

HTML (HyperText Markup Language) est un langage de description de documents. Quelques autres langages, ayant la même vocation existaient au démarrage du World Wide Web, mais le besoin de définir un nouveau langage s'est imposé : il fallait fournir une fonctionnalité supplémentaire : la navigation hypertexte. La navigation hypertexte consiste à fournir la possibilité de cliquer sur certaines zones du document (les [liens hypertextes](#)) pour accéder une autre partie du site, voir un autre site Web.

### Les tags en HTML

Rien n'est plus facile que d'écrire un document HTML : il vous suffit d'écrire votre texte dans un fichier (d'extension .htm ou .html). Cependant, il existe des instructions HTML permettant de rendre un document plus complet, plus agréable. Toute instruction HTML, pour être reconnue du texte proprement dit, doit être entourée par les caractères < et > : **<HR>** est donc une instruction du langage HTML (elle permet de placer un ligne horizontale de séparation dans votre document). On nomme aussi **tag (marque en français)** une telle instruction : c'est d'ailleurs la terminologie que j'emploierai par la suite de ce cours. Pour compliquer un peu les choses, un tag HTML peut prendre des paramètres, chacun d'eux ayant une valeur par défaut et un type particulier (valeur numérique, chaîne de caractères, [valeur RGB pour les couleurs](#), [URLs](#), ...). L'instruction citée précédemment, aura put donc s'écrire **<HR Align="left">** : on fixe ainsi l'alignement de la ligne de séparation à gauche (mais "left" est la valeur par défaut du paramètre **Align**).

Remarque : le langage HTML ne fait pas la distinction, au niveau des tags HTML, entre minuscules et majuscules ! Cependant tous les exemples de ce cours auront (sauf fautes de frappe) les tags HTML marqués en majuscules, les paramètres commençant par une majuscule mais finissant par des minuscules et les valeurs de paramètres en minuscules. Si vous mettez tout en minuscules ou autre, cela ne changera normalement rien à l'affichage final de votre document.

### Les tags <HTML> ... </HTML>

Un document HTML doit débuter par le tag **<HTML>** et finir par la marque **</HTML>**. Entre ces deux marques, on trouve essentiellement deux parties. La première est [l'en-tête](#) : celle-ci ne contient normalement aucun élément visuel du document, mais sert plutôt à donner des indications quant-au document en cours d'affichage dans le navigateur WEB. La seconde, [le corps](#), contient le contenu proprement dit de la page. Le corps du document doit suivre l'en-tête (logique, non ?). Voici un petit exemple de définition de document HTML.

```
<HTML>
  <HEAD>
    <!--En-tête du document-->
  </HEAD>
  <BODY>
    Ceci est le début de votre document
    ...
  </BODY>
</HTML>
```

### Les commentaires

Quelle que soit la partie HTML considérée, il vous est possible de commenter votre code HTML. Tout texte inclut entre **<!--** et **-->** sera considéré comme étant un commentaire, et ne sera donc pas affiché dans le document. Un commentaire peut servir pour laisser une indication (par exemple) sur votre code HTML, au cas où vous il faudrait apporter des modifications dans le code HTML.

```
<!-- Ceci est un commentaire HTML -->
```

### Caractères spéciaux

L'intérêt du langage HTML est de pouvoir écrire des documents visualisables quelle que soit la plateforme utilisée (Système Unix, Windows, ...). Un problème se pose alors : la table de caractères ASCII souvent utilisée, ne permet que de code 128 caractères. Les caractères accentués ne font pas partie de cette table de caractères. Cependant, chaque plate-forme a étendu, à sa guise, le jeu de caractères ASCII à

256 caractères : bien entendu, simple e accentué (é ou è) n'est pas codé de la même manière. Pour palier à ce petit problème, des séquences de caractères ont été définies pour pouvoir spécifier un caractère non ASCII. Le tableau qui suit, donne les correspondances entre les séquences de caractères et le caractère sous-jacent (notez que les signes < et >, utilisés par le langage, sont donc eux aussi utilisables dans vos documents par le biais d'une séquence particulière).

Séquences	Caractères	Séquences	Caractères
&lt;	<	&acute;	é
&gt;	>	&egrave;	è
&amp;	&	&ecirc;	ê
&quot;	"	&ocirc;	ô
&agrave;	à	&ugrave;	ù
&acirc;	â	&ucirc;	û
&icirc;	î	Û	ü
&ccedil;	ç		

L'en-tête d'un fichier HTML sert à définir diverses choses propres à la page HTML, mais qui n'ont pas forcément d'incidence sur le contenu de la page, comme par exemple le titre apparaissant dans la barre de titre du navigateur WEB. On définit un en-tête de fichier HTML par le couple de tags **<HEAD>** ... **</HEAD>**. Voici donc une liste des tags les plus couramment utilisés dans un en-tête de document.

### Donner un titre à son document

Il vous est possible de donner un titre à votre document : celui-ci apparaîtra dans le bandeau supérieur de la fenêtre de votre navigateur (mais pas dans le document). Notez aussi, que si vous cherchez à faire indexer vos documents par des moteurs de recherche sur Internet, le titre de votre document est une information capitale : n'oubliez donc pas de bien le définir.

Ce titre s'introduit dans l'en-tête de votre document par le couple de tags **<TITLE>** ... **</TITLE>**. La spécification du [W3C](#) dit que celui-ci doit être obligatoire et unique (pas de double définition.

ci-dessous montre le squelette d'un document HTML comportant un titre.

```
<HTML>
  <HEAD>
    <TITLE>Ceci est le titre de mon document</TITLE>
  </HEAD>
  <BODY>
    Ceci est le début de votre document
    ...
  </BODY>
</HTML>
```

### Base de résolution des liens hypertextes

Il y a deux façons de définir un lien hypertexte. Soit on spécifie un lien absolu (par exemple « <http://www.voila.fr> », soit on définit un lien hypertexte en relatif (normalement, en rapport à d'où vient la page contenant le lien). dans ce dernier cas, il est possible que tout lien relatif soit résolu non plus en rapport à la localisation de la page contenant le lien, mais en rapport à un autre base de résolution de lien (une autre adresse de base). Pour ce faire, il vous suffit d'utiliser le tag **<BASE>**. Ce tag admet deux paramètres : *Href* et *Target*.

