

# *Cahiers* **GUT** *enberg*

☞ AFFICHEZ VOS DOCUMENTS L<sup>A</sup>T<sub>E</sub>X SUR LE  
WEB AVEC T<sub>E</sub>X4HT

☞ Fabrice POPINEAU

*Cahiers GUTenberg*, n° 37-38 (2000), p. 5-43.

<[http://cahiers.gutenberg.eu.org/fitem?id=CG\\_2000\\_\\_37-38\\_5\\_0](http://cahiers.gutenberg.eu.org/fitem?id=CG_2000__37-38_5_0)>

© Association GUTenberg, 2000, tous droits réservés.

L'accès aux articles des *Cahiers GUTenberg*

(<http://cahiers.gutenberg.eu.org/>),

implique l'accord avec les conditions générales

d'utilisation (<http://cahiers.gutenberg.eu.org/legal.html>).

Toute utilisation commerciale ou impression systématique

est constitutive d'une infraction pénale. Toute copie ou impression

de ce fichier doit contenir la présente mention de copyright.



---

# Affichez vos documents L<sup>A</sup>T<sub>E</sub>X sur le Web avec T<sub>E</sub>X<sub>4</sub>ht

---

Fabrice Popineau

SUPELEC – Campus de Metz

2, rue E. Belin

F-57070 Metz

email: Fabrice.Popineau@supelec.fr

**Résumé.** Eitan Gurari est l’auteur de T<sub>E</sub>X<sub>4</sub>ht, un outil original permettant la conversion de documents T<sub>E</sub>X et L<sup>A</sup>T<sub>E</sub>X vers HTML et XML. Mon but est de montrer que T<sub>E</sub>X<sub>4</sub>ht est simple d’utilisation, puissant et extensible en effectuant un petit tour de ses possibilités.

**Abstract.** Eitan Gurari is T<sub>E</sub>X<sub>4</sub>ht’s author, a clever tool which allows T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X documents to be translated to HTML and XML. I’d like to show here that T<sub>E</sub>X<sub>4</sub>ht is of simple use, powerful and extensible. Let’s have a look at its features.

## 1. Les convertisseurs T<sub>E</sub>X et L<sup>A</sup>T<sub>E</sub>X vers HTML/XML

### 1.1. Le contexte

La demande pour des convertisseurs T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X vers HTML/XML est très forte. Le milieu scientifique (recherche et édition) est un gros producteur de documents balisés en T<sub>E</sub>X et le web est aujourd’hui l’outil incontournable de communication. On supposera que le lecteur est familier des technologies du web que sont HTML, CSS, XML, MATHML, etc. Si ce n’était pas le cas, la bibliographie regroupe des pointeurs sur ces sujets.

L<sup>A</sup>T<sub>E</sub>X possède un intérêt indéniable : il procure une syntaxe d’entrée simple et générale pour des documents scientifiques<sup>1</sup>. Cela permet d’obtenir un document orienté papier (Postscript, PDF) ou orienté écran (HTML, PDF) à partir d’un seul fichier source. On peut penser qu’il est facile de passer de L<sup>A</sup>T<sub>E</sub>X à HTML : ce sont des langages très structurés et L<sup>A</sup>T<sub>E</sub>X n’est pas loin d’être un langage de balisage au même titre que HTML.

---

1. C’est une des conclusions d’une table ronde à TUG’99.

En fait, les choses ne sont pas si simples pour plusieurs raisons. La première réside dans le fait que HTML est un langage très limité : il ne permet pas d'afficher les symboles, ni les vraies formules mathématiques, ni les graphiques vectoriels<sup>2</sup>. Les balises sont en petit nombre, les extensions non-conformes des navigateurs ont donc proliféré. Tout ceci rend difficile la production de code HTML « universellement » lisible<sup>3</sup>. Une limitation supplémentaire et contradictoire vient du fait que HTML laisse à la discrétion du navigateur le positionnement des objets sur la page (du moins sans les CSS), contrairement à T<sub>E</sub>X qui, en tant que langage typographique, permet de préciser très finement ce positionnement.

La deuxième raison se situe à l'entrée du convertisseur : L<sup>A</sup>T<sub>E</sub>X n'oblige pas le rédacteur à une grande rigueur dans l'utilisation des balises. En fait, on peut allègrement mélanger du code T<sub>E</sub>X dans un document L<sup>A</sup>T<sub>E</sub>X, ou bien utiliser les balises de manière inadaptée *vis-à-vis de L<sup>A</sup>T<sub>E</sub>X* comme par exemple :

```
{\it texte en italique}      % Inadapté !
\textit{texte en italique} % Correct.
{\large gros texte}         % Inadapté et pas d'équivalent !
```

Bien évidemment, détecter automatiquement la structure du document devient alors plus difficile, car le nombre de possibilités augmente et on a rapidement besoin de mettre en place des règles heuristiques qui, par définition, ne sont pas certaines. Ne parlons pas non plus des définitions de macros personnelles : comment les interpréter de manière fiable sans réécrire T<sub>E</sub>X ?

Néanmoins, il y a eu de très nombreuses tentatives pour créer des traducteurs automatiques :

1. des systèmes autonomes, indépendants de T<sub>E</sub>X : HEVEA, IBM TechExplorer, L<sup>A</sup>T<sub>E</sub>X<sub>2</sub>HTML, L<sub>TX</sub>2X, Texi<sub>2</sub>HTML, Tth et bien d'autres,
2. un processeur de fichiers DVI : T<sub>E</sub>X<sub>4</sub>ht,
3. des moteurs T<sub>E</sub>X modifiés (ajout de nouvelles primitives) : MicroPress TeXpider, Oméga.

On se reportera à [8] pour une liste plus complète d'outils.

Steve Mayer a réalisé *T<sub>E</sub>X Converter*, un programme d'interface à plusieurs convertisseurs vers HTML sous Windows. Parmi les pages concernant cet outil [11], on peut trouver une comparaison [10] très intéressante et très instructive de divers traducteurs. Les traducteurs testés utilisent des combinaisons d'images générées et de symboles disponibles dans le navigateur pour rendre les formules mathématiques avec des résultats plus ou moins heureux.

---

2. Certains navigateurs tel Internet Explorer procurent des possibilités dans ce domaine.  
 3. Voir plus loin dans l'article un exemple de problème avec Netscape 4.7.

## 1.2. La structure spécifique de T<sub>E</sub>X<sub>4</sub>ht

La plupart des traducteurs reposent sur un ensemble de règles heuristiques plus ou moins solides pour simuler (ou « deviner ») ce que L<sup>A</sup>T<sub>E</sub>X aurait produit comme sortie. Ensuite, cette sortie simulée est traduite à l'aide des balises HTML pour produire un effet comparable à l'écran. Ces traducteurs sont en général monolithiques :

- LaTeX2HTML consiste en 500 ko de code Perl, il arrive à produire des résultats très convaincants, comme par exemple le site web de TUG'99 (<http://www.tug.org/tug99>),
- TTH, maintenant commercialisé, consiste en un analyseur syntaxique de plus d'un mégaoctet de code C produit à l'aide du générateur d'analyseurs lexicaux lex.

