DES CARACTERES

<**SMALL**> et </**SMALL**>. Balises pour petite police.

<**BIG**> et </**BIG**>. Balises pour grande police.

<**SUP**> et </**SUP**>. Balises pour exposant.

<**SUB**> et </**SUB**>. Balises pour indice.

<**STRIKE**>. Balise pour rayer (une ligne est dessinée sur le texte).

<**U**> et </**U**>. Balises pour souligner.

<**TT**> et </**TT**>. Balises pour police Courier.

<**PRE**> et </**PRE**>. Balises pour police Courier en conservant les espaces et les retours à la ligne.

<**EM**> ou <**I**> et </**EM**> ou </**I**>. Balises pour italique.

<**STRONG**> ou <**B**> et </**STRONG**> ou </**B**>. Balises pour caractères gras.

Evitez l'utilisation des balises <U> et </U>, le texte souligné représente souvent un lien dynamique.

La balise <TT> applique la plupart du temps la police Courier New. Certains navigateurs Web donnent la possibilité aux utilisateurs de la transformer en une police de leur choix (option des navigateurs). Vous pouvez même remplacer cette police à largeur fixe par une police proportionnelle. La grande majorité des utilisateurs conservent cependant les valeurs standards de leur navigateur. Vous devez donc concevoir et tester vos pages avec ces polices par défaut (la police proportionnelle standard est en général Times New Roman, et la police à chasse fixe standard est presque toujours Courier ou Courier New).

La balise <PRE> fait également apparaître le texte en police à largeur fixe, mais elle permet l'application précise de l'espacement et les retours à la ligne normalement ignorés dans les fichiers HTML.

Note : Maintenant avec l'utilisation des feuilles de style, ce genre de codage a tendance à disparaître.

## TAILLE DES CARACTERES

<**FONT**> et </**FONT**>. C'est cette balise qui définit la taille des caractères. Elle permet un contrôle plus précis de la taille des caractères par rapport aux balises <BIG> et <SMALL>, mais elle n'est pas supportée par les anciens navigateurs. C'est avec son attribut **SIZE** que vous fixez la taille des caractères. Les valeurs de SIZE sont comprises entre 1 (petit) et 7 (très gros). Par exemple

<FONT SIZE=5>Ce texte aura une grande taille.</FONT>

La valeur par défaut de l'attribut SIZE est 3.

Note : Maintenant avec l'utilisation des feuilles de style, ce genre de codage a tendance à disparaître.

## COULEUR DES CARACTERES

Vous pouvez améliorer l'aspect d'une page en définissant certaines polices en couleur. Par exemple :

<FONT SIZE="4" COLOR="purple">La taille de ce texte est 4 et sa couleur est violette.</FONT>

L'attribut color peut-être défini avec les noms de couleur standard suivants :

black(noir), white(blanc), red(rouge), green(vert), blue(bleu), yellow(jaune), aqua(bleu vert), fushia(fushia), gray(gris), lime (vert citron), maroon(marron), purple(violet), navy(bleu marine), olive(vert olive), silver(argent), ou teal(bleu vert).

Sur les nouveaux navigateurs, il est possible d'avoir accès à une palette de couleur plus importante, mais il faut coder l'attribut color en hexadécimal. Par exemple :

<FONT SIZE="4" COLOR="#0033FF">La taille de ce texte est 4 et sa couleur est violette.</FONT>

Note : RVB = 255 X 255 X 255 = 16 581 375 couleurs (donc bien plus que ce que l'oeil humain peut percevoir).

Connectez-vous sur <http://www.december.com/html/spec/color.html>

Note : Maintenant avec l'utilisation des feuilles de style, ce genre de codage a tendance à disparaître.

## LA POLICE DE CARACTERE

L'attribut FACE permet de contrôler l'apparence d'une police et donc d'ajouter une texture à la page sans avoir à manipuler une seule image. La typographie a autant d'impact dans l'élaboration d'une page que les graphiques. Par exemple :

<FONT FACE="arial" SIZE="4" COLOR="#0033FF">La taille de ce texte est 4 et sa couleur est violette.</FONT>

Si la police élaborée n'existe pas, elle sera remplacée par la police par défaut de l'ordinateur de l'utilisateur. Les polices sont légèrement différentes sur MAC et sur PC ainsi que leurs noms. Par exemple, la police Arial est très courante sur les PC, ainsi que son équivalent sur MAC la police Helvetica. Par exemple :

<FONT FACE="arial,Helvetica" SIZE="4" COLOR="#0033FF">La taille de ce texte est 4 et sa couleur est violette.</FONT>

<B>BASEFONT> cette balise définit la taille du texte pour un document complet, en-tête compris. Cette balise ne possède pas d'autres attributs que SIZE, et n'est pas accompagnée de la balise de fin </BASEFONT>.