- **HREF** : Ce paramètre permet donc de déterminer la base de résolution des liens du document, qui sont non complètement spécifier (lien relatif). A titre d'exemple, considérons le cas suivant. Une page quelconque contient le lien hypertexte "page1.html". Dans cette même page est mentionné le tag **<BASE Href="http://www.serveur.fr/rep/">**. Ainsi, si une personne active ce lien, elle sera redirigée vers le document ciblée par l'**URL** <http://www.serveur.fr/rep/page1.html>.
- **TARGET** : Ce paramètre à un comportement sensiblement identique au précédent. Mais à la place

de déterminer une localisation du document, il sert à choisir dans quelle fenêtre les documents pointés par les liens doivent (par défaut) s'ouvrir. Ce paramètre est très utile dans le cas où les [cadres](#) sont utilisés, mais il peut aussi servir à localiser une autre fenêtre indépendante.

### Le meta-tag

<META> : ce tag sert à définir des couples nom-valeur relatifs aux propriétés du document. Voici quelques exemples plus parlant. Des informations supplémentaires suivent.

```
<META NAME="AUTHOR" CONTENT="Jean DUPONT">
<META NAME="GENERATOR" CONTENT="un générateur quelconque">
<META HTTP-EQUIV="Pragma" CONTENT="no-cache">
<META HTTP-EQUIV="Refresh"
  CONTENT="3; URL=http://une.adresse.quelconque">
<META HTTP-EQUIV="Content-Type"
  CONTENT="text/html; charset=iso-8859-1">
<META HTTP-EQUIV="Expires"
  CONTENT="Tue, 10 Jul 1997 20:00:00 GMT">
```

La troisième ligne permet de faire en sorte que la page ne soit pas sauvegardée dans le cache du navigateur (utile dans certains cas). La ligne suivante est aussi très utile : elle permet de forcer un changement de document automatique au bout d'un certain nombre de secondes.

### Références à d'autres types de langage

#### Les langages de script

Il est possible de rajouter des scripts à une page HTML. Ces scripts permettent d'ajouter un comportement dynamique à votre document (validation de données saisies au sein d'un formulaire, animation graphique, ...). Il existe plusieurs langages utilisables pour décrire ces scripts : les plus connus sont [Javascript](#) et VBScript.

Pour insérer de tels scripts, il suffit d'insérer un couple de tags <SCRIPT>...</SCRIPT>. Le tag <SCRIPT> accepte au moins les deux paramètres suivants.

- Language : ce paramètre permet de spécifier le nom du langage (et le numéro de version) utilisé par le script.
- Src : permet éventuellement de spécifier un nom de fichier externe contenant le script.

```
<SCRIPT LANGUAGE="Javascript">
<!--
  window.alert("Ceci est un petit script !");
//-->
</SCRIPT>
```

#### Les feuilles de style

Dans les dernières versions de navigateurs, il vous est maintenant possible de dissocier la définition du contenu de votre document, de la spécification de sa présentation dans le navigateur. Pour ce faire un nouveau langage a été défini : [Cascading Style Sheet \(un cours sur ce langage est disponible\)](#).

En fait, il y a trois façons de définir un style. Un de ces trois façon consiste à définir une feuille de style qui sera contenu directement dans le document. Pour ce faire, il faut utiliser le couple de tags <STYLE>...</STYLE>.

#### Lier un autre fichier

Il est possible de lier un fichier au document. C'est ce qui est systématiquement fait pour attacher une feuille de style à une page web.

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

Le corps d'un fichier HTML est introduit par le tag <BODY>, et se termine par </BODY>. Ce couple de tag qui succède à [l'en-tête du document HTML](#), est normalement inclut dans un couple <HTML> . . . </HTML>.

Ce tag peut prendre un certain nombre de paramètres qui permettent de spécifier les caractères généraux de la page tels que les différentes [couleurs](#) utilisées dans la page. Notez un détail important, l'utilisation de ces paramètres est maintenant dépréciée au profit de l'utilisation de [feuilles de style CSS](#). Malgré cela, voici la description de quelques uns de ces paramètres.

- BACKGROUND : ce paramètre permet de spécifier le motif pour le fond de votre page. Il est important de noter que les dimensions de votre motif (au format [GIF](#) ou [JPEG](#)) ne sont pas importantes. En effet, si le motif est plus grand que la page alors il sera tronqué. Dans le cas inverse où le motif est plus petit, alors il sera répété autant de fois qu'il sera nécessaire pour couvrir la totalité de la page.
- BGCOLOR : cet autre paramètre est utilisé pour spécifier la [couleur](#) de fond de votre page. En présence d'un paramètre 'BACKGROUND' c'est l'image de fond qui prévaut sur la couleur de fond. Malgré tout, il peut être intéressant de spécifier les deux paramètres. En effet, si la personne qui visualise votre page est distante, ou si le réseau est encombré, les images vont arriver avec un certain décalage par rapport au code HTML : il pourra donc être judicieux d'avoir, durant ce laps de temps, une [couleur](#) rappelant les nuances du motif de fond, afin d'assurer une certaine [cohésion](#) pour le document.
- TEXT : permet de fixer la [couleur](#) par défaut du texte de la page HTML. Il est bien entendu possible d'écrire des pages avec des textes de différentes [couleurs](#), mais en l'absence de spécification, celle définie par défaut sera prise. Si ce paramètre n'est pas initialisé, c'est la [couleur](#) par défaut du navigateur Web qui sera prise le cas échéant.
- LINK : de la même manière, ce paramètre permet de définir la [couleur](#) des [liens hypertextes](#) non encore explorés.
- VLINK : spécifie la [couleur](#) des [liens hypertextes](#) déjà visités.
- ALINK : ce dernier paramètre permet donc définir la [couleur](#) que prendront les [liens hypertextes](#) lorsqu'ils seront sélectionnés par un click souris ou bien le clavier (un instant relativement bref).

A titre d'exemple, voici ce que pourrait être un tag 'BODY' pour un document sur fond blanc, et dont la couleur des textes tendrait vers un bleu turquoise.

```
<HTML>
  <HEAD>
    <TITLE>The title</TITLE>
  </HEAD>
  <BODY BgColor="#FFFFFF" Text="#0080C0" Link="#00A0FF"
VLink="#00C0FF">
    <!-- suite du document HTML -->
  </BODY>
</HTML>
```

Nous allons, dans ce chapitre étudier de nouveaux tags. Ces derniers servent à structurer le document (le mettre en forme). Vous ne devez donc le utiliser qu'uniquement dans le corps des document HTML (et non dans l'entête).

<CENTER> ... </CENTER> : ce couple sert à centrer une zone du document, en rapport à la largeur de la fenêtre. La ligne qui suit vous donne un exemple d'utilisation du centrage.