T<sub>E</sub>X<sub>4</sub>ht suit une approche résolument originale. Son auteur, Eitan Gurari l'a présenté pour la première fois à TUG'97 [9]. Ce convertisseur est disponible sur les CD-ROM T<sub>E</sub>X-Live pour les architectures Unix et Windows, et bien évidemment depuis la page de l'auteur [7].

T<sub>E</sub>X<sub>4</sub>ht utilise L<sup>A</sup>T<sub>E</sub>X pour convertir un document L<sup>A</sup>T<sub>E</sub>X vers HTML. On peut dire que T<sub>E</sub>X<sub>4</sub>ht *instrumente* L<sup>A</sup>T<sub>E</sub>X pour produire du HTML. Toutes les instructions L<sup>A</sup>T<sub>E</sub>X pertinentes sont encapsulées et leur appel dérivé pour produire un fichier DVI très « spécial ». En effet, ce fichier DVI contient, sous forme de commandes `\special`, une partie du code HTML souhaité, en fait tout ce qui correspond à la structure du fichier d'entrée. Le texte lui-même sera extrait du fichier DVI généré par L<sup>A</sup>T<sub>E</sub>X, et c'est à ce moment que les problèmes de polices seront résolus au moyen d'un mécanisme de polices hypertextes virtuelles (fichiers `.htf`). Le convertisseur est composé de :

- un ensemble de fichiers d'extension pour L<sup>A</sup>T<sub>E</sub>X : `tex4ht.sty` accompagné des fichiers `*.4ht` ;
- deux petits programmes `tex4ht` et `t4ht` ; le premier sert à extraire la sortie HTML du fichier DVI et à résoudre les références croisées, le deuxième à générer sous forme d'image à inclure les éléments qui ne peuvent pas être traduits en HTML.

On se reportera à la figure 2.2 pour une vue d'ensemble.

Cette technique d'instrumentation de L<sup>A</sup>T<sub>E</sub>X se révèle très puissante, même si elle est complexe à mettre en œuvre : T<sub>E</sub>X<sub>4</sub>ht est à ma connaissance le seul traducteur de ce type qui peut revendiquer comprendre tout code L<sup>A</sup>T<sub>E</sub>X, puisqu'il utilise L<sup>A</sup>T<sub>E</sub>X pour effectuer la traduction !

## 2. Premier contact : la simplicité

### 2.1. La mise en œuvre

L'installation et l'utilisation qui seront décrites ici correspondent à ce qu'on peut trouver sur le CD-ROM  $\TeX$ -Live 5.

$\TeX$ 4ht est un programme en pleine évolution et son auteur y apporte de fréquentes améliorations. Il est judicieux de se reporter fréquemment à son site web [6] pour trouver des mises-à-jour.

La distribution de  $\TeX$ 4ht se compose des fichiers suivants :

- `bin/<arch>/tex4ht` et `bin/<arch>/t4ht` ;
- `texmf/tex4ht/base/*` : ce répertoire contient entre autres un fichier de paramètres `tex4ht.env` nécessaire aux deux programmes précédents ;
- `texmf/tex4ht/ht-fonts-*` : ces répertoires contiennent les fichiers de polices virtuelles ;
- `texmf/tex/generic/tex4ht/*` : le fichier d'extension `tex4ht.sty` et ses fichiers compagnons `*.4ht`
- `texmf/doc/html/tex4ht/*` : la documentation au format HTML dont le point d'entrée est `mn.html`
- `bin/<arch>/{ht,httex,htlatex,httexi}` : des fichiers de script qui pilotent la conversion par  $\TeX$ 4ht pour la rendre aussi transparente que possible<sup>4</sup>.

Les éléments du source  $\LaTeX$  qui ne sont pas adaptables en HTML seront inclus sous forme d'images. Ces images sont produites à partir du document source sous forme d'un fichier `DVI`, puis traduites en PostScript par la commande `dvips`. Pour inclure ces images dans une page HTML, il faut encore transformer ce PostScript en GIF ou en PNG. C'est le rôle du programme `t4ht` de construire ces images. Pour ce faire, `t4ht` devra être aidé par un convertisseur de formats d'images tel que `ImageMagick` ou `NetPBM`. Les phases de conversion sont décrites dans le fichier de paramètres principal de  $\TeX$ 4ht dont le chemin d'accès est `texmf/tex4ht/base/tex4ht.env` :

```
% dvi-to-gif
Gdvips -mode ibmvga -D 110 -f %1 -pp %2 > tex4ht.ps
Gconvert -crop 0x0 -density 110x110 -transparency "#FFFFFF" \
    tex4ht.ps %3
```

---

4. Sous Windows, ce sont des programmes qui possèdent des options légèrement différentes des scripts Unix.

Les lignes introduites par le caractère G concernent la traduction graphique et sont exécutées en séquence comme des appels de programmes externes. Par défaut, la conversion utilise ImageMagick. Elle s'effectue à 110 dpi (points par pouce) et la couleur de transparence est le blanc.

Sous Windows, à l'issue de l'installation du CD-ROM TeXLive 5, T<sub>E</sub>X4ht devrait être opérationnel sans plus de modifications.

Sous Unix, il faut s'assurer de la disponibilité de l'une des familles de convertisseurs graphiques, puis vérifier le paramétrage dans le fichier `tex4ht.env`.

## 2.2. Création d'un premier document HTML

Voici le listing d'un fichier source L<sup>A</sup>T<sub>E</sub>X tout à fait standard que l'on souhaite afficher sous forme de page web.

```

1  \documentclass[a4paper]{article}
   \usepackage[latin1]{inputenc}
   \usepackage[T1]{fontenc}
   \usepackage[frenchb]{babel}
5  \usepackage{colortbl}
   \author{P. Olynome}
   \title{Quelques notions de mathématiques}

   \begin{document}

10  \maketitle

   \HCode{<HR>} % Inersion directe de code HTML
                % HR = Horizontal Rule
15  \tableofcontents
   \HCode{<HR>}

   \section{Première section}
   \subsection{Première sous-section}
20  Une équation :
   $ x^2 + y^2 = z^2 $

   \noindent Une autre équation :
   \[ \frac{\partial f}{\partial x} =
25  \frac{\partial f}{\partial y} \]

   \subsection{Deuxième sous-section}
   Un tableau :
```

*Exemple 1  
fichier source.*

```

30 \begin{tabular}{|>{\columncolor{yellow}\bfseries}l|>
    {\columncolor{red}}p{4cm}||}
    A & 1 \\
    \hline
    B & 2 \\
35 \end{tabular}

\section{Deuxième section}

\end{document}

```

Le résultat de la compilation du fichier donné en exemple 1 par la commande :

```
d:\>htlatex exemple1
```

apparaît dans la page web affichée figure 1.