## LES CARACTERES SPECIAUX

Le mot café sera codé comme suit :

Caf&#233; ou Caf&eacute;

Un tableau vous est fourni. <SUP>TM</SUP> ou <SMALL><SUP>TM</SUP></SMALL> vous permet de créer le symbole de la marque non déposée.

# PRÉSENTATION DES LISTES

## LES LISTES ORDONNÉES OU NUMÉROTÉS

Une liste ordonnée débute avec un nombre et l'incrément à chaque nouvel élément.

`<OL>` cette balise marque le début d'une liste ordonnée et `</OL>` en marque la fin. Les nombre et les retours à la ligne seront insérés automatiquement par la balise `<LI>`.

```
<HTML>
<HEAD>
<TITLE> Les crêpes</TITLE>
</HEAD>
<BODY>
Comment assaisonner les crêpes.
<OL>
<LI> au beurre</LI>
<LI> au sucre </LI>
<LI> à la confiture </LI>
<LI> au miel </LI>
</OL>
</BODY>
</HTML>
```

L'attribut TYPE permet le choix de type de numéro ou de lettre à placer devant chaque élément `TYPE= "A"`, `TYPE= "a"`, `TYPE= "I"`, `TYPE= "i"`, `TYPE= "1"`.

L'attribut START permet de commencer une liste numérotés à partir de n'importe quel chiffre ou lettres. Si vous voulez que votre liste débute à partir de la lettre C vous devrez coder comme suit : `<OL TYPE= "A" START= "3">`.

## LES LISTES A PUCES

Une liste à puces est une liste non ordonnée. L'ordre des informations contenues n'a donc aucune importance. Ces listes sont en général utilisé pour faire ressortir certains points spécifiques d'un texte.

`<UL>` et `</UL>`. C'est ces balises que vous pouvez créer une liste à puces. Le résultat obtenu est identique à celui d'une liste ordonnée mais le symbole sélectionné pour la puce apparaît en lieu et place des nombres.

Plusieurs navigateurs permettent la sélection d'un symbole de puces en spécifiant l'attribut `TYPE= "circle"` `TYPE= "disk"` `TYPE= "square"`.

## LISTE DE DÉFINITION

Les listes de définition ont été conçues à l'origine pour fournir une liste de termes accompagnés de leur signification. Vous pouvez les utiliser pour d'autres applications telles que le décalage de section de texte. Pour créer une liste de définition, vous devez connaître la balise `<DL>` et les balises associées mais également que son utilisation devient de plus en plus rares avec l'évolution de HTML.

Une liste de définition débute avec la balise `<DL>` et se termine avec la balise `</DL>`.

La balise `<DT>` doit précéder chaque terme à définir et la balise `<DD>` doit être placé devant chaque définition. Le décalage et les retours à la ligne seront insérés automatiquement. Les balises `</DT>` et `</DD>` associées sont facultatives.

## AUTRES APPLICATIONS DES LISTES DE DÉFINITION

Les auteurs de pages Web font également appel aux listes de définition pour introduire un décalage dans un texte. Par exemple :

```
<DL><DD>Cet élément sera décalé une fois
<DL><DD>Cet élément sera décalé une fois de plus
<DL><DL><DD>Cet élément sera décalé deux fois de plus
</DL></DL></DL></DL>
```

## HIÉRARCHISATION DES LISTES

Par exemple :

```
<HTML>
<HEAD>
<TITLE> Classement des mammifères </TITLE>
</HEAD>
<BODY>
<OL TYPE= "I"><LI> Les insectivores </LI>
<OL><LI> Taupes </LI>
<LI> Musaraignes </LI>
<OL TYPE= "A"><LI> Genre Crocidura (dents blanches)</LI>
<OL TYPE= "a"><LI> se nourrissent d'insectes et d'autres invertébrés </LI>
</OL>
<LI> Genre Scorex (dents rouges)</LI>
</OL>
<LI> et 426 autres espèces </LI>
</OL>
<LI> Lagomorphes (80 espèces dont lièvres et lapins)</LI>
<LI> Chiroptères (chauve-souris, 925 espèces)</LI>
<LI> Rongeurs (2000 espèces)</LI>
<LI> Artiodactiles </LI>
<OL><LI> Cervidés </LI>
<OL TYPE= "A"><LI> Cerfs </LI>
<OL TYPE= "a"><LI> Cerf sika (origine Extrême-Orient)</LI>
</OL>
<LI> Chevreuil </LI>
<LI> Daim </LI>
</OL>
<LI> Cochons </LI>
<LI> et 218 autres espèces </LI>
</OL>
<LI> Carnivores </LI>
<LI> Cétacés </LI>
</OL>
</BODY></HTML>
```