Essai de centrage

<DIV> ... </DIV> : permet de déterminer l'alignement d'un bloc de texte. Quatre valeurs sont possibles pour l'unique paramètre ALIGN de la marque <DIV> : "left" (par défaut), "center", "right" ou "justify". Il est bien clair que l'effet de l'utilisation de la valeur "center" est équivalent à celui produit par l'utilisation du couple de marques <CENTER> ... </CENTER>

<H1> ... </H1> ; <H2> ...</H2> ; ... ; <H6> ... </H6> : ces différents couples de tags permettent d'introduire de titres dans vos documents. Il existe donc six niveau de titre en HTML (les titre de niveau 1 indique une importance plus élevée que pour pour un titre de niveau 2 et ainsi de suite). Si vous regardez au plus haut de ce document, vous verrez un titre de premier niveau ("Mise en forme d'un document HTML").

Vous pouvez de plus utiliser le paramètre ALIGN , pour chacune des marques <H1> ... <H6>, afin de choisir l'alignement du titre. Quatre valeurs sont possibles : "left" (aligné à gauche : par défaut), "center" (centré par rapport à la largeur de la fenêtre), "right" (aligné à droite) et

## INTERNET

## LES BASES DU LANGAGE HTML

"justify" (qui permet de justifier la portion de texte). Notez cependant que l'utilisation de ce paramètre est maintenant dépréciée au profit de l'utilisation des [feuilles de styles CSS](#).

<P> ... </P> : ce couple là permet d'introduire un paragraphe dans le document. Pour ce faire, un retour à la ligne est effectué au début et à la fin du paragraphe. La marque <P> admet un paramètre ALIGN dont l'utilisation est la même que pour la marque précédente (<DIV>). Il est donc clair que pour les deux lignes de code HTML suivantes, le résultat sera le même.

- o <DIV ALIGN="justify"><BR> blablabla <BR></DIV>
- o <P ALIGN="justify"> blablabla </P>

<UL> ... </UL> : permettent d'introduire une liste (non énumérée) d'éléments. Ces éléments (introduits par la marque <LI>) seront présentés décalés vers la droite avec une pastille de forme diverse selon le niveau de profondeur de la liste (une liste peut être insérée dans une autre).

```
blabla
<UL>
  <LI>élément      1   o   élément 1
  <LI>élément      2   o   élément 2
  <LI>élément      3   o   élément 3
</UL>
blabla
```

<OL> ... </OL> : de même que le couple précédent, celui-ci permet de créer une liste énumérée. Son utilisation reste la même.

```
blabla
<OL>
  <LI>élément      1   6. élément 1
  <LI>élément      2   7. élément 2
  <LI>élément      3   8. élément 3
</OL>
blabla
```

<LI> : comme précédemment vu, sert à définir un élément d'une liste.

<PRE> ... </PRE> : on utilise ce couple afin de pouvoir visualiser un texte formaté. Une fonte de caractères spéciale est alors utilisée et tous les blancs saisis dans le texte, ainsi que les retours à la ligne seront visualisés dans le document. On utilise généralement ce mode de présentation pour la visualisation des morceaux de code de programme dans un document.

```
<PRE>A
Blablabla
C</PRE>
A
Blablabla
C
```

<BR> : cette marque sert à effectuer un retour à la ligne suivante dans le document.

<HR> : permet de séparer deux parties d'un document HTML par une ligne horizontale. Cette marque admet plusieurs paramètres dont voici une brève description.

- o ALIGN : permet de placer différemment la barre de séparation. Trois valeurs sont possibles : "left", "center" ou "right".
- o SIZE : spécifie la hauteur en pixel de la barre de séparation.
- o WIDTH : on utilise ce paramètre afin de fixer la largeur de la barre de séparation. Soit on spécifie la taille en pixel (ex: WIDTH="400"), soit on la donne en proportion de la largeur de la fenêtre (ex: WIDTH="80%").
- o NOSHADE : ce paramètre est utilisé pour dessiner la barre de séparation d'une couleur unique (pas d'effet d'ombrage).

Nous allons maintenant nous attaquer à l'étude d'un certains nombres de tags HTML qui permettent d'apporter un style à votre document. En effet nous allons voir comment définir les polices de caractères à utiliser, comment choisir les couleurs de vos textes, ... En clair comment rendre vos pages agréables. Cependant, il faut noter une chose importante : beaucoup de ces tags sont maintenant dépréciés au profit de l'utilisation des feuilles styles CSS. En effet ce vous permet de séparer l'aspect structuration de votre document de l'aspect présentation. Ainsi, si toutes les pages d'un même site utilisent toutes la (ou

## INTERNET

## LES BASES DU LANGAGE HTML

les) même feuilles de style, il sera très simple de changer radicalement la présentation de ces document. Notez que la majeure partie des cours de ce site utilise une unique feuille de style. Ainsi, si un jour je souhaite changer le look de l'Infini, je peux en avoir seulement pour quelques heures de travail (l'Infini représente pour l'heure 5 Mo de textes). Pour de plus amples informations sur les feuilles de styles, [je vous conseille de jeter un coup d'œil sur le cours en question](#).

<FONT> ... </FONT> : ce couple de tags permet de spécifier les caractéristiques de la fonte de caractères utilisée pour un groupe de mots, ce au travers de quelques paramètres dont la définition suit. Il est bien entendu que tous ces paramètres peuvent être cumulés.

- o SIZE : permet de préciser la taille de la police de caractères. Les valeurs acceptées vont de 1 (taille la plus petite) à 7 (taille la plus grande).
- o FACE : on utilise ce paramètre pour choisir la police de caractères.
- o COLOR : permet de définir une [couleur](#) pour la section de texte.

<FONT Size=3 Face="arial" Color="#008080"> du texte </FONT> =>  
du texte

<BIG> ... </BIG> : on utilise ce couple de tags dans le but d'augmenter la taille de la police de caractères pour une portion de texte donnée. Cette augmentation est relative par rapport à la taille courante de la police.

normal <BIG>grand</BIG> normal => normal grand normal

<SMALL> ... </SMALL> : de la même manière, ce couple permet d'écrire une portion de texte dans une police de taille inférieure.

normal <SMALL>petit</SMALL> normal => normal petit normal

<U> ... </U> : ce couple de tags est utilisé pour souligner une section de texte, afin de faire ressortir une importance accrue. Attention, les liens hypertextes sont très souvent soulignés, mais cela est fait automatiquement par [le tag HTML <A>](#) que nous étudierons par la suite.

<U>encore du texte</U> => [encore du texte](#)

<B> ... </B> : ce couple sert à mettre une section de texte en gras, comme le montre l'exemple qui suit.

encore <B>plus</B> => de texte encore **plus** de texte

<I> ... </I> : ces deux tags servent à mettre une section de texte en italique.

