

DHTML :

Donnez vie à vos pages

Animations, interaction, mises à jour facilitées... À la croisée entre le HTML classique, les feuilles de style et le Javascript, la technologie DHTML permet de créer des pages animées dont l'interactivité est indépendante du serveur et des délais de transmission. Le DHTML est peut-être l'avenir du HTML.

Christophe Lagane

Jusqu'à récemment, pour créer un site web animé, il n'y avait pas trente-six solutions : l'installation de gif animés et de plug-ins ou encore d'applets Java pas toujours facile à mettre en place. L'exploitation de plug-ins impose la présence du logiciel « lecteur » sur la machine de l'utilisateur, ce qui n'est pas toujours le cas. Quant aux gif animés, si les fichiers sont simples à créer et à mettre en place, leur accumulation alourdit considérablement le poids de la page et ralentit d'autant son chargement. De plus, les animations, limitées par leur répétition en boucle, n'offrent aucune interactivité à l'exception de la mise en place d'un lien. Aujourd'hui, en digne successeur évolué du HTML, le DHTML (Dynamic HTML en anglais) combine animation et interactivité sans pour autant créer des pages trop lourdes. Sa conception reprend la « philosophie » de Java, à savoir qu'il intègre dans la page destinée au client



<http://www.htmlguru.com>, un bon exemple des possibilités en HTML

(l'ordinateur, ou plus précisément le navigateur de l'utilisateur qui visite un site) tous les éléments animés et interactifs qui s'exécutent en local sur la machine de l'internaute. Cette solution offre des poids de fichiers acceptables et la possibilité d'agir sur une page-écran sans avoir à la recharger, c'est-à-dire à faire appel au serveur. Le DHTML enrichit donc l'interface du web tout en économisant de la bande passante d'un réseau déjà bien surchargé. Ce n'est pas un nouveau langage mais une amélioration de la façon de programmer le HTML.

COMMENT FAIRE ?

On fait du DHTML en utilisant les feuilles de styles (CSS pour Cascading Style Sheets) et en y intégrant du script (JavaScript de Netscape, VBScript de Microsoft ou n'importe quel script interprétable par le navigateur). On s'intéressera ici plus particulièrement à JavaScript reconnu par la

plupart des navigateurs et au CSS analysé par le W3C². On ne saura donc trop vous conseiller de vous intéresser à ce langage pour éditer vos propres pages, surtout si vous aimez les animations. De bonnes notions sur les feuilles de style et les scripts vous seront nécessaires. Répétons-le, le DHTML est autonome. Il ne nécessite donc aucun plug-in et autres contrôles de type ActiveX pour être lu tout en sachant gérer les modules extérieurs. Il s'exécute, en théorie, quelle que soit la plate-forme d'accueil (Windows, Mac, Unix, Linux, etc.). Il n'est pas supporté (ou si peu) par les versions 3 et antérieures des navigateurs Netscape et Microsoft qui pourront cependant afficher le contenu textuel de la page. Problème : les deux éditeurs vedettes ne gèrent pas le DHTML de la même façon. Cette incompatibilité complique le travail des auteurs amenés à jouer sur des variables pour permettre à la page

d'être lue, quel que soit le navigateur utilisé (ou à limiter l'accès à un seul des deux navigateurs). C'est Netscape qui, avec son tag LAYER, a introduit la notion de HTML dynamique. Mais ni Microsoft (rien d'étonnant), ni le W3C n'ont avalisé cette propriété au profit des CSS que allons d'abord étudier. Il est cependant recommandé de connaître les bases du HTML pour comprendre les exemples qui suivent. Si ce n'est pas le cas, contentez-vous de recopier les codes proposés à l'aide d'un éditeur de texte quelconque, sauvez votre fichier au format txt mais avec l'extension.htm et ouvrez votre fichier dans Internet Explorer.

LES FEUILLES DE STYLE EN CASCADE

Les feuilles de styles (cascading style sheets) sont un des éléments du DHTML. Comparables aux feuilles de styles des logiciels de traitement de texte et de PAO, elles permettent de définir une mise en page constante et reproductible sur plusieurs documents et en facilitent la gestion puisqu'il suffit d'affecter un nouvel attribut à un style pour que tous les éléments (textes ou autres) qui en dépendent soient modifiés instantanément. Le HTML 4.0 n'est pas lié à un type particulier de feuilles de style ce qui permet d'utiliser n'importe quel langage reconnu par le navigateur. Cependant, on se limitera ici à étudier la version 1 du CSS (ou CSS1) validée par le W3C.