# INTÉGRATION D'IMAGES

<http://www.iconbazaar.com>

**<IMG>**. Cette balise est très importante pour la manipulation des images. On peut lui associer de nombreux attributs. Elle est composée dans sa forme la plus simple des deux éléments suivants:

la balise **<IMG>** elle-même. Elle signale au navigateurs de nécessité de charger une image. Et n'est associée à aucune balise de fin.

**SRC** cet attribut signifie source ou sources de l'image. Il indique le nom du fichier graphique que vous voulez placer dans la balise. L'attribut SRC présente quelques similitudes avec l'attribut HREF: le premier représente, dans ce cas, une image graphique, alors que le dernier représente une page ou une partie de pages. Par exemple :

```
<IMG SRC="image.gif">
```

Comme pour la balise **<A HREF>** vous pouvez spécifier dans la balise **<IMG SRC>** une adresse Internet complète ou le nom de fichier si l'image se trouve dans le même répertoire que le fichier HTML. Vous pouvez aussi utiliser une adresse relative.

**ALT** cet attribut signifie alternate text. Le message défini avec cet attribut apparaîtra à la place de l'image dans les navigateurs qui ne supporte pas les graphiques. Ou bien lorsque le chargement automatique des images est désactivé. Sous certains navigateurs, c'est une bulle d'aide qui apparaîtra lorsque que la souris passera au-dessus de l'image. Il est donc conseillé d'inclure l'attribut ALT approprié dans chaque balise **<IMG>** de vos pages Web.

Les attributs **WIDTH** (largeur) et **HEIGHT** (hauteur) définissent la taille, en pixels, avec laquelle est enregistrée l'image. Si vous omettez ses attributs le navigateur calculera lui-même la taille de l'image. Le téléchargement est beaucoup plus long avec cette méthode. Il est donc bien plus pratique de préciser à l'avance les dimensions avec les attributs **WIDTH** et **HEIGHT**.

## CADRAGE DES IMAGES

### ALIGNEMENT VERTICAL

**<IMG ALIGN="top">** cette balise permet d'aligner le haut d'une image avec le haut du texte ou de la lettre la plus grande sur la même ligne.

**<IMG ALIGN="bottom">** cette balise permet d'aligner le bas d'une image avec la base du texte.

**<IMG ALIGN="absbottom">** cette balise permet d'aligner le bas de l'image avec la base du texte ou de la lettre la plus basse sur la même ligne. Ceci placera image plus pas que **ALIGN=bottom**.

**<IMG ALIGN="middle">** cette balise aligne le milieu d'une image avec le milieu du texte.

**<IMG ALIGN="absmiddle">** cette balise aligne verticalement le milieu d'une image avec le centre de tous les éléments de la ligne. Le résultat obtenu sera plus haut ou plus bas que **ALIGN=middle**, en fonction de la taille de l'alignement des autres images sur cette ligne.

### ALIGNEMENT HORIZONTAL

Les valeurs **left** et **right** de l'attribut **ALIGN** permettent de cadrer l'image à gauche ou à droite de la page. Par exemple :

```
<IMG SRC="image.gif" ALT="Une belle image" ALIGN="left">
```

### LIAISON D'UNE IMAGE

La balise **<A HREF>** va permettre de transformer une image en un lien vers une autre page. Par exemple :

```
<A HREF="page.htm"> <IMG SRC="image.gif" ALT="Une belle image" ALIGN="left" BORDER=0>
```

HTML entoure toujours d'un cadre standard les images utilisées comme hyperliens (qui correspond au soulignement de hyperliens texte. Pour supprimer ce cadre fort peu esthétique, utiliser l'attribut **BORDER** avec la valeur 0 n'aura commande **<IMG>**.