<I>un autre texte</I> => *un autre texte*

<S> ... </S> ou <STRIKE> ... </STRIKE> : ces couples de tags sont utilisés pour rayer une section de texte.

<STRIKE>des caractères rayés</STRIKE> =>  
~~des caractères rayés~~

<BLINK> ... </BLINK> : on utilise ce couple de tags afin de rendre une section de texte clignotant. Attention, tous les navigateurs ne supportent pas ce couple de tags.

<BLINK>ça clignote</BLINK> => ça clignote

<TT> ... </TT> : ces deux tags servent à inclure une portion de texte dite formatée : une police particulière est utilisée pour faire ressortir le texte, dans laquelle toutes les lettres ont la même largeur et la même hauteur (pour introduire un paragraphe complet formaté on préfère utiliser [<PRE> ... </PRE>](#)).

<TT>du texte formaté</TT> => du texte formaté

<SUB> ... </SUB> : permet de mettre une section de texte en indice.

Normal<SUB>Indice</SUB> => Normal<sub>Indice</sub>

<SUP> ... </SUP> : permet de mettre une section de texte en exposant.

Normal<SUP>Exposant</SUP> => Normal<sup>Exposant</sup>

Pour les couples de tags suivants, le résultat ne sera pas forcément identique d'un navigateur à un autre : la décision du rendu est laissée libre aux concepteurs du navigateur.

<EM> ... </EM> : on utilise ce couple pour mettre une section de texte en valeur (souvent visualisé en italique).

<EM>Du texte en valeur</EM> => *Du texte en valeur*

<CITE> ... </CITE> : utilisées pour insérer une citation dans le document (en général, visualisé en italique).

`<CITE>Citation</CITE>` => *Citation*

`<ADDRESS> ... </ADDRESS>` : ce couple est lui utilisé pour faire ressortir les adresses URL dans le document (en général, visualisé en italique).

`<ADDRESS>dliard@pop.multimania.com</ADDRESS>` =>  
*dliard@pop.multimania.com*

En HTML, vous avez deux façons de spécifier une couleur. La première consiste à donner les trois composantes RGB de la couleur désirée sous le format "#RRGGBB", où RR représente l'intensité de rouge entre 00 et FF (en hexadécimal), GG l'intensité de vert et BB celle du bleu : il vous est donc possible de définir jusqu'à 16 581 375 couleurs. La seconde méthode, plus simple, consiste à simplement nommer la couleur désirée. Le tableau suivant vous donne quelques nom de couleurs connues du langage HTML, ainsi que les correspondances RGB et une pastille colorée vous permettant d'apprécier chaque couleur.

Black	"#000000"	
Silver	"#C0C0C0"	
Gray	"#808080"	
White	"#FFFFFF"	
Maroon	"#800000"	
Red	"#FF0000"	
Purple	"#800080"	
Fuchsia	"#FF00FF"	
Green	"#008000"	
Lime	"#00FF00"	
Olive	"#808000"	
Yellow	"#FFFF00"	
Navy	"#000080"	
Blue	"#0000FF"	
Teal	"#008080"	
Aqua	"#00FFFF"	

### Insertion d'une image dans votre document

Vous pouvez, en HTML, inclure des images dans un document HTML. Deux formats d'images sont acceptés : le format ['JPEG'](#) et le format ['GIF'](#) (y compris ce que l'on nomme le format 'GIF animés'). Pour ce faire, il vous suffit d'utiliser le tag `<IMG>` : il permet d'introduire une image dans le document. Voici la liste des paramètres que ce tag accepte.

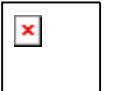
- Src : ce paramètre permet de référencer l'image qui doit apparaître

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

- Width : permet de forcer la largeur de l'image. Si l'image est trop large ou trop étroite, elle sera redimensionnée. Vous remarquerez qu'en l'absence du paramètre fixant la hauteur de l'image, cette dernière est redimensionnée de façon à conserver sa proportionnalité.

`<IMG Src="smile.gif" Width=40>` 

- Height : de même, ce paramètre fixe la hauteur de l'image (si aucune information est apportée sur la largeur de celle-ci, le redimensionnement sera proportionnel).

`<IMG Src="smile.gif" Width=50 Height=50>` 

- Align : ce paramètre permet de contrôler le positionnement de l'image. Si vous l'affectez avec la

## INTERNET

## LES BASES DU LANGAGE HTML

valeur "right", l'image sera placée à droite de la fenêtre du navigateur. De même la valeur "left" la placera à gauche et la valeur "center" centrera votre image.

```
<IMG SRC="smile.gif" ALIGN="right">
```



- 
- Alt : ce paramètre permet de définir une info-bulle pour l'image considérée. Une info-bulle est un petit texte apparaissant lorsque vous laissez durant un bref instant votre souris immobile sur l'image. Vous pouvez tester cela sur l'image suivante.

```
<IMG Alt="Une petite image" Src="smile.gif">
```



- 
- Border : ce paramètre permet de fixer la taille de la bordure de l'image. Attention : cette bordure ne s'affiche que si un lien hypertexte est défini sur cette image.

```
<A Href="mailto:dliard@pop.multimania.com">
  <IMG SRC="smile.gif" Border=3>
</A>
```



- 
- LowSrc : ce paramètre permet de spécifier (de même que SRC) la localisation d'une autre image, qui sera chargée avant celle spécifiée par SRC. Cette image doit, bien entendu, être plus petite (en octets) que la précédente (souvent moins détaillée et bicolore), car elle s'affichera en prévision de la suivante. Ainsi, même pour les machines privées de connexion rapide, une ébauche de la page sera disponible assez rapidement. Il doit cependant être clair que si vous utilisez ce paramètre, il faudra attendre plus de temps pour au final avoir l'image principale. En conséquence, et comme les connexions Internet sont de plus en plus rapides, ce paramètre est très peu employé.
- Usemap : ce dernier paramètre est utilisé pour associer à une image, une carte de zones réactives (par défaut, aucune). C'est ce que je vous propose de voir maintenant.

### Définition de carte de zones réactives

Il existe de plus une possibilité intéressante : on peut utiliser ce que l'on nomme des images réactives. Cela consiste à définir des zones de l'image qui vont pouvoir être activées par click souris, pour atteindre un nouveau document.

Vous pouvez donc découper votre image en plusieurs parties. Pour ce faire, vous pouvez spécifier trois types de zones géométriques (des rectangles, des cercles et de polygones). Ce découpage est appelé une carte (ou map en anglais). Nous avons déjà vu, plus haut, que nous pouvons associer une carte à une image grâce au paramètre Usemap du tag IMG. Le couple de tags <MAP> ... </MAP> permet de définir une carte. Ce couple peut contenir un nombre quelconque de tags <AREA>. Chacun d'eux définit une zone de l'image.