La simplicité d'emploi est remarquable : un document  $\LaTeX$  est converti vers HTML sans aucune modification. La commande `htlatex` effectue le traitement suivant :

- ajout de `\usepackage{tex4ht}` à l'entête du document ;
- exécution des commandes :

```

d:\>latex exemple1
d:\>latex exemple1
d:\>latex exemple1
d:\>tex4ht exemple1
d:\>t4ht exemple1

```

Il est nécessaire de lancer au moins trois fois  $\LaTeX$  pour résoudre les références et les hyperliens<sup>5</sup> ; pendant ces phases les fichiers `.4ht` constituant l'extension `tex4ht` de  $\LaTeX$  vont créer un fichier `dvi` criblé de commandes `\special`. Ensuite, les deux processeurs spécifiques sont appliqués pour générer tout d'abord le code HTML extrait du fichier `dvi`, puis les éléments manquants tels que les images. On peut résumer le traitement effectué par  $\TeX4ht$  par le schéma de la figure 2.2.

### 2.3. Le paramétrage et les aspects standards de $\LaTeX$

La plupart des fonctionnalités de  $\LaTeX$  ainsi qu'un grand nombre d'extensions et de composants sont pris en compte par défaut. Mais l'un des principaux intérêts de  $\TeX4ht$  est d'être totalement paramétrable par l'utilisateur. C'est ainsi que l'on peut modifier aisément l'apparence du document, soit au moyen d'options dans la ligne de commande lors de l'appel de `htlatex`, soit au moyen d'un fichier de paramétrage.

---

5. Pour la mise en page de tableaux complexes, plus d'exécutions de  $\LaTeX$  peuvent être nécessaires.



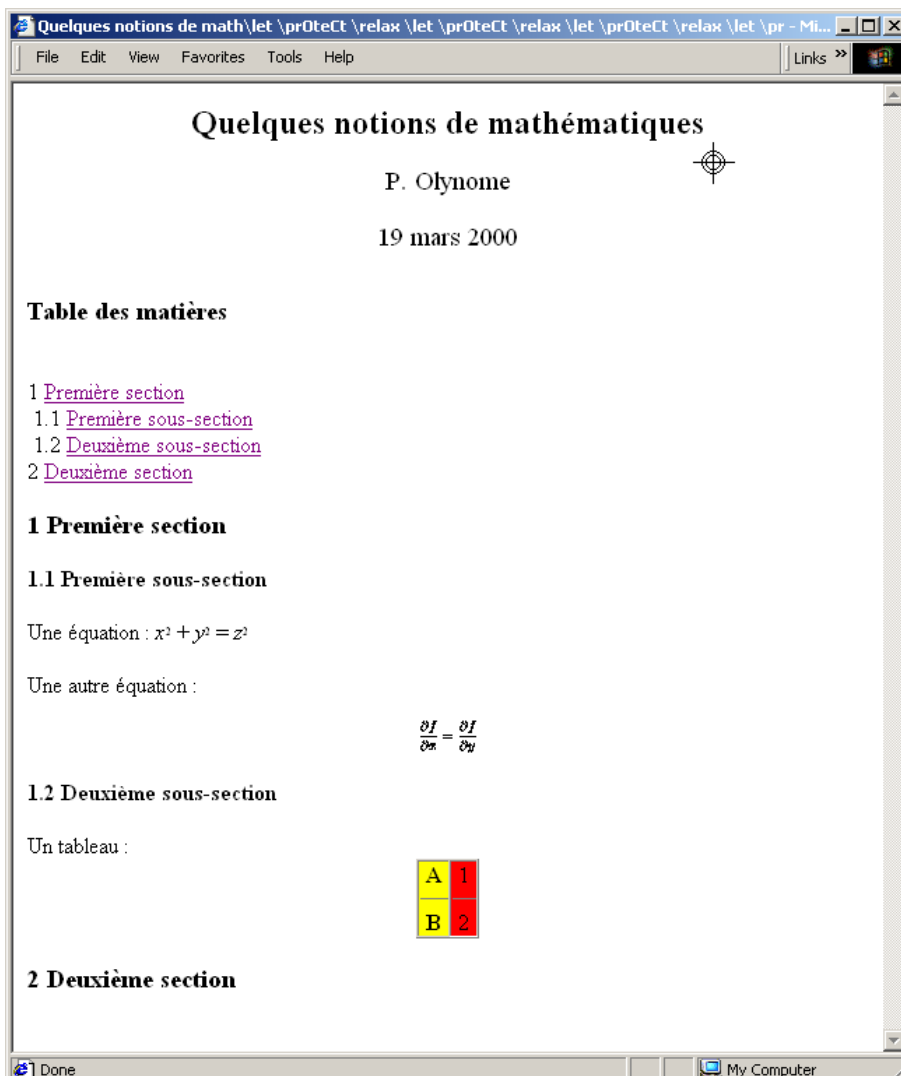
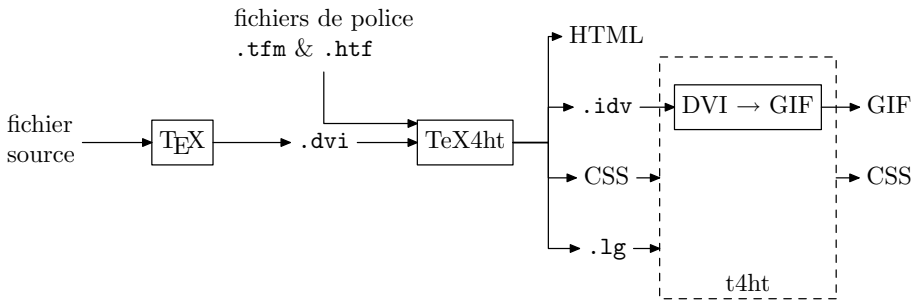


FIGURE 1 – Un premier exemple de page HTML

FIGURE 2 – Synoptique du traitement par  $TeX4ht$ 

### 2.3.1. Les options de la ligne de commande

L'appel de la commande `htlatex` nous indique plusieurs options disponibles (ici sous Windows):

```

D:\>htlatex --help
Usage: htlatex [options] file.tex [params1] [params2] [params3]
  options = --math=(bitmap|mathml)
            --dtd=(html|xhtml|tei|docbook|ebook)
            --mozilla
            --output-dir=<dir>
            --output-name=<name>
            --force
            --dry-run
            --cleanup
            --help
  params1 = (html|xhtml|3.2|mathml.4ht|unicode.4ht|...)*
  params2 = symbol,unicode,...
  params3 = (-p)
  
```

Cette commande se comporte légèrement différemment en environnement Unix, mais les principes généraux restent les mêmes.