## COULEUR DU TEXTE ET DES LIENS

Dans la balise <BODY> vous avez la possibilité de spécifier les couleurs du texte, des liens et de l'arrière-plan de page. Par exemple :

```
<BODY BGCOLOR= "red" TEXT= "yellow" LINK= "white" VLINK= "gray" ALINK="green">
```

LINK= "white" met en blanc les liens non encore visités.

VLINK= "gray" met en gris les liens récemment visités.

ALINK= "green" provoque un léger clignotement vert lorsque que l'utilisateur clique sur le lien.

L'attribut BACKGROUND dans la balise <BODY> permet de définir une image d'arrière-plan. Cette image est automatiquement reproduite afin de couvrir la totalité de la page. Par exemple :

```
<BODY BACKGROUND= "image.gif">
```

## LES IMAGES RÉACTIVES

**<MAP>** et **</MAP>** cette balise indique au navigateur que l'image est découpée en zones réactives.

Elle possède l'attribut NAME. Par exemple :

```
<MAP NAME= "nom">
```

**<AREA>** cette balise permet de définir les zones d'une image réactive. Elle possède trois attributs :

SHAPE= "RECT" indique que la zone est rectangulaire.

SHAPE= "POLY" indique que la zone est polygonale.

SHAPE= "CIRCLE" indique que la zone est circulaire.

COORDS= "40,10,130,130" indique les coordonnées décroît supérieur gauche et inférieur droit de la zone rectangulaire.

HREF= "page.htm" définit la page de destination du lien correspondant à la zone. Vous pouvez utiliser une adresse ou un nom de fichier comme vous le feriez pour une balise de liens ordinaires de type **<A HREF>**.

Pour placer l'image réactive dans la page, la balise habituelle **<IMG>** convient à condition de lui ajouter l'attribut USEMAP .  
Par exemple :

```
<IMG SRC= "image.gif" USEMAP= "#nom">
```

Exemple :

```
<HTML>
<HEAD>
<TITLE>Image réactive</TITLE>
<BODY>
<MAP NAME= "nom">
<AREA SHAPE="RECT" COORDS="40,10,130,130" HREF="page1.htm">
<AREA SHAPE="RECT" COORDS="131,60,200,170" HREF="page2.htm">
<AREA SHAPE="RECT" COORDS="201,120,270,240" HREF="page3.htm">
<AREA SHAPE="RECT" COORDS="280,120,380,250" HREF="page4.htm">
</MAP>
<DIV ALIGN="center">
<IMG SRC="image.gif" WIDTH=400 HEIGHT=263 USEMAP="#nom">
</DIV>
</BODY>
</HTML>
```





# LES CADRES

## POURQUOI UTILISER LES CADRES

A la différence d'une cellule de tableaux, un cadre peut contenir des liens qui modifie le contenu d'un autre cadre ou son propre contenu. Par exemple, certains cadres peuvent afficher une table des matières immuable tandis que d'autres changent en fonction des liens suivis par le visiteur. Cela signifie que les cadres peuvent servir à mettre à jour des parties de la page, alors que les parties constantes ne sont pas modifiées; le lecteur gagne en temps de téléchargement.

Un cadre est une zone rectangulaire dans la fenêtre du navigateur qui affiche une page Web, en compagnie d'autres pages, situées dans d'autres cadres.

## LA BALISE <FRAMESET> et </FRAMESET>

Frameset est la commande HTML centrale pour les cadres. Il contient les informations concernant les pages à charger, et il indique dans quel cadre. Aucune balise normalement situées à l'intérieur d'un conteneur <BODY> ne peut se trouver dans une balise <FRAMESET>. Si vous souhaitez que vos cadres soient disposés en rangée ou en colonne vous devrez utiliser l'attribut ROWS ou COLS. Vous devez définir les dimensions de ROWS et COLS soit en pixels, soit en pourcentage de la dimension totale de la fenêtre du navigateur. Vous avez aussi la possibilité d'utiliser l'astérisque (\*) pour indiquer que le cadre doit exploiter l'espace disponible dans la fenêtre. Si vous imposez une valeur \* à plus d'un cadre, l'espace restant sera divisé en parts égales entre eux. Par exemple:

```
<HTML>
<HEAD><TITLE>Frameset</TITLE></HEAD>
<FRAMESET COLS="80,*">
<FRAME SRC="page1.htm" NAME="partiegauche">
<FRAME SRC="page2.htm" NAME="partieprincipale">
</FRAMESET>
</HTML>
```

```
<HTML>
<HEAD><TITLE>Frameset</TITLE></HEAD>
<FRAMESET ROWS="60,*">
<FRAME SRC="page3.htm" NAME="partiehaute">
<FRAME SRC="page2.htm" NAME="partieprincipale">
</FRAMESET>
</HTML>
```

## LA BALISE <FRAME>

Entre les balises <FRAMESET> et </FRAMESET>, il faut insérer une balise <FRAME> indiquant le fichier HTML à afficher pour chaque cadre. Si vous avez moins de balises <FRAME> que de cadres définis au sein de la balise <FRAMESET>, tout cadres excédentaire sera vide. Il n'est pas nécessaire de spécifier une balise de fermeture </FRAME>.