- <MAP> ... </MAP> : ce couple de tags est donc utilisé pour introduire une carte, qui sera utilisée par une image.
  - NAME : ce paramètre fixe le nom de la carte. Ce nom sera utilisé pour réaliser l'association entre la carte et l'image.
- <AREA> : ce tag sert à décrire une des zones de la carte. Voici la liste des paramètres qu'il admet.
  - SHAPE : sert à spécifier la nature de la zone décrite. Trois valeurs sont possibles : "rect" (pour un rectangle), "circle" (pour un cercle) ou "poly" (pour un polygone).
  - HREF : ce paramètre détermine l'[URL](#) appelée lorsque l'utilisateur cliquera sur la zone que vous êtes en train de décrire.
  - COORDS : ce dernier paramètre fixe les coordonnées de la zone décrite. Ces coordonnées doivent être données sous la forme d'une liste d'entiers séparés par des virgules. Dans le cas d'une zone rectangulaire, on spécifie les coordonnées de deux points diamétralement opposés. Dans le cas d'une sphère, on donne son centre et un point de la circonférence. Enfin, pour un polygone, on énumère les coordonnées de tous les sommets, deux à deux consécutifs. Il doit bien être clair que l'utilisation d'un générateur de code HTML vous simplifiera grandement la vie pour ce qui est de la saisie des coordonnées (imaginez les départements de la carte de France).

Voici un petit exemple : on va couper notre image de lettre en deux parties (une partie gauche et un



partie droite) en utilisant une zone rectangulaire. Cliquez sur la partie (gauche ou droite) de la lettre désirée et vous verrez un message s'afficher (cette affichage est réalisé par l'intermédiaire de [Javascript](#) : [pour de plus amples informations, consultez le cours sur ce langage](#)).

```
<MAP NAME="MyMap" >
  <AREA Shape="rect" Href="javascript:gauche();" Coords="0,0,35,36">
  <AREA Shape="rect" HREF="javascript:droite();" Coords="36,0,65,36">
</MAP>
```

```
<IMG Src="smile.gif" UseMap="#MyMap" >
```



Nous allons dans ce chapitre étudier le tag HTML <A>. Nous allons voir qu'il permet de définir deux types d'éléments : les liens hypertextes et les étiquettes.

Un lien hypertexte une zone d'un document HTML sensible aux clicks souris. En cliquant le lien (en en l'activant par le clavier) vous être rediriger vers une autre partie du document, voir vers un autre document (dans votre site, ou dans un autre). Pour réaliser la re-directionil est nécessaire de fournir une [URL \( Uniform Ressource Locator\)](#). Cette dernière précise exactement la localisation du document à afficher.

Une étiquette correspond à une partie du document que l'on peut atteindre par le biais d'un lien hypertexte. Au niveau de l'URL, une étiquette se note en rajoutant, à la suite de l'URL, un caractère # et le nom de l'étiquette.

#### Définition de liens hypertextes

Pour définir un lien hypertexte, il suffit de rajouter au tag <A> le paramètre HREF qui reçoit comme valeur une [URL](#). Lorsque le lecteur activera le lien, il sera redirigé vers l'[URL](#) désirée.

```
<A Href="http://dliard.multimania.com">
  Cliquez
</A>
```

[Cliquez](#)

On peut aussi utiliser un second paramètre (target) pour spécifier [l'éventuelle fenêtre](#) dans laquelle doit s'ouvrir le lien : par défaut, le lien s'ouvre dans la fenêtre qui contient le lien hypertexte.

```
<A Href="http://dliard.multimania.com" Target="fenetre_menu">
  Cliquez
</A>
```

#### Définition d'une étiquette

Comme nous l'avons dit précédemment, il est aussi possible de définir une étiquette. Pour ce faire, il faut fournir, au tag <A>, le paramètre NAME : ce dernier reçoit le nom de l'étiquette. Une fois l'étiquette définie, vous pourrez créer des liens hypertextes permettant d'accéder à cette étiquette. A titre d'exemple, j'ai placé deux étiquettes dans ce document : l'exemple de code suivant vous montre comment je les ai défini, et il vous montre aussi comment définir deux liens hypertextes sur ces deux étiquettes.

```
<A Name="LIEN"><H2>Définition de liens hypertextes</H2></A>
<!-- ... Suite ... -->
<A Name="ETIQUETTE"><H2>Définition d'une étiquette</H2></A>
<!-- ... Définition des liens hypertextes ... -->
<A Href="liens.html#LIEN">Comment définir un lien hypertexte</A> -
<A Href="liens.html#ETIQUETTE">Comment définir une étiquette</A>
```

[Comment définir un lien hypertexte](#) - [Comment définir une étiquette](#)

Nous allons voir dans ce chapitre qu'il est bien entendu possible de définir des tableaux en HTML. Comme nous allons le voir, à partir de l'exemple suivant, en HTML, un tableau est constitué de lignes. Dans chaque ligne, on peut y placer un certain nombre de cellules. Le nombre de cellule pour chaque ligne est égale au nombre de cellule de la ligne qui en comporte le plus ! Si pour une ligne donnée, toutes les cellules ne sont pas définies, elle existerons mais leur contenu sera vide. Le tableau suivant contient trois lignes comportant chacune trois cellules. Ce tableau est suivant du code HTML équivalent.

Elément 1	Valeur 1	Etat 1
Elément 2	Valeur 2	Etat 2
Elément 3	Valeur 3	Etat 3
Ceci est le titre du tableau !		

```
<TABLE BORDER=3 WIDTH="50%">
  <TR>
    <TD WIDTH="33%">Elément 1 </TD>
    <TD WIDTH="33%">Valeur 1 </TD>
    <TD WIDTH="33%">Etat 1 </TD>
  </TR>
  <TR>
    <TD>Elément 2 </TD>
    <TD>Valeur 2 </TD>
    <TD>Etat 2 </TD>
  </TR>
  <TR>
    <TD>Elément 3 </TD>
    <TD>Valeur 3 </TD>
    <TD>Etat 3 </TD>
  </TR>
  <CAPTION Align="bottom">Ceci est le titre du tableau ! </CAPTION>
</TABLE>
```

Nous allons maintenant faire une petite étude de chacun des tags utilisable dans la définition d'un tableau. Pour chaque tag, vous trouverez, si nécessaire, un description de chacun des paramètres supportés par le tag considéré.