Les deux listes `params1` et `params2` peuvent comprendre plusieurs éléments séparés par des virgules. Si le premier paramètre de la liste `params1` est un nom de fichier de paramétrage (l'extension `.cfg` est ajoutée au nom fourni), alors ce fichier sera chargé et il doit avoir la structure donnée dans le listing suivant.

*Structure du  
fichier de  
paramétrage.*

```

1  %
   % Définitions
   %
   \Preamble{options}
5  % Exemple
   %\Preamble{2,sections+,html4.4ht,picmath4.4ht}
   % Signification des options
   % 2          : découpage en fichiers html selon les
   %             2 premiers niveaux de la table des matières
10 % sections+  : générer des entrées de section avec un lien
   %             vers la table des matières
   % html4.4ht  :
   % picmath4.4ht : utiliser ces feuilles de style
   %
15 % Définitions
   % Exemple
   % - Spécification de styles
   % \Css{H1, H2, H3 { font-family: comic sans MS; } }
   % - nouveaux paramétrages
20 % \ConfigureList{maliste}
   %   {\HCode{<div class="maliste">}}{\HCode{</div>}}
   %   {*}{\HCode{<BR>}}
   %
   \begin{document}
25 %
   % A insérer dans l'entête du fichier HTML
   % Exemple
   % Commentaire pour mentionner l'auteur
   % \HCode{<!--   created by FP -->}
30 %
   \EndPreamble

```

La liste `params1` est passée comme argument optionnel à `tex4ht.sty`. De manière équivalente, on aurait pu inclure la ligne :

```
\usepackage[params1]{tex4ht}
```

dans le fichier d'entrée. Le seul objectif des commandes telles que `htlatex` est de rendre la conversion transparente, sans nécessiter aucune modification du fichier d'entrée.

La liste `params1` correspond aux options de `tex4ht.sty`.

La liste `params2` est passée au processeur `tex4ht` ; elle spécifie des polices de caractères virtuelles `.htf`. C'est à l'aide de cette option qu'on demandera l'utilisation de la police `symbol` ou bien de polices `unicode`.

Enfin, la liste `params3` est passée au deuxième processeur `t4ht` (la seule option spécifique pour l'instant est `-p` qui permet de ne pas régénérer les images déjà existantes.).

### 2.3.2. Navigation dans le document

Notre premier exemple n'a produit qu'une seule page HTML. Si l'on veut que le document soit scindé en plusieurs pages correspondant chacune à une unité de texte, c'est très simple à réaliser : il suffit de compiler à nouveau le document avec les options ci-dessous :

```
htatex exemple1.html,2,sections+
```

L'option `HTML` demande explicitement une sortie en HTML 4.0, ce qui est le comportement par défaut. Nous avons besoin de le préciser ici uniquement à cause de la deuxième option (2), qui ne peut pas se trouver en première position (où cette option aurait demandé le chargement d'un fichier de nom `2.cfg`). Cette option spécifie que le document sera découpé sur les frontières de `\section`. Le niveau peut aller de 1 pour `\chapter` à 4 pour `\subsubsection`. La troisième option, `sections+` demande l'ajout de liens hypertexte depuis chaque section vers sa table des matières.

En effet, un fichier  $\LaTeX$  va générer plusieurs fichiers HTML, chacun d'entre eux pouvant être doté de sa propre table des matières. À cet effet,  $\TeX_4\text{ht}$  définit une extension de la commande `\tableofcontents` :

```
\tableofcontents[chapter,appendix,section,likesection]
```

qui demande la création d'une table des matières pour les entrées de type `\chapter` (y compris les annexes), `\section` et `\section*`. La commande `\TocAt{units}` requiert l'insertion automatique d'une table des matières pour chaque type d'unité et permet de spécifier les entrées qui y apparaîtront. L'ordre :

```
\TocAt[chapter, section, /likesection]+
```

crée une table des matières à chaque commande `\chapter`, qui inclue les titres de niveau `\section`, et qui s'arrête à la prochaine commande `\chapter` ou `\section*`.

Une variante `\TocAt*{units}` insère la table des matières locale après le préambule de la section (c'est-à-dire, le texte entre la commande de section et sa première sous-section) plutôt que juste derrière le titre.

### 2.3.3. Du code HTML en ligne à volonté

Il est très simple d'inclure du code HTML directement dans le document  $\LaTeX$  au moyen des commandes suivantes :

- `\HCode{ }`, se reporter au listing de l'exemple 1,
- `\Hnewline`, insertion forcée de retour chariot,
- `\HChar{ }`, insertion littérale d'un caractère ayant le code donné.

Dans l'exemple fourni, des lignes horizontales sont ajoutées pour délimiter la table des matières.

Il est à noter également que la condition `\if\html...\else...\fi` permet de commander l'inclusion ou l'exclusion de parties du document à la présence de l'option `HTML`.

### 2.3.4. Les mathématiques

Pour reproduire une formule mathématique, les environnements `\(...\)` et `\[...\]` provoquent la création d'une image pour l'ensemble de la formule, alors que les environnements `$$...$$` et `$$...$$` utilisent une composition à base des symboles disponibles en `HTML` et d'images pour le reste.

Il est possible de redéfinir ce comportement en ajoutant dans un fichier de paramétrage (voir section 2.4) une commande du type :

```
\Configure{$}{\Picture+{}}{\EndPicture}{} | %$
```

### 2.3.5. Les hyperliens et les images

T<sub>E</sub>X<sub>4</sub>ht travaille conjointement avec `hyperref` qui permet de gérer les hyperliens de manière automatique. T<sub>E</sub>X<sub>4</sub>ht ajoute quelques possibilités à la panoplie offerte par L<sup>A</sup>T<sub>E</sub>X, notamment au niveau de la commande `Picture`.

La commande :

```
\Picture[Mon image]{http://chez.moi.org/image.gif ID="image"}
```

va provoquer la création d'une référence à une image située à l'url indiquée.

Une variante de cette commande se présente sous la forme suivante :

```
\Picture+[Mon image]{image.gif}
  \begin{tabular}{cc}
  ...
  \end{tabular}
\EndPicture
```

Dans ce cas, T<sub>E</sub>X<sub>4</sub>ht a généré une image nommée `image.gif` pour représenter le contenu du tableau, alors qu'il aurait généré une table `HTML` sans l'appel de `Picture+`. On peut utiliser cette commande pour tout fragment du source L<sup>A</sup>T<sub>E</sub>X.

Une option spécifique de l'extension `tex4ht.sty` permet de choisir le type d'image caractérisé par son extension : cette valeur vaut `gif` par défaut, mais peut être changée en spécifiant `png` ou `jpg` comme argument au chargement de `tex4ht.sty`.