PRINCIPES

Les feuilles de style servent à créer des « objets » (texte, image...) et s'appliquent à différents niveaux : au document entier ; à tous les éléments d'un type particulier (les éléments marqués de la balise <P>...</P> par exemple); ou à certains d'entre eux repérés par l'attribut CLASS ou ID. Ainsi, l'exemple suivant H2 {color: blue; font-family: Arial} indique que tous les textes de niveau 2 (H2) seront affichés en bleu (color: blue) avec la police Arial (font-family: Arial). H2 est un sélecteur qui définit la règle encadrée par des accolades. Elle se compose de couples pro-

priété : valeur^s séparés par des points-virgules. Une même règle peut s'appliquer à plusieurs sélecteurs qu'il suffit de séparer par une virgule : H1, H2 {color: blue} indique que les éléments de niveau 1 et 2 s'afficheront en bleu.

NOUVELLES BALISES

L'introduction d'une feuille de styles dans le document HTML se fait par la balise <STYLE>...</STYLE> à laquelle on attribue... des attributs : Type est quasiment le seul utilisé actuellement. Il s'agit de mots clés MIME¹ qui précisent le langage utilisé par la feuille de styles. Pour celles qui appartiennent au type CSS on écrit : <STYLE type="text/css">. On peut aussi faire appel à une ou plusieurs feuilles de style externes dont l'extension du fichier sera ".css" en intégrant le marqueur LINK dans le conteneur <HEAD>. HREF renvoie au nom

du fichier et REL indique s'il s'agit d'une feuille de style permanente (stylesheet) ou provisoire (alternate). En prenant fds1.css comme exemple de nom de fichier de feuille de style extérieure, on écrira donc <LINK AHREF="fds1.css" REL="stylesheet" TYPE="text/css"> autant de fois qu'il y a de feuilles à introduire.

Pour modifier un style localement, on utilisera le tag STYLE pour un certain nombre d'éléments HTML (P, IMG...).

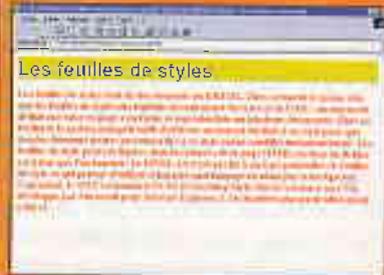
Dans l'exemple suivant



INTERNET EXPLORER VS NETSCAPE

Tous les navigateurs ne savent pas forcément interpréter les fonctions du style et du script. La plupart de ceux-là afficheront le texte brut, sans respect du format indiqué. Mais pour être sûr que le navigateur n'interprète pas mal ce qu'il ne comprend pas, il faut mettre la règle du style entre les balises de commentaires : <!-- -->

Dans l'exemple suivant, Netscape Communicator 4 ne reconnaît pas le type css et n'applique donc pas les attributs tout en affichant le contenu du document dont voici le code :



```

<HTML-HEAD>
<TITLE>Les feuilles de styles</TITLE>
<H2 color="blue" font-family="Arial" style="font-size: 1.2em; text-align: center; text-decoration: underline; text-decoration-color: red; text-decoration-line: underline; text-decoration-style: solid;">
ET TU T'EN SERAS</H2>
</HTML-HEAD>
<BODY>
<H1>Les feuilles de styles</H1>
</BODY>

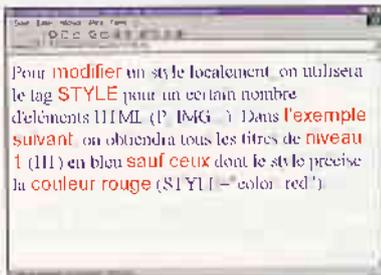
```

Les feuilles de styles sont un des éléments du DHTML... On étudiera plus particulièrement celui-ci.