Ajoutez ensuite un attribut SRC dans chaque balise <FRAME>, contenant l'adresse de la page Web correspondant à ce cadre. Vous pouvez insérer l'adresse d'un fichier graphique au lieu d'une page Web si vous souhaitez afficher un cadre ayant une image unique. Vous pouvez afficher dans un cadre de toute page HTML de votre choix.

Vous pouvez donner un nom un cadre à l'aide de l'attribut NAME. Vous pouvez ainsi faire en sorte qu'un lien de la page modifie le contenu de ce cadre grâce à l'attribut TARGET de la balise <A>. Par exemple:

```
<A HREF="page4.htm" TARGET="partieprincipale">Lien vers page 4.</A>
```

Lorsque l'utilisateur clique sur ce lien, lien vers page 4. La page 4 s'affiche dans le cadre nommé "partieprincipale". Si vous n'avez pas ajouté l'attribut TARGET, la page 4 serait affichée dans le cadre de gauche, si le lien appelant s'était trouvé dans la page de gauche.

## CADRES SANS BORDURES ET CADRES IMBRIQUES.

Les cadres sans bordures sont actuellement considérés comme branchés. C'est cette technique qui va les rendre invisibles, permettant ainsi de contrôler mieux que jamais le placement des éléments dans la page.

Malheureusement, Microsoft et Netscape ne se sont pas mis d'accord sur le contrôle des bordures, et le W3C a retenu et rejeté des caractéristiques de l'un et de l'autre dans sa propre solution. Vous voilà donc obligé de jongler avec trois jeux de balise HTML non-standards. Pour MSIE, vous allez supprimer les bordures en ajoutant l'attribut FRAMEBORDERS="NO" dans la balise <FRAMESET>. Pour Netscape Navigator, c'est BORDERS="0" qu'il faut utiliser dans la balise <FRAMESET> pour supprimer les bordures de vos cadres. Quant à HTML 4.0, il vous demande d'inclure l'attribut FRAMEBORDERS="0" dans chaque balise <FRAME> correspondant à un cadre adjacent.

En imbriquant un <FRAMESET> dans un autre, vous pouvez créer des mises en page assez complexes à l'aide de cadre. Par exemple:

```
<HTML>
<HEAD><TITLE>Frameset</TITLE></HEAD>
<FRAMESET ROWS="60,*,60">
<FRAME SRC="page3.htm" NAME="partiehaute">
<FRAMESET COLS="80,*">
<FRAME SRC="page1.htm" NAME="partiegauche">
<FRAME SRC="page2.htm" NAME="partieprincipale">
</FRAMESET>
<FRAME SRC="page4.htm" NAME="partiebasse">
</FRAMESET>
</HTML>
```

## NAVIGATEURS NON COMPATIBLES

Les navigateurs Web compatibles avec les cadres ignorent tout ce qui se trouve entre les balises <NOFRAMES> et </NOFRAMES>. C'est à l'intérieur de ses balises que vous pouvez indiquer un message pour les navigateurs plus anciens non compatibles. Par exemple:

```
<HTML>
<HEAD><TITLE>Frameset</TITLE></HEAD>
<FRAMESET ROWS="60,*,60">
<FRAME SRC="page3.htm" NAME="partiehaute">
<FRAMESET COLS="80,*">
<FRAME SRC="page1.htm" NAME="partiegauche">
<FRAME SRC="page2.htm" NAME="partieprincipale">
</FRAMESET>
<FRAME SRC="page4.htm" NAME="partiebasse">
</FRAMESET>
<NOFRAMES>
<BODY>
Votre navigateur ne lit pas les cadres, mais vous pouvez tout de même visiter notre site.
</BODY>
</NOFRAMES>
</HTML>
```

# LES FORMULAIRES

## DÉFINITION

Les formulaires Web permettent de recevoir des informations de la part des lecteurs. Si vous avez déjà utilisé un moteur de recherche, vous savez déjà ce qu'est un formulaire. Le bouton de commande est également un exemple très répandu d'utilisation de formulaires en ligne. Un formulaire HTML fait partie d'une page Web et comprend des zones dans lesquelles le lecteur peut saisir des informations destinées à l'éditeur de la page Web, c'est-à-dire vous.