## 2.4. Modifier la présentation visuelle du document

T<sub>E</sub>X<sub>4</sub>ht encapsule les commandes L<sup>A</sup>T<sub>E</sub>X à sa manière et fournit en plus des points d'entrée (*hooks*) pour redéfinir ou modifier le comportement par défaut. Ces points d'en-

trée sont accessibles au travers du fichier de paramétrage qui est chargé par l’extension `tex4ht.sty` lors de la compilation par  $\LaTeX$ .

L’introduction de modifications dans le fichier de paramétrage peut se faire de deux manières.

1. Soit le point d’entrée est déjà défini et utilisé par  $\TeX4ht$  ; dans ce cas, il suffit de le redéfinir, en sachant que la nouvelle définition par l’utilisateur remplace celle par défaut.
2. Soit le point d’entrée n’existe pas – c’est le cas pour un environnement défini par l’utilisateur comme l’environnement `test` ci-dessous, dans ce cas il est toujours possible de définir un point d’entrée par la commande `\NewConfigure` selon le schéma suivant :

```
\newcommand{\test}{\Avant Pour un essai. \Après}
\NewConfigure{test}[2]{%
  \newcommand\Avant{#1}%
  \newcommand\Après{#2}}
\Configure{test}{\HCode{<H2>}}{\HCode{</H2>}}
\test                % Texte en << style >> H2
\Configure{test}{}{}
\test                % Texte en << style >> paragraphe
```

La commande `\Configure{test}` permet d’encadrer par des balises – ou quoi que ce soit de plus général – le contenu de l’environnement `test`.

C’est par ce mécanisme très général appliqué extensivement au travers des fichiers d’extension `*.4ht` que  $\TeX4ht$  arrive à traiter presque tout fichier  $\LaTeX$  sans modification. En fait,  $\TeX4ht$  n’est pas un traducteur vers HTML au sens habituel, mais plutôt un traducteur de  $\LaTeX$  vers ... tout ce que l’on souhaite. D’ailleurs,  $\TeX4ht$  est capable de générer du XML ou du MATHML comme nous le verrons dans la section 3.

On peut se demander où trouver la liste exhaustive des points de configuration. Une option supplémentaire de l’extension `tex4ht.sty` nommée `info` va en fournir la liste dans le fichier `.log` généré lors de la compilation par  $\LaTeX$  :

```
\Preamble{2,html,info} % Demande le rapport d’informations
                        % dans le fichier .log
```

Ceci est particulièrement utile pour la génération XML (voir section 3.3).

Certaines parties des fichiers d’extension `.4ht` reprennent des constructions des extensions  $\LaTeX$  correspondantes. Il arrive que des incompatibilités apparaissent lors de changements de version. On peut les régler assez rapidement en notant les différences entre les fichiers `.sty` ou `.cls` et en les reportant dans les fichiers `.4ht` ou des macros (presque) identiques sont utilisées. Dans tous les cas, Eitan Gurari publie très fréquemment des mises à jour de l’ensemble de ses macros (voir [6]).

### 2.4.1. Un deuxième exemple

Plutôt qu'un long discours, voici un second exemple utilisant un fichier de paramétrage.

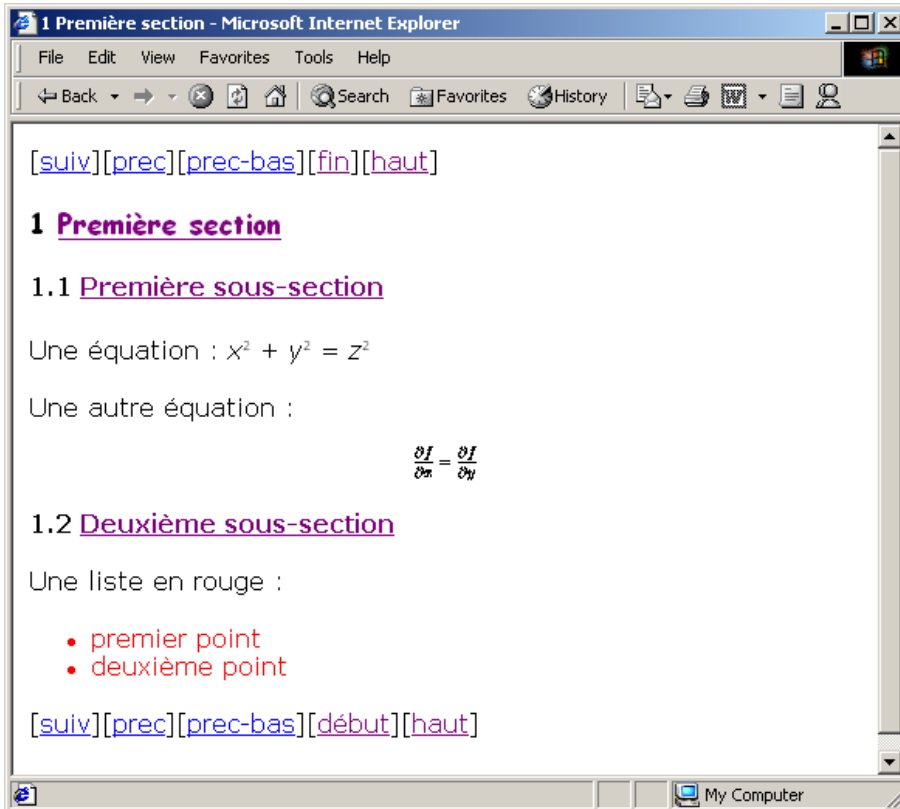


FIGURE 3 – Une page HTML obtenue à partir du fichier `exemple2.tex`

La compilation par la commande :

```
d:\>htlax exemple2 exemple2
```

permet d'obtenir le résultat observé en figure 3.

Nota : dans la suite, aucune commande de compilation plus complexe que celle-ci ne sera utilisée.

Les fichiers `exemple2.tex` et `exemple2.cfg` sont reproduits ci-dessous.