Avec le DHTML, les deux principaux navigateurs du marché n'utilisent pas la même syntaxe pour accéder aux propriétés de positionnement d'un objet. Ainsi avec un objet ayant pour nom : creanum (<DIV ID="creanum">), IE utilisera la syntaxe creanum.style.attribut tandis que Navigator document creanum.attribut pour accéder à la propriété attribut. De même, la valeur de la propriété visibility sera hidden/visible sous IE alors que Netscape utilise hide/show. Écarter les différentes vues syntaxiques entre Microsoft et Netscape n'est pas le but de cet encadré. Il est difficile de prédire l'avenir mais on peut espérer que les prochaines versions des navigateurs, notamment Netscape qui donne accès gracieusement au code source de Navigator, s'en tiendront aux recommandations du W3C et permettront une compatibilité qui simplifiera le travail des auteurs.

```
<HTML><BODY>
<STYLE type="text/css">
H1 {color: blue}
</STYLE>
<H1>LES FEUILLES DE STYLES C'EST
PRATIQUE</H1>
<H1 STYLE="color: red">...ET C'EST
BIEN UTILE</H1>
</BODY></HTML>
```

on obtiendra tous les titres de niveau 1 (H1) en bleu sauf ceux dont le style précise la couleur rouge (STYLE="color: red"). Si l'on souhaite limiter la modification à des mots, il est préférable d'employer l'élément SPAN avec l'attribut CLASS. Comme le montre l'écran, c'est un moyen très efficace de mettre en valeur du texte :

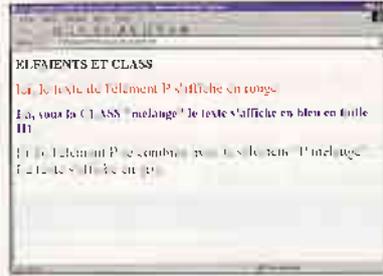


```
<HTML>
<STYLE type="text/css">
P {color: blue; font-size: 39}
.rouge {color: red; font-family: Arial;
font-size: 41}
</STYLE>
<BODY>
<P><SPAN class=rouge>modifier</SPAN> un style localement, on
utilisera le tag <SPAN
class=rouge>STYLE</SPAN> pour un
certain nombre d'éléments HTML (P,
IMG ...) Dans <SPAN
class=rouge>l'exemple
suivant</SPAN>, on obtiendra tous
les titres de <SPAN
class=rouge>niveau 1</SPAN> (H1)
en bleu <SPAN class=rouge>sauf
ceux</SPAN> dont le style précise la
<SPAN class=rouge>couleur
rouge</SPAN> (STYLE="color: red").
</P>
</BODY></HTML>
```

Comme l'indique le code, il suffit de définir le style .rouge défini par son point à l'attribut CLASS et de placer cet attribut avec l'élément SPAN sur le texte à faire ressortir. Il existe un autre attribut, ID, dont les styles sont définis par un « # » au lieu du « . ». La distinction entre CLASS et ID n'est pas frappante mais dans le cas où la modification du style ne porte qu'à un seul endroit du document, il est conseillé d'utiliser ID. On peut

aussi « combiner » les éléments et les attributs.

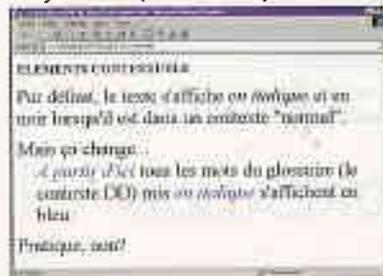
Dans l'exemple suivant :



```
<HTML>
<STYLE type="text/css">
P {font-size: 31; color: red}
.melange {color: blue}
P.melange {color: gray}
</STYLE>
<BODY>
<H1>ELEMENTS ET ATTRIBUTS</H1>
<P>Ici, le texte de l'élément P s'affiche
en rouge <P>
<H1 CLASS="melange">Là, sous l'attribut
"melange" le texte s'affiche en
bleu en taille H1 </H1>
<P CLASS="melange">Et là, l'élément
P se combine avec le sélecteur
"P.melange". Le texte s'affiche en
gris.</P>
</BODY></HTML>
```

on constate qu'en combinant l'attribut « melange » à l'élément H1, ce dernier s'affiche en bleu (défini par la CLASS) mais garde sa taille de niveau 1 (défini par l'élément H1 sans règle). De plus, un attribut P.melange permet d'afficher en gris un texte formaté en bleu par sa CLASS. Bien sûr, ces règles ne s'appliquent pas qu'aux couleurs mais à toutes les propriétés des éléments (taille, police...).