## FONCTIONNEMENT

Chaque formulaire contient un bouton de soumission. Lorsqu'il clique dessus, l'utilisateur déclenche le transfert des informations saisies vers une adresse spécifiée dans le formulaire en soi. L'utilisation la plus courante du formulaire reste la collecte de données. Pour vous que les informations vous parviennent, vous devez veiller à l'installation d'un programme spécifique de traitement de formulaires. Demandé à votre prestataire de services Internet ce qu'il propose en matière de script de traitement de formulaires et à quelle adresse exacte vos réponses doivent être expédiées.

## CRÉATION DU FORMULAIRE

Un formulaire commence par la balise <FORM>; celle-ci peut se trouver n'importe où dans le corps du documents HTML . La balise <FORM> reconnaît en principe deux attributs, METHOD et ACTION. Par exemple :

```
<FORM METHOD="POST" ACTION="/cgi-bin/scriptcgi.exe">
```

Actuellement, l'attribut METHOD est presque toujours équivalent à POST, ce qui revient à envoyer les données saisies en tant que document. Dans certains cas précis, vous utiliserez peut-être METHOD="GET", qui soumet les résultats en tant que partie de l'URL d'en-tête. "get" est parfois utilisé pour soumettre des requêtes à des moteurs de recherche à partir de formulaire Web.

L'attribut ACTION spécifie l'adresse de programme ou de script résidant sur le serveur, qui sera chargé du traitement des informations saisies dans le formulaire par l'utilisateur. Il s'agit de l'adresse fournie par votre prestataire de services ou votre administrateur Web.

Pour demander à l'utilisateur d'indiquer une information précise, utiliser la balise <INPUT>. Celle-ci doit se situer entre les balises <FORM> et </FORM>, mais elle peut se trouver n'importe où dans la page par rapport au texte, aux images, et aux autres balises HTML.

L'attribut TYPE indique le type de champ de saisies à afficher, qui peut-être une simple ligne de texte. Par exemple :

```
Quel est votre prénom? <INPUT TYPE="text" NAME="prenom">
```

```
quel est votre nom ? <INPUT TYPE="text" NAME="nom">
```

Les cases à cocher: par exemple :

```
<INPUT TYPE="checkbox" NAME="chat" VALUE="yes">j'aime les chats.
```

```
<INPUT TYPE="checkbox" NAME="chien">j'aime les chiens.
```

Les boutons radio : au contraire des cases à cocher qui permettent des choix multiples dans une liste options, les boutons radio n'autorisent qu'un seul choix et sont presque aussi simple à mettre en oeuvre. Par exemple :

```
<INPUT TYPE="radio" NAME="animal " VALUE="chats">j'aime les chats.
```

```
<INPUT TYPE="radio" NAME=" animal " VALUE="chiens">j'aime les chiens.
```

Les listes de choix : les listes déroulantes et autres listes de sélection sont créées à l'aide de la balise <SELECT>. Vous devez utiliser celle-ci avec <OPTION>. Par exemple :

```
<SELECT NAME="animal" SIZE=2 MULTIPLE>
```

```
<OPTION SELECTED>loup
```

```
<OPTION>chien
```

```
<OPTION>chat
```

```
<OPTION>renard
```

```
</SELECT>
```

Vous ne devez insérer que des balises <OPTION> entre <SELECT> et </SELECT>.

L'attribut SIZE détermine le nombre d'éléments à afficher simultanément dans la liste de sélection.