*Exemple 2* *fichier source.*

```

1 \documentclass[a4paper]{article}
  \usepackage[latin1]{inputenc}
  \usepackage[T1]{fontenc}
  \usepackage[frenchb]{babel}
5  \usepackage{colortbl}
  \author{P. Olynome}
  \title{Quelques notions de mathématiques}

  \newenvironment{maliste}{\begin{itemize}}{\end{itemize}}
10 \begin{document}

  \maketitle

  \tableofcontents
15  \section{Première section}

  \subsection{Première sous-section}

20  Une équation :
  $ x^2 + y^2 = z^2 $

  \noindent Une autre équation :
  \[ \frac{\partial f}{\partial x} =
25  \frac{\partial f}{\partial y} \]

  \subsection{Deuxième sous-section}

  Une liste en rouge :
30  \begin{maliste}
  \item premier point
  \item deuxième point
  \end{maliste}

35  \section{Deuxième section}

  \end{document}

```



*Exemple 2  
fichier de  
paramétrage.*

```
1  %
   % Les arguments de \Preamble peuvent également
   % être fournis sur la ligne de commande.
   % Ici, on requiert HTML4.0, coupure sur \section
5  % et des liens de chaque page vers sa table des matières.
   \Preamble{2,sections+,info}

   \catcode`\:=11 % inactivation de ce caractère
                   % qui est actif en français

10 %
   % Modification de style : changement de police pour les
   % titres H1, H2, H3 et pour le corps du texte.
   %
   \Css{H1, H2, H3 { font-family: comic sans MS; } }
15 \Css{BODY { font-family: verdana; } }
   \Css{.alltt .underline{ text-decoration:none;
                           color:green; }}
   \Css{.maliste {color:red; }}

20 %
   % Contribution d'un accent malheureux dans le titre
   %
   % \Contribute{TITLE}{\def é{&eacute;}}

25 % Si Babel et frenchb sont détectés, on change les
   % étiquettes des boutons de navigation.
   \ifx\languagenam\undefined
   \else
     \iflanguage{frenchb}%
30   { \Configure{crosslinks}{[]{}
        {suiv}{prec}
        {prec-bas}{début}{fin}{haut}}
     {}
   \fi

35 % Inclusion de mots-clés pour indexation par les moteurs
   % de recherche. Les deux paramètres seront utilisés pour
   % baliser le titre.
   \Configure{TITLE}
40   { \HCode{<title>}
     { \HCode{</title>\Hnewline
        <meta name="description"
        content="HTML PDF XML tex latex">
     \Hnewline}
45   }
```

```

%
% Modification de présentation : justification à gauche
% Paramètres 1 et 2 : configurent une marque de note de
50 %                bas de page ainsi que le corps de
%                la note correspondante.
% Paramètres 3 et 4 : balisent l'auteur.
% Paramètres 5 et 6 : balisent la date.
% Paramètre 7      : représentation du "and".
55 % Paramètre 8      : définit les fins de ligne à utiliser
%                dans le cas de l'inclusion d'un
%                environnement "tabular".
% \HChar{160} donne un espace insécable
%
60 \Configure{thanks author date and}
    {}{}
    {\HCode{<div align="left">}}{\HCode{</div>}}
    {\HCode{<div align="left">}}{\HCode{</div>}}
    {\HChar{160}\HChar{160}\HChar{160}}{\HCode{<br>}}
65
%
% Ajout de lignes horizontales et des icônes des validateurs
%
% Paramètre 1 et 3 : balises de la tables des matières
70 % Paramètre 2 : inséré en fin de ligne
% Paramètre 4 : inséré en fin de paragraphe
% Paramètre 5 : ?
\Configure{tableofcontents}
  {\HCode{<div class="tableofcontents">} \HCode{<HR>}}
75  {}
  {\IgnorePar\HCode{</div>}
   \HCode{<HR>}\ShowPar \HCode{<br>}
   \HCode{}
80  \HCode{}
%    On devrait plutôt spécifier des pointeurs vers le W3C !
    }
    {\HCode{<br\xml:empty>}}
    {}
85
%
% Cette liste sera mise en valeur en suivant les attributs
% spécifiés plus haut (ie: couleur rouge)
% C'est une configuration naïve, la configuration par défaut
90 % des listes classiques est plus complexe.
% Paramètre 1      : type de liste
%                (itemize, enumerate, etc.)

```

```

% Paramètre 2 et 3 : balisent la liste
% Paramètre 4 et 5 : balisent l'étiquette
95 \ConfigureList{maliste}
    {\HCode{<ul class="maliste">}}
    {\HCode{</ul>}}
    {\HCode{<li class="maliste"> \DeleteMark}
    {}
100 \catcode`\:=13 % Restauration du catcode avec french.sty

\begin{document}

% Exemple de code à placer en entête du fichier HTML
105 % Attention aux caractères actifs en Français !
%
\ifnum\catcode`\!=13
\HCode{<\noexpand!--      created by FP -->}
\else
110 \HCode{<!--      created by FP -->}
\fi

\EndPreamble

```

L'utilisation de feuilles de style css permet très facilement de changer l'apparence des pages obtenues. Chaque élément L<sup>A</sup>T<sub>E</sub>X se voit attribuer une classe, et la commande `\Css{}` permet la redéfinition des attributs de cette classe. C'est ainsi que l'on change les polices par défaut des titres de section et du corps du paragraphe ou bien la couleur utilisée pour afficher l'environnement maliste.

## 2.5. Les panneaux ou *frames*

Il existe un support minimal pour les panneaux – la division de la page du navigateur en plusieurs zones indépendantes, ou frames – qui peut être illustré par l'exemple donné dans les listings suivants.

```

1 \Preamble{html,frames}

\begin{document}

5 \HorFrames{*,*}
  \Frame{2} \Frame{1}
\EndFrames

\EndPreamble

```

*Exemple 3  
utilisation des  
panneaux  
fichier de  
paramétrage.*

*Exemple 3*  
*utilisation des*  
*panneaux*  
*fichier source.*

```

1 \documentclass[a4paper]{article}
  \usepackage[latin1]{inputenc}
  \usepackage[T1]{fontenc}
  \usepackage[frenchb]{babel}
5 \author{P. Olynome}
  \title{Quelques notions de mathématiques}

  \begin{document}

10 \HPage{Toc}

  \maketitle

  \tableofcontents
15 \Link{1}{2}Content\EndLink
  \EndHPage{}

  \HPage{Content}
  \section{Première section}
20 \subsection{Première sous-section}

  \subsection{Deuxième sous-section}

25 \section{Deuxième section}

  \Link{2}{1}Toc\EndLink
  \EndHPage{}

30 \end{document}

```

Le résultat apparaît en figure 4.

Récemment, Eitan Gurari a paramétré la table des matières pour faire pointer les entrées dans le deuxième panneau, de manière à obtenir un catalogue analogue à celui disponible dans les documents PDF.

En utilisant sur un fichier L<sup>A</sup>T<sub>E</sub>X standard un fichier de paramétrage tel que celui ci-dessous, on obtient automatiquement une disposition telle que celle présentée à la figure 5.