Autre propriété des feuilles de style, les sélecteurs contextuels ou la possibilité d'attribuer un style à un élément de formatage physique (B, I, TT...). Supposons que l'on veuille afficher en bleu le texte en italique (<I>) qui se trouve dans une liste de glossaire (<DD>). On définira alors le style DD I {color: blue}.



```
<HTML><HEAD>
<FONT size="49">
<STYLE type="text/css">
DD I {color: blue}
</STYLE>
</HEAD>
<BODY>
<H1>ELEMENTS CONTEXTUELS</H1>
Par défaut, le texte s'affiche <I>en
italique</I> et en noir lorsqu'il est
dans un contexte "normal".
<DL>
<DT>Mais ça change ..
<DD><I>A partir d'ici</I> tous les
mots du glossaire (le contexte DD)
mis <I>en italique</I> s'affichent en
bleu.
</DL>
Pratique, non?
</FONT></BODY></HTML>
```

Certaines propriétés sont transmissibles par « héritage », c'est-à-dire qu'elles accumulent les valeurs précédemment accordées. Exemple, la CLASS « BIG » définie par BIG {font-size: 150%} permet d'afficher des lettres de plus en plus grandes selon le code suivant :



```
<HTML><HEAD>
<FONT size="49">
<STYLE type="text/css">
BIG {font-size: 150%}
</HEAD></STYLE>
G<SPAN class="BIG">R<SPAN
class="BIG">A<SPAN
class="BIG">N<SPAN
class="BIG">D</SPAN></SPAN></S
PAN></SPAN>
</FONT></BODY></HTML>
```

Maintenant que nous connaissons les grands principes des feuilles de style, abordons le DHTML.

Dynamic HyperText Markup Language (DHTML)

La programmation du DHTML se rapproche du développement classique d'applications : on construit une interface composée d'objets (titres, textes, images, boutons, etc.) auxquels sont associées des propriétés (polices, couleurs, tailles...) que l'on anime en définissant des événements (clic, affichage, saisie, heure...).

PROPRIÉTÉS ET FONCTIONS

Le DHTML possède des propriétés clés et autant de valeurs dont la syntaxe s'approche de l'anglais courant. Par exemple, la propriété `cursor` et sa valeur `hand` (pointeur main) transforme la flèche du pointeur de la souris en main. Tapez le code suivant et constatez :

```
<HTML><BODY>
<SPAN STYLE="cursor: hand">
Le curseur se transforme en main
quand on me passe dessus.
</SPAN></BODY></HTML>
```

Cette fonction est très pratique pour signaler un lien ou une action sans avoir à souligner ce lien ou cette action. La transformation de la main au survol de l'objet suffit à attirer l'attention de l'utilisateur. Les fonctions `onmouseover` (quand la souris est dessus) et `onmouseout` (quand la souris n'est pas dessus) modifient les propriétés d'un objet au passage de la souris ou plutôt de son pointeur. Composons un code qui colorera le texte « Je deviens tout rouge quand on me passe dessus » en rouge quand la souris passe dessus. Déclarons les propriétés du sélecteur P avec de l'arial en 51 points :

```
<HTML><BODY>
<STYLE text="text/css">
P {font-family: Arial; font-size: 51;}
</STYLE>
```

Appliquons les fonctions `onmouseover` avec un affichage en rouge et en noir pour `onmouseout` :

```
<P onmouseover="style.color='red';"
onmouseout="style.color='black';">
Je deviens tout rouge quand
on me passe dessus.
</P>
</BODY></HTML>
```

Saisissez les quelques lignes précédentes et constatez le résultat. Il existe des centaines de propriétés⁵ que nous ne pouvons développer ici. Nous nous contenterons de quelques exemples.

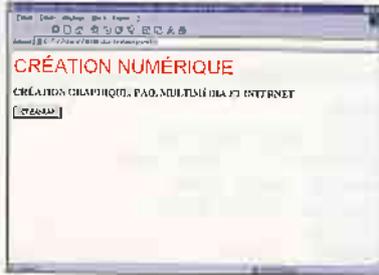
INTERACTIVITÉ

Les propriétés du DHTML permettent également une certaine interactivité. Grâce à la fonction `onclick`, le code suivant transforme le texte « Cliquez moi ! » formaté en Arial rouge de taille 31 en Times bleu 51.

```
<HTML><BODY>
<SPAN style="color: red; cursor: hand;
```

```
font-family: Arial; font-size: 31"
onclick="style.fontSize='51',
style.color='blue',
style.fontFamily='Times'">
Cliquez moi !
</SPAN></BODY></HTML>
```