- `<TABLE> ... </TABLE>` : comme vous l'avez certainement remarqué, ce couple de tags vous permet d'introduire le tableau dans le document. La tag `<TABLE>` accepte un certain nombre de paramètres, dont voici la description des principaux.
  - `Border` : ce paramètre permet de définir l'épaisseur de la bordure du tableau, ce en nombre de [pixels](#).
  - `Width` : on utilise ce paramètre pour déterminer la largeur du tableau, soit en nombre de pixels soit en pourcentage de la zone utilisable.

```
<TABLE WIDTH="500">
  <!-- Un tableau de 500 pixels -->
  ...
</TABLE>
```

```
<TABLE WIDTH="80%">
  <!-- Un tableau occupant 80% de l'espace disponible -->
  ...
</TABLE>
```

- `BgColor` : ce paramètre permet de définir [une couleur de fond](#) (par défaut) pour les cellules du tableau. On peut aussi définir une couleur de fond sur une ligne et sur chaque cellule. C'est la couleur définie le plus localement qui l'emporte.
- `CellSpacing` : ce paramètre permet de déterminer le nombre de pixels qui vont séparer les cellules du tableaux.
- `CellPadding` : ce dernier sert à fixer la taille (en pixels) de la zone séparant le contenu

d'une cellule d'avec son cadre. En quelque sorte, une marge interne.

- `<CAPTION> ... </CAPTION>` : ces deux tags permettent de donner un titre au tableau en cours de définition. Le tag `<CAPTION>` admet un paramètre *Align* qui peut prendre comme valeurs soit "top" (par défaut) soit "bottom", pour que le titre apparaisse respectivement en haut ou en bas du tableau. On ne peut trouver ce couple de tags (comme toutes ceux qui suivent) forcément qu'à l'intérieur d'un couple `<TABLE> ... </TABLE>`.
- `<TR> ... </TR>` : ce couple permet de déclarer une nouvelle ligne. Une ligne sera constituée de cellules introduites par les couples de tags `<TD> ... </TD>` et `<TH> ... </TH>`.
  - *BgColor* : permet de spécifier la couleur de fond pour toutes les cellules de la ligne en cours.
- `<TD> ... </TD>` : ce couple permet de définir une cellule (de donnée : Table Data) de la ligne en cours. Comme nous allons le voir, le tag `<TD>` accepte de nombreux paramètres.
  - *Align* : ce paramètre permet de fixer l'alignement du texte dans la cellule. Trois valeurs possibles pour l'alignement horizontal : "left" (par défaut), "right" ou "center". De même, trois valeurs permettent de réaliser l'alignement vertical : "top", "middle" ou "bottom".
  - *ColSpan* : permet d'étendre la cellule en cours sur les cellules voisines de droite. Ce paramètre de type valeur entière détermine le nombre de cellules occupées (y compris celle concernée). Cliquez [ici](#) pour voir un exemple concret.
  - *Width* : de même que pour le tag `<TABLE>`, ce paramètre fixe la largeur de la cellule (si on utilise les pourcentages, ils s'expriment en rapport à la largeur de la table).
  - *BgColor* : spécifie la couleur de fond de la cellule.
  - *Background* : permet de définir un motif de fond pour la cellule (attention uniquement pour les nouveaux navigateurs ; faire les tests qui s'imposent).
- `<TH> ... [</TH>]` : ce dernier couple de tags est sensiblement identique au couple précédent. En effet, il sert à définir le contenu d'une cellule du tableau. La différence réside dans le fait que la cellule ainsi définie sera une cellule d'en-tête (Table Header), et non une cellule de données (Table Data). Une cellule d'en-tête a, par défaut, son texte centré et mis en gras.

Si vous souhaitez voir absolument un joli tableau HTML, alors il vous suffit [d'activer le lien suivant](#) !

Nous allons, dans ce chapitre, voir qu'en HTML, il vous est possible de définir des formulaires. Un formulaire est un ensemble de champs de saisis de données : une fois tout les champs remplis, on peut soumettre les données du formulaire considéré. Lors de la soumission, les données sont envoyées du poste client vers le serveur Web qui vous à délivré le document contenant le formulaire.

### Définition d'un formulaire

Le langage HTML vous propose d'utiliser des formulaires afin de collecter diverses informations qui seront envoyées à un programme CGI (Common Gateway Interface) ou bien, par exemple, à une page [ASP \(Active Server Page\)](#). On utilise à cet effet le couple de tags `<FORM> ... </FORM>`. Ce couple sert à délimiter le début et la fin du formulaire. Voici la liste des paramètres que le tag `<FORM>` peut prendre.

- *ACTION* : fixe le nom du document, ou du programme CGI, qui sera appelé lors de la soumission du formulaire.
- *METHOD* : détermine la méthode de passage des paramètres du formulaire lors de la soumission des données.
  - *GET* : les paramètres (et leur valeur) seront passés dans l'URL du CGI appelé (*ACTION*) de la façon suivante : `nomCGI?param1=val1&param2=val2&...&paramn=valn`.
  - *POST* : les paramètres sont passés sous le même format, mais il ne sont pas concaténés à l'URL. Ils sont au contraire passé dans le corps de la requête HTTP.

Vous êtes en droit de vous poser une question : qu'elle est la meilleur méthode de postage de données pour un formulaire ? En fait, ça dépend de ce que l'on cherche à faire. Cependant, en règle générale, c'est la méthode "Post" qui est la plus souvent choisie. En effet, la méthode "Get" pose différents problèmes. Premièrement elle n'est pas adaptée à la saisie de mot de passe : une fois la page demandée affichée, le mot de passe apparaît en clair dans la barre d'adresse du navigateur. De plus, la taille de l'URL est limitée : il est donc pas toujours possible de passer la totalité des informations via cette méthode.

Mais l'aspect le plus important différenciant ces deux méthodes réside dans le fait que la méthode "Post" impose obligatoirement une re-connexion au serveur, sans tenir compte de la version précédente de la page disponible dans le cache du navigateur. La méthode "Get" peut amener à prendre la version dans le cache plutôt que celle du serveur.

Cela peut nous permettre de comprendre pourquoi tous les moteurs de recherche proposent des

formulaires privilégiant la méthode "Get". En effet, dans ce cas, les choses sont particulières : la base de données change, globalement, peut d'état. Deux mêmes recherches aboutissent donc normalement au même résultat, même à quelques jours d'intervalle. Or une recherche par mots clefs dans un moteur de recherche coûte chère en temps. Si l'on peut donc minimiser l'utilisation d'un tel serveur au profit de données disposées dans les caches (ou même les proxy) cela est mieux.