## EXEMPLE DE FICHIER POUR REPONDRE AU FORMULAIRE

Attention ce fichier est écrit en langage de programmation. Il n'est donc pas compris par un navigateur.

```
#!/usr/bin/perl -w
# cgi-bin/repform.pl : ceci est un programme de réponse et créant un formulaire
# contenant la réponse. (Alain ALEXANDRE version 1)

use strict;
use CGI qw(:standard);

print header, start_html("reponse au formulaire"), h2("Bonjour");
if (param()) {# le formulaire a ete rempli
    my $name = param("nom");
    my $firstname = param("prenom");
    my $town = param("ville");
    print p("$firstname $name, je suis tres heureux que vous ayez choisi ce stage. $town est une tres jolie ville. Vos autres reponses
ne sont pas traitees!");
} else {# le formulaire n'a pas ete rempli
    print p("veuillez remplir le formulaire");
}
print end_html;
```

# POUR AMÉLIORER VOS PAGES WEB

## JAVA

Les applications complexes, quelle que soit leur origine, sont peu propices à une intégration dans une page HTML par le biais d'un langage de script tel que Javascript. Elles occupent tellement de ligne de code qu'elles noient totalement la page Web elle-même.

À l'instar de Javascript, Java a été spécialement conçu pour le Web. Et, tout comme les scripts écrits en Javascript, les programmes Java s'installent et s'exécutent automatiquement lors du chargement de la page Web. Les programmes Java présentent l'avantage supplémentaire d'être compilés sous une forme compacte et efficace (appelée bytecode) et stockés dans un fichier distinct de la page Web qui va les appeler. Le seul facteur réellement limitatif pour ces mini-applications en langage Java, que l'on nomme applets, est que ces dernières doivent être réellement réduites afin de ne pas retarder l'affichage de la page Web au-delà de l'acceptable. Heureusement, les applets Java sont très compactes sous leur forme compilée et occupent souvent nettement moins d'espace que les images habituellement incluses dans une page Web.

**<APPLET>** et **</APPLET>**. C'est entre ces deux balises, que le code d'appel de l'applet sera inclus. L'attribut **CODE** désigne le nom du fichier de l'applet appelée. Ce fichier doit se trouver dans le même répertoire que la page appelante. Les attributs **WIDTH** et **HEIGHT** spécifient les dimensions de la zone contenant le résultat produit par l'applet dans la page Web.

**<PARAM>** cette balise sert à donner à l'applet les informations sur lesquelles elle doit agir. La plupart des applets exigent plus d'une balise **<PARAM>** pour établir l'ensemble des options. En principe le concepteur de l'applet doit vous indiquer quels sont les attributs **NAME** à insérer, ainsi que les informations à affecter aux attribut **VALUE** de chaque **NAME**.

Par exemple:

```
<HTML>
<HEAD>
<TITLE>Intégration de Java</TITLE>
</HEAD>
<BODY>
<APPLET CODE=TicTacToe.class WIDTH=120 HEIGHT=120>
</APPLET>
<APPLET CODE="Clock2.class" WIDTH=170 HEIGHT=150>
  <PARAM NAME=bgcolor VALUE="000000">
  <PARAM NAME=fgcolor1 VALUE="ff0000">
  <PARAM NAME=fgcolor2 VALUE="ff00ff">
</APPLET>
</BODY>
</HTML>
```

## FLASH

Vous pouvez aussi intégrer des animations de type flash dans vos pages html. Il vous faudra alors utiliser la balise **<OBJECT>** et **</OBJECT>** ainsi que la balise **<EMBED>** et **</EMBED>** pour la compatibilité entre les différents navigateurs.

Exemple :

```
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
  codebase="http://active.macromedia.com/flash2/cabs/swflash.cab#version=4,0,0,0"
  ID=objects WIDTH=550 HEIGHT=320>
  <param NAME=movie VALUE="Images/marctotal.swf">
  <param name="LOOP" value="false">
  <param NAME=quality VALUE=high>
  <param NAME=devicefont VALUE=true>
  <param NAME=bgcolor VALUE=#FFFFFF>
  <param NAME=ALIGN VALUE=CENTER>
  <param NAME=VALIGN VALUE=MIDDLE>
  <br>
```

```
<embed src="Images/marctotal.swf" quality=high devicefont=true bgcolor=#FFFFFF ALIGN="CENTER"
VALIGN="MIDDLE" WIDTH=600 HEIGHT=350 LOOP="false" TYPE="application/x-shockwave-flash"
PLUGINSPAGE="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=ShockwaveFlash"></emb
ed>
</object>
```

## JAVASCRIPT

Un langage de script de un moyen de réaliser des opérations complexes du côté client de votre navigateur, comme modifier l'apparence de la page, afficher des informations, gérer des formulaires, des cadres et d'autres éléments qu'il peut être difficile et long de coder manuellement et qui seraient gourmands en capacités serveur. Les scripts sont plus appropriés et rapides, pour presque tout ce que vous pouvez mettre en place côté client, et sont beaucoup plus puissants. Les scripts viennent donc à point nommé pour tout ce qui concerne l'interactivité de vos pages Web.