On peut également faire apparaître du texte en cliquant, par exemple, sur un bouton. Dans l'exemple suivant :



```
<HTML><HEAD>
<SCRIPT Type="text/javascript">
function f(texte)
{titre.innerHTML=texte}
</SCRIPT></HEAD>
<BODY>
<FONT FACE="arial" SIZE="21"
COLOR="red">CRÉATION NUMÉRIQUE</FONT>
<H2 ID=titre>&nbsp;</H2>
<FORM>
<BUTTON TYPE=button
onclick="f('CRÉATION GRAPHIQUE,
PAO, MULTIMÉDIA ET INTERNET')">
CRÉANUM
</FORM>
</BODY></HTML>
```

Le fait de cliquer sur le bouton CRÉANUM rajoute du texte à la page. Cette propriété s'obtient grâce à l'instruction `titre.innerHTML=texte` (respectez bien le « T » majuscule). `innerHTML` est la propriété qui représente le contenu de l'objet titre alors remplacé par le contenu de texte.

CONSTRUCTIONS ET POSITIONNEMENT D'OBJET

Dans l'exemple suivant, nous allons voir comment placer des objets (texte, image...) à un endroit précis de la page et comment les faire se superposer.



```
<HTML><HEAD>
<STYLE type="text/css">
.crea {color: Blue; position: absolute;
top:15px; left: 20px; visibility: visible;
z-index: 1}
.num {color: red; position: absolute;
top: 85px; left: 300px; visibility:
visible; z-index: 2}
.creabas {color: Blue; position: absolute;
top:215px; left: 20px; visibility:
visible; z-index: 4}
.numbas {color: red; position: absolute;
top: 285px; left: 300px; visibility:
visible; z-index: 3}
</STYLE></HEAD>
<BODY>
<SPAN class="crea" style="font-family: arial; font-size: 81pt; font-weight: bold">
CRÉATION
</SPAN>
<SPAN class="num" style="font-family: arial; font-size: 35pt; font-style: italic;">
NUMÉRIQUE
</SPAN>
<SPAN class="creabas" style="font-family: arial; font-size: 81pt; font-weight: bold">
CRÉATION
</SPAN>
<SPAN class="numbas" style="font-family: arial; font-size: 35pt; font-style: italic;">
NUMÉRIQUE
</SPAN>
</BODY></HTML>
```

La propriété « position » va nous permettre de placer l'objet par rapport au haut (`top`) et à la gauche (`left`) de la page avec des valeurs absolues (pour définir la taille et la position en pixel ou cm), relative (pour définir la taille et la position en % par rapport à la taille de la fenêtre) ou statique (l'objet est figé). `Visibility` peut prendre les valeurs `Hidden` ou `Visible` pour cacher ou montrer un objet, `width` et `height` déterminent sa largeur et sa hauteur (non employés dans l'exemple), et `z-index` précise sa priorité d'apparition. Dans la partie supérieure de l'écran, le mot « CRÉATION » (placé à 15 pixels du haut et 20 de la gauche de l'écran) se trouve derrière le mot « NUMÉRIQUE » (placé à 85 pixels du haut et 300 de la gauche de l'écran), grâce à la propriété `z-index` qui indique que l'objet `.num` a une valeur prioritaire (2) sur celle de l'objet `.crea` (1). Inversement, dans la partie inférieure de l'écran, les priorités sont inversées et « CRÉATION » s'affiche par dessus « NUMÉRIQUE ».

ANIMATION

Nous l'avons vu, ce sont les feuilles de style qui permettent de créer des objets munis de propriétés et de positionnement. C'est avec du script, que l'on déclare par `<SCRIPT type="text/javascript">`, que l'on va animer ces objets. C'est généralement le tag `<DIV>` qui permet d'introduire l'objet sur la page que l'on nomme avec le marqueur ID et ses attributs avec CLASS. Nous allons écrire une petite animation qui déplacera le texte « Création Numérique » de la droite vers la gauche de l'écran. Créons l'objet « logo » (qui contiendra le texte « Création Numérique ») de police Arial, couleur rouge, placé à 15 pixels du haut et 200 pixels de la gauche de l'écran, visible et en priorité 1 :

```
<HTML><HEAD><STYLE>
.logo {color: #FF9900; position: absolute;
top:15px; left: 200px; visibility:
visible; z-index: 1;}
</STYLE></HEAD>
```

Déclarons l'objet dans la page sous le marqueur « Pixel »

```
<BODY>
<DIV ID = "Pixel" CLASS = "logo">
Création Numérique
</DIV>
```