### Définition des éléments de formulaire

Pour ce qui est de la constitution du formulaire, on utilise le tag `<INPUT>`, qui selon les valeurs de certains paramètres, donnera tel ou tel objet graphique de saisie. Dans tous les cas, vous pouvez leur donner un nom en l'affectant au paramètre **Name**. Voici une brève description des objets disponibles.

- Une ligne de saisie de texte : pour ce faire, on affecte au paramètre **TYPE** la valeur "text". Si vous désirez donner une valeur initiale à la variable, vous pouvez affecter cette valeur au paramètre **VALUE**. Dans tous les cas, ce dernier contiendra la chaîne rentrée. Enfin, le paramètre **SIZE** permet de fixer la taille de la zone de saisie (en nombre de caractères). Voici un petit exemple : vous cliquez à l'intérieur de la zone de saisie pour apporter des modifications.

```
<FORM>
  <INPUT TYPE="text" VALUE="Valeur initiale"
        NAME="nom de l'objet de saisie" SIZE="40">
</FORM>
```

Valeur initiale

- Une zone d'acquisition de mot de passe : pour obtenir un tel objet d'acquisition, il vous suffit d'affecter la valeur "password" au paramètre **Type**. Vous pouvez de plus, fournir un mot de passe par défaut en l'affectant à la variable **Value**.

```
<FORM>
  <INPUT TYPE="password" VALUE="Valeur initiale"
        NAME="nom de l'objet de saisie" SIZE="40">
</FORM>
```

\*\*\*\*\*

- Un bouton de soumission du formulaire : il suffit d'affecter la valeur "submit" au paramètre **TYPE**, et le texte devant apparaître dans le bouton au paramètre **VALUE**. Le paramètre **WIDTH** permet de fixer la taille du bouton en octets.

```
<FORM>
  <INPUT TYPE="submit"
        VALUE="Appuyez ici pour valider le formulaire">
</FORM>
```

Appuyez ici pour valider le formulaire

- Un bouton simple (possibilité d'utilisation avec [Javascript](#)) : pour le créer, il faut juste affecter la valeur "button" au paramètre **TYPE**, et le texte devant apparaître dans le bouton au paramètre **VALUE**. La différence essentielle avec le bouton précédent, c'est que dans le cas présent, aucune soumission du formulaire n'est effectuée (pas d'appel CGI). Le paramètre **WIDTH** permet de fixer la taille du bouton en octets.

```
<FORM>
  <INPUT TYPE="button"
        VALUE="Appuyez ici, c'est un bouton">
</FORM>
```

- Un bouton de ré-initialisation du formulaire : il suffit pour cela de fixer la valeur "reset" au paramètre **TYPE** et, de même que précédemment, le texte devant apparaître dans le bouton au

paramètre **VALUE**. Le paramètre **WIDTH** permet de fixer la taille du bouton en octets.

```
<FORM>
  <INPUT TYPE="text" VALUE="valeur initiale">
  <INPUT TYPE="reset" VALUE="Reset">
</FORM>
```

valeur initiale	Reset
-----------------	-------

- Pour passer une variable non directement visible dans le formulaire : pour ce faire, il suffit d'affecter la valeur "hidden" au paramètre **TYPE**, le nom de la variable au paramètre **NAME** et enfin, la valeur de la variable au paramètre **VALUE**.
- Pour obtenir une case à cocher : c'est simple, il suffit d'affecter la valeur "checkbox" au paramètre **TYPE** et un nom au paramètre **NAME** (histoire de pouvoir l'utiliser dans le CGI). Si vous désirez qu'il soit coché à l'état initial, ajoutez le paramètre **CHECKED**.

```
<FORM>
  Un bouton à cocher <INPUT TYPE="checkbox">
</FORM>
```

Un bouton à cocher

- Pour avoir une batterie bouton à choix unique : affectez la valeur "radio" au paramètre **TYPE** de toutes les tags **<INPUT>** réservés à cet effet. De même, tous les paramètres **NAME** devront avoir le même nom. Un seul des tags **<INPUT>** pourra contenir un paramètre **CHECKED**.

```
<FORM>
  Bouton 1 <INPUT TYPE="radio" CHECKED NAME="bouton" VALUE="B1"><BR>
  Bouton 2 <INPUT TYPE="radio" NAME="bouton" VALUE="B2"> <BR>
</FORM>
```

Bouton

1



Bouton 2

- Une boîte de sélection de fichier : il suffit, pour obtenir cet objet, d'affecter la valeur "file" au paramètre **Type**. Ce type de composant permet de mettre en oeuvre ce que l'on appelle l'**uploading**. Cela consiste à télécharger un fichier du poste client (du navigateur) vers le serveur HTTP. Attention : pour que l'upload puisse fonctionner, il faut impérativement rajouter l'attribut **ENCTYPE="multipart/form-data"** au tag **<FORM>**. Il faut de plus qu'un bout de code, sur le serveur Web, soit présente pour pouvoir réceptionner le fichier : cela peut se faire via de nombreux environnements ([ASP](#), [ASP.NET](#), ...).

```
<FORM Enctype="multipart/form-data" Method="POST" Action="Upload.asp">
  <INPUT TYPE="file" Name="file"> <BR/>
  <INPUT Type="Submit" Value="Télécharger">
</FORM>
```

Télécharger
-------------

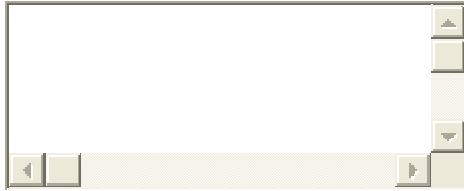
### Autres éléments de saisie de formulaire

Il reste encore deux objets de saisie, mais ils ne sont pas introduits grâce au tag que nous venons de voir. Le premier est une zone de saisie de texte, comme **<INPUT TYPE="text">**, mais sur plusieurs lignes. On utilise à cet effet le couple de tags **<TEXTAREA> ... </TEXTAREA>**. Voici les paramètres que la marque **<TEXTAREA>** peut prendre :

- Name : pour fixer le nom de l'objet de saisie.