*Paramétrage*  
*pour une*  
*utilisation*  
*avancée des*  
*panneaux.*

```

1 \Preamble{3,frames}
  \Configure{BODY}
  {%
5   \HorFrames{*,3*}\Frame{s1}\Frame[ name="f2" ]{s2}
   \NoFrames

```

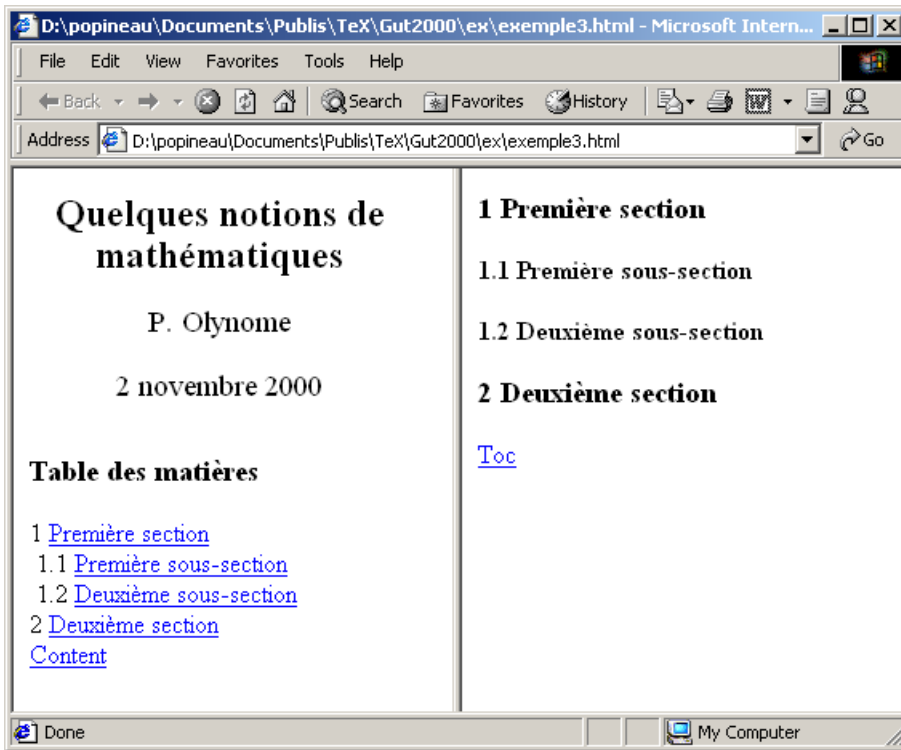


FIGURE 4 – Écran obtenu en utilisant des panneaux

```

\PushStack\eBody{
\SubConfig
\HPage{p1}{table of contents}\Link{}{s1}\EndLink
10 \bgroup
    \Configure{Link}{a}{target="f2" href={}{name={}}
    \tableofcontents
\egroup
\EndHPage{p1} \Tg<br />
15 \HPage{p2}{text}\Link{}{s2}\EndLink
}
{ \EndHPage{p2}}
\Configure{DOCTYPE}
{\HCode{<!DOCTYPE html
20 PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
\Hnewline
"http://www.w3.org/TR/html4/frameset.dtd">

```

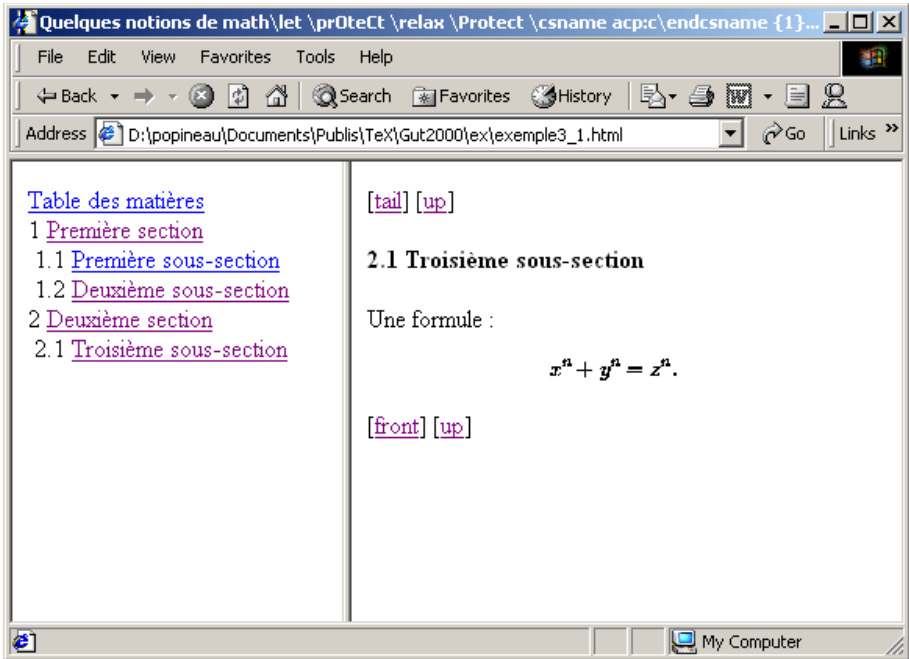


FIGURE 5 – Table des matières automatique dans un panneau séparé.

```

\Hnewline}}

25 \def\SubConfig{%
    \Configure{BODY}
        {\PushStack\eBody{x}\Tg<body>}
        {\PopStack\eBody\temp
         \ifx \temp\empty \EndFrames \else\Tg</body>\fi}
30 \Configure{DOCTYPE}
    {\HCode{%
     <!DOCTYPE html
      PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
        \Hnewline
35     "http://www.w3.org/TR/html4/loose.dtd">
        \Hnewline}}
    }

\begin{document} \EndPreamble

```

## 2.6. Un exemple plus complexe

Les pages web du site de la conférence GUT2001 situées à : <http://www.gutenberg.eu.org/manif/gut2001/index.html> ont été générées à l'aide de T<sub>E</sub>X4ht ; résultat est donné en figure 6.



FIGURE 6 – *Le site web de la conférence GUT2001.*

Le listing complet nécessaire à l'obtention de ce résultat est un peu long pour être inclus intégralement ici, néanmoins nous pouvons donner quelques indications sur les moyens d'obtenir ce résultat.

Tout d'abord, l'affichage de la date courante nécessite d'inclure du code Javascript :

```
\Preamble{3,html,xhtml,frames,javascript,info}

\Configure{JavaScript}
{\HCode{<script type="text/JavaScript" ><!--\Hnewline}}
{\HCode{/-->\Hnewline </script>}}

% Autres instructions de paramétrage.
```

```

\begin{document}
\catcode\;11 % Actif en français
\JavaScript
function mytoday(w) {
  today=new Date();
  var semaine=new Array('Dimanche','Lundi','Mardi','Mercredi',
                        'Jeudi','Vendredi','Samedi');
  var mois=new Array('janvier','février','mars','avril',
                    'mai','juin','juillet','août','septembre',
                    'octobre','novembre','décembre');
  if (today.getYear()== 99)
    w.document.write("Aujourd'hui ",today.getDate(),
                    " ",mois[today.getMonth()],
                    " 19",today.getYear());
  else
    w.document.write("Aujourd'hui ",today.getDate(),
                    " ",mois[today.getMonth()]," 2000");
}
\EndJavaScript
\catcode\;13 % Remettre le code
\EndPreamble

```

Il faut donc spécifier l'option correspondante dans le préambule, et également configurer les balises pour délimiter le code JAVASCRIPT. Ce code est lui même inclus dans la partie suivant le `\begin{document}`. La fonction JAVASCRIPT prend ici comme argument la fenêtre dans laquelle la date doit être affichée. Le caractère « ; » a été inactivé temporairement, mais une autre manière de procéder pourrait consister à utiliser la commande `\string`, éventuellement pour d'autres caractères qui sont activés en français et moins fréquemment utilisés.