Il existe de nombreux langages de scripts : JavaScript, VBScript, JScript, etc.

Si votre script contient des variables, elles doivent être initialisées dans la partie <HEAD> de votre page HTML et entre les balises <SCRIPT> et </SCRIPT>. L'attribut LANGUAGE informe le navigateur du langage de script utilisé. Le script est toujours placé entre des balises de commentaire (<!-- ... //-->), ce qui permet de les cacher aux anciens navigateurs non compatibles avec la balise <SCRIPT>.

Par Exemple:

```
<HTML>
<HEAD>
<SCRIPTt LANGUAGE="javascript">
<!--
image1 = new Image(100,100);
image2 = new Image(100,100);
image3 = new Image(100,100);
image4 = new Image(100,100);
image5 = new Image(100,100);
image6 = new Image(100,100);
image1.src = "crosbone.gif";
image2.src = "convert3.gif";
image3.src = "dafo08.gif";
image4.src = "dafo09.gif";
image5.src = "crosbone.gif";
image6.src = "dafo08.gif";
//-->
</SCRIPT>

</HEAD>
<BODY>
<a href="#" onMouseOver = "image5.src = image2.src" onMouseOut = "image5.src =
image1.src"></a>
<a href="#" onMouseOver = "image6.src = image4.src" onMouseOut = "image6.src =
image3.src"></a>
<!--
//-->
</BODY>
</HTML>
```

Les six premières lignes créent six variables, les six lignes suivantes les initialisent en leur affectant des images.

Pour qu'il se passe quelque chose un événement doit se produire. Les deux événements utilisés sont assez courant, il s'agit de OnMouseOver et de OnMouseOut. Le premier se produit lorsque le curseur de la souris entre dans une zone désignée. Le second se produit lorsque le curseur la quitte.

Un événement est une action dont un script ou un programme attend la manifestation. Lorsqu'un événement se produit, cela déclenche l'exécution du script.

Deuxième exemple:

```
<HTML>
```

```

<HEAD>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
  <meta name="GENERATOR" content="Mozilla/4.5 [fr] (Win95; I) [Netscape]">
  <meta name="Author" content="Alain ALEXANDRE">
  <TITLE>bilan</TITLE>
<!-- DATE DE CREATION: 11/05/99 -->
<SCRIPT LANGUAGE="JavaScript">
<!-- Début du script
var ms = navigator.appVersion.indexOf("MSIE")
ie4 = (ms>0) && (parseInt(navigator.appVersion.substring(ms+5, ms+6)) >= 4)

function deplace3(from, to) {
  if (from > to) {
    document.all.texte3.style.top = (from -= 10);
    setTimeout('deplace3(' + from + ', ' + to + ')', 75);
  }
}
function deplacen3(from, to) {
  if (from > to) {
    document.texte3.top = (from -= 10);
    setTimeout('deplacen3(' + from + ', ' + to + ')', 75);
  }
}
// Fin du script -->
</SCRIPT>
<SCRIPT LANGUAGE="JavaScript">
<!-- Début du script
var ms = navigator.appVersion.indexOf("MSIE")
ie4 = (ms>0) && (parseInt(navigator.appVersion.substring(ms+5, ms+6)) >= 4)

function deplace4(from, to) {
  if (from < to) {
    document.all.texte4.style.top = (from += 10);
    setTimeout('deplace4(' + from + ', ' + to + ')', 75);
  }
}
function deplacen4(from, to) {
  if (from < to) {
    document.texte4.top = (from += 10);
    setTimeout('deplacen4(' + from + ', ' + to + ')', 75);
  }
}
// Fin du script -->
</SCRIPT>
</HEAD>
<body text="#000000" bgcolor="#B5DEB5" link="#0000EE" vlink="#551A8B" alink="#FF0000" onLoad="ie4?deplace4
(-100,150,20):deplacen4(-100,150,20);ie4?deplace3(450,150,20):deplacen3(450,150,20)">

<div ID="texte3" STYLE="text-align:center;font-size:50pt;font-
family:TimesNewRoman;color:#0000FF;position:relative"><i>LE
BILAN</i></div>

<div ID="texte4" STYLE="text-align:center;font-size:14pt;font-family:Times New
Roman;color:#000000;position:relative"><i></i>&nbsp;</div>

</BODY>
</HTML>

```