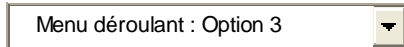
- Cols : fixe le nombre de colonnes de la zone de saisie.
- Rows : fixe le nombre de lignes de la zone de saisie.

```
<FORM>
  <TEXTAREA NAME="Zone" COLS="40" ROWS="5">
</TEXTAREA>
</FORM>
```



Le dernier objet, utilisé par les formulaires, est le menu déroulant. On l'introduit par le couple de marques **<SELECT> ... </SELECT>**. On peut affecter l'unique paramètre **NAME** de la marque **<SELECT>** avec le nom de la variable qui va recevoir le résultat. Ensuite, on définit les choix du menu déroulant avec la marque **<OPTION>**. Vous pouvez donner un choix par défaut en y ajoutant le paramètre **SELECTED**. Voici un exemple d'utilisation.\*

```
<FORM>
  <SELECT NAME="Menu">
    <OPTION> Menu déroulant : Option 1
    <OPTION> Menu déroulant : Option 2
    <OPTION SELECTED> Menu déroulant : Option 3
  </SELECT>
</FORM>
```



Pour rendre vos documents encore plus conviviaux, vous pouvez diviser la fenêtre du navigateur WEB en plusieurs sous-fenêtres. Pour ce faire, on utilise le couple de tags **<FRAMESET> ... </FRAMESET>** pour diviser une fenêtre en plusieurs autres, et le tag **<FRAME>** pour choisir le document HTML qui sera visualisé dans une sous-fenêtre. Voici quelques explications supplémentaires.

- **<FRAMESET> ... </FRAMESET>** : ce couple de tags permet de définir les sous-fenêtres d'une autre.
  - Rows : ce paramètre détermine un ensemble de fenêtres, positionnées les unes sous les autres. La valeur de ce paramètre doit être une liste de valeurs correspondantes aux hauteurs, en pixels, des différentes fenêtres. Une étoile à la place d'une valeur indique que la fenêtre considérée doit prendre toute la hauteur de la fenêtre mère qu'il sera possible de prendre. Voici un exemple de valeur : `ROWS="80,* ,70"`, ce qui indique que la première fenêtre prendra 80 pixels de haut, celle d'en bas 70 pixels et celle du milieu prendra tout ce qui reste. Vous pouvez aussi définir les hauteurs comme proportions de la hauteur de la fenêtre du navigateur (ex `ROWS="33%,* ,33%"`).
  - Cols : idem, mais définit des fenêtres qui seront les unes à côté des autres.
  - Border : définit la taille de la séparation entre deux fenêtres.

Une fois la place allouée à la fenêtre, il faut spécifier ses caractéristiques. Pour cela le couple de tags **<FRAMESET> ... </FRAMESET>**, attend autant de définition de fenêtres (**<FRAME>**) qu'il y a de valeur dans le **ROWS** ou dans le **COLS**. Notez aussi qu'une définition de fenêtre peut être une nouvelle division.

- **<FRAME>** : ce tag sert à associer un document HTML à une sous-fenêtre. Voici la liste de quelques paramètres autorisés.
  - Name : ce paramètre permet de nommer la fenêtre.
  - Src : celui-ci sélectionne le document HTML qui va y être associé.
  - Scrolling (= "yes" = "auto" = "no") : ce paramètre, pouvant prendre trois valeurs, permet de contrôler les barres de scrolling de la fenêtre. Vous pouvez les demander en permanence

## INTERNET

## LES BASES DU LANGAGE HTML

("yes"), toujours les supprimer ("no") ou bien les afficher que si nécessaire ("auto" qui est la valeur par défaut).

- o NoResize : ce paramètre, qui ne prend pas de valeur, interdit le re-dimensionnement de la fenêtre.

```
<HTML>
<HEAD>
  <TITLE>Essai de multi-fenatrage</TITLE>
</HEAD>
<FRAMESET COLS="100,*,100">
  <FRAME NAME="frame1"
    SRC="http://adresse.quelconque"
    SCROLLING="yes">
  <FRAMESET ROWS="100,*" BORDER=1>
    <FRAME NAME="frame2"
      SRC="http://nom.de.fichier" NORESIZE>
    <FRAME NAME="frame3"
      SRC="http://nom.image">
  </FRAMESET>
  <FRAME NAME="frame4"
    SRC="http://une.autre.adresse"
    SCROLLING="no">
</FRAMESET>
</HTML>
```

Une dernière chose à préciser : si vous faites des essais de cadres, un simple rechargement de la page ne suffira pas toujours pour que vous puissiez voir les modifications (les paramètres d'utilisation du cache, pour le navigateur, feront que seuls les codes des pages des fenêtres seront rechargés, et pas celui de la définition des fenêtres). Dans ce cas, videz les caches (voir supprimez les pour la durée des tests) et recharger le code ensuite.

Les fichiers Index.dat sont des fichiers qui renferment un certain nombre d'informations concernant les adresses des sites que vous avez visités. Les adresses URL, tapées ou accédées via des clics sont stockées dans l'index.dat. Ces fichiers ne sont pas effacés par le nettoyage des fichiers temporaires d'Internet Explorer ou des cookies.

### INDEX DAT SUITE

Le programme [Index.dat Suite 2.9.0 Full](#) (2.40 Mo - freeware) permet non seulement de consulter le contenu du fichier index.dat, mais aussi de le vider. Avec ce logiciel, vous pourrez ainsi voir les premières URL consultées sur votre ordi, depuis votre toute première connexion internet. [Voir l'illustration](#).

### SYSTEM SECURITY SUITE

2ème logiciel : En plus de permettre de réduire le volume des fichiers Index.dat, [System Security Suite v 1.04](#) supprime les traces de navigation sur internet et les fichiers inutiles: Le cache et l'historique Internet Explorer, les mots de passe auto-complétés, les formulaires auto-complétés, les listes de documents récemment utilisés, les recherches dans l'historique, les fichiers récemment imprimés, les listes de windows media player, les adresses URL tapées dans la barre d'adresse, le dossier temporaire de Windows, la corbeille. [Voir l'illustration](#). [Téléchargez System Security Suite v 1.04](#) - 2.41Mo - Freeware.

### DISC CLEANER

## INTERNET

## LES BASES DU LANGAGE HTML

A l'exception index.dat, Disk Cleaner vous permet d'effacer pratiquement tout ce qui encombrement votre PC inutilement:

System Temporary Files, Temporary, Internet Files, Internet Cookies, Internet History, Firefox Cache, Opera Cache, Opera History, Recycle Bin, Lavasoft Adaware, Recent Documents, URLs in IE Address Bar, Media Player Recent URLs, Zone Alarm Log File, etc... [Téléchargez Disk Cleaner](#) - 362 Ko - Freeware.

**IE PRIVACY CLEANER**

Mais le meilleur reste le freeware [IE Privacy Keeper](#) :

Efface l'historique des navigateurs Internet Explorer et Mozilla Firefox , Nettoie des fichiers index.dat , les documents ouverts ou effacés, photos et films visionnés, applications lancées... IE Privacy Keeper possède une fonction d'effacement sécurisé et fera échouer toutes tentatives de restauration des éléments supprimés. Téléchargez [IE Privacy Keeper](#) Version: 2.7.3 - Taille: 1.2 Mo - Freeware - Français - [Site officiel](#) .