Pour animer cet objet, il faut créer un script qui contient une fonction (fonction) « bouge (de, a) » (de 0 à 600) laquelle incrémente de 10 pixels la valeur de la position left. La fonction `setTimeout` est une routine qui fait avancer l'objet à raison d'une fréquence de 20.

```
<BODY>
<DIV ID = "Pixel" CLASS = "logo">
Création Numérique</DIV>
<SCRIPT type="Text/JavaScript">
block = (Pixel style; function bouge(de,
a) { if (de < a) { block left = (de +=
10); setTimeout('bouge(' + de + ',' + a
+ ')', 20); } } bouge(0,600);
</SCRIPT></BODY></HTML>
```

Si l'on souhaite faire descendre le texte « Création numérique », remplacez la propriété `block.left` par `block.top`. Pour faire descendre latéralement le texte, il suffit de déclarer les deux propriétés de `bouge` et leur valeur : `block.top = (de += 10)`; `block.left = (de += 1)`; `setTimeout('bouge...`

Le DHTML permet donc, avec un minimum de connaissances, de donner de la « gueule » à vos pages web tout en facilitant la gestion de votre site. Et la satisfaction d'un beau résultat fera oublier des premiers pas peut-être difficiles. Apprendre le Javascript n'est pas forcément votre objectif et, aujourd'hui, nombre d'éditeurs génèrent très bien le DHTML. Mais le principe du script est quasiment indispensable pour se repérer lorsqu'il faut « mettre les mains dans le cambouis ». Disons que ces connaissances de base vous permettront d'aller copier/coller les animations qui vous intéressent à partir d'autres sites et de pouvoir les adapter à vos besoins. Cependant, n'oubliez pas que l'habit ne fait pas le moine ! Un site doté d'une belle présentation sera toujours plus agréable à consulter qu'une série de paragraphes textuels accessibles par des liens classiques. Mais passé les premiers éblouissements de la « mise en animation », l'internaute n'y restera et n'y reviendra que s'il y trouve ce qu'il cherche. C'est donc à travers le contenu et sa mise à jour que votre site manifestera sa vraie valeur. ■

¹ Microsoft donne la documentation du VBScript : <http://www.microsoft.com/scripting/>. La documentation Javascript de Netscape se consulte à <http://home.netscape.com/eng/mozilla/3.0/handbook/javascript/>.

² La recommandation du W3C (le consortium chargé de statuer sur la norme HTML) sur le CSS1 se consulte à <http://www.w3.org/TR/REC-CSS1>. Page d'entrée au <http://www.w3.org/Style/>.

³ Il existe près d'une centaine de sélecteurs et une cinquantaine de propriétés dans le modèle CSS1.

⁴ Le Multipurpose Internet Mail Extension (MIME) permet d'acheminer, au moyen d'un codage spécifique, des fichiers binaires de type variés sur le réseau.

⁵ Le site de Microsoft (<http://www.microsoft.com/msdn/sdk/inetsdk/asetup/default.htm>) propose le SDK, un document constitué de 362 propriétés et 200 méthodes.

EXEMPLE DE DHTML SUR LE NET

De quoi nous rendons de voir n'est qu'un petit aperçu des possibilités qu'offrent les feuilles de styles combinées au script. Comme il est impossible d'approfondir le sujet dans le cadre de ce magazine, nous vous invitons à naviguer sur le réseau pour voir les capacités de cette technologie. Le site de Microsoft (<http://www.microsoft.com/development/activex/dhtml/>)

fournit (en anglais) conseils, astuces, exemples... Même chose du côté de Netscape (<http://www.netscape.com/eng/mozilla/3.0/handbook/javascript/>), mais en français. Citéweb (<http://www.citeweb.com/>) propose son guide du HTML (y compris feuilles de style et DHTML) en français.

Cette page (<http://www.citeweb.com/animation/>) sur le site de CITEWEB offre une démonstration de l'animation et la superposition des éléments.



Le fond de la page change en fonction des saisons que l'on sélectionne en passant la souris sur l'un des quatre top-modèles : rendez-vous sur le site de Télé 7 jour.



Enfin, www.citeweb.com (à voir) vous aurez les programmes télé du jour mais surtout un aperçu de la fonction Data-binding (non étudiée dans cet article) qui offre une interface web avec une base de données : cliquez sur l'une des chaînes, s'affichent alors instantanément la grille des programmes correspondants, sans avoir à « interroger » le serveur (7) etc. Enfin, All-Link (<http://www.all-link.com/>) est une plateforme de liens qui renvoie sur les principaux sites de documentation en cours sur le DHTML.