Le paramétrage des panneaux est un peu plus complexe que celui vu précédemment :

```

\def\FrameName{"menu"}

\Configure{BODY}{%
  \HorFrames[frameborder="NO" border="0"
             framespacing="0" rows="*"]{* , 3*}
  \Frame[ name = "menu" ]{s1}
  \Frame[ name="main" ]{s2}
  \NoFrames
  \PushStack\eBody{}
  \SubConfig
  \HPage[p1]{table of contents}\Link{}{s1}\EndLink
  \bgroup
  \Configure{Link}{a}{target="main" href=}

```



```

        {onMouseOut="this.style.color='white';"
          onMouseOver="this.style.color='orange';\
                      window.status=''; return true"
          name={}}
\tableofcontents
\egroup
\def\FrameName{"main"}
\EndHPage{p1} \Tg<br />
\HPage{p2}{text}\Link{{s2}\EndLink
}
{\EndHPage{p2}}

\Configure{DOCTYPE}
{\HCode{<!DOCTYPE html
        PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
        \Hnewline
        "http://www.w3.org/TR/html4/frameset.dtd">
        \Hnewline}}

\def\SubConfig{%
  \Configure{BODY}
  {\PushStack\eBody{x}\Tg<body id=\FrameName>
  {\PopStack\eBody\temp
   \ifx \temp\empty \EndFrames \else\Tg</body>\fi}
  \Configure{DOCTYPE}
  {\HCode{<!DOCTYPE html
        PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
        \Hnewline
        "http://www.w3.org/TR/html4/loose.dtd">
        \Hnewline}}
  }
}

```

Dans les tables de matières, les liens ont été reconfigurés de manière à changer dynamiquement de couleur quand la souris passe dessus. Ceci est pris en charge par des instructions DHTML. Malheureusement le code associé n'est pas compatible avec Netscape 4.7. Ceci ne serait pas gênant si la console JAVASCRIPT ne s'ouvrait pas pour indiquer l'erreur.

On a ajouté un attribut `id=` pour identifier les panneaux. De cette manière, les feuilles de styles peuvent servir à donner des attributs spécifiques aux deux panneaux :

```

% Le fond est bleu pour le panneau d'ID menu
\Css{BODY\#menu {background-color: blue; }}
% Les liste énumérées sont en blanc
\Css{\#menu ul.white { color: white;
                      font-family: Verdana, Helvetica, Arial;
                      font-face: sans-serif; font-size: 13}}

```

```

% Une classe pour écrire en blanc en tout petit
\Css{\#menu .small-white { color: white;
      font-family: Verdana, Helvetica, Arial;
      font-face: sans-serif; font-size: 10 }}
% Un lien dans le contexte d'un ID=menu
\Css{\#menu A:link {color: white;}}
% Un line dans le contexte d'un ID=menu
\Css{\#menu A:visited {color: white; }}

```

Enfin, voici le paramétrage de la table des matières qui permet d'obtenir le résultat escompté:

```

\Configure{tableofcontents}{%
  \HCode{<table cellpadding=10 cellspacing=0> <tr> <td>}
  \HCode{<div class="center">}
  \HCode{<a target="main" href="http://www.gutenberg.eu.org">}
  \Picture[GUTenberg]{images/petitGUT.gif width="145pt"
    alt="GUTenberg" border="0"}
  \HCode{</a>}
  \HCode{<span class="small-white">}
  \HCode{<script language="Javascript"> <!--\Hnewline}
  \HCode{parent.mytoday(this);\Hnewline}
  \HCode{/-->}
  \HCode{</script>}
  \HCode{</span>}
  \HCode{</div> </td> </tr>}
  \HCode{\Hnewline <tr> <td> <div class="tableofcontents">}
  } % Before
{} % At end
{
  \IgnorePar\HCode{</div> <div class="center">}
  \HCode{<HR>}\ShowPar \HCode{<br>}
  \HCode{<address> <span class="small-white">}
  \HCode{<a target="main" href="http://www.gutenberg.eu.org">}
  Copyright \copyright\ GUTenberg 2000, tous droits réservés
  \HCode{</a><br>}
  Last modified on: \today\ \now
  \HCode{</span> </address></div> </td> </tr> </table>}
  } % After
{\HCode{<br>}} % At paragraph break
{}

```

On remarque en particulier l'appel au code JAVASCRIPT pour insérer la date du jour, encadré par une balise span locale pour changer les attributs du texte. L'ensemble est inclus dans une table qui permet de jouer sur les espacements au moyens de styles appropriés (attribut padding).

L'ensemble du texte est contenu dans un fichier source `.tex` qui utilise des commandes `\section*{}` pour obtenir des titres de section non-numérotés. Par défaut, T<sub>E</sub>X<sub>4</sub>ht génère également le titre de la section dans le panneau de droite, ce qui n'est pas très heureux car il est déjà dans le panneau de gauche. Il faut donc trouver un moyen de le faire disparaître :

```
\Css{.likesectionHead { color: "\#FFFFFF"; }}
```

Il suffit d'identifier le nom de la classe utilisée par T<sub>E</sub>X<sub>4</sub>ht et de lui affecter une couleur identique à celle du fond et le tour est joué. On pourrait souhaiter le faire dynamiquement en fonction du fait que le navigateur supporte ou non les panneaux<sup>6</sup>.

Il est à noter que les panneaux semblent actuellement délaissés au profit du découpage de la page en zones à l'aide de la balise DIV et des options de positionnement fin qu'ils permettent.

Pour terminer, il est bon de noter que l'extension `hyperref` de L<sup>A</sup>T<sub>E</sub>X supporte la création de formulaires, qui sont pris en compte aussi bien par PDF<sup>L</sup>A<sub>T</sub>E<sub>X</sub> que par T<sub>E</sub>X<sub>4</sub>ht. Le formulaire de préenregistrement pour la conférence est également généré à partir du source L<sup>A</sup>T<sub>E</sub>X donné ci-dessous.

*Création d'un  
formulaire.*

```
1  \documentclass[a4paper]{article}
   \input gut2001-preamble.tex
5  \title{}
   \begin{document}
   % Le logo de la conférence
10 \begin{center}
    \Picture[GUT2001]{images/logogut.gif width="320pt"}
    \end{center}
   \begin{center}
15  \Large \textbf{Pré-inscription au congrès GUT2001}\!
    \mbox{}
    \end{center}
   Pour vous pré-inscrire complétez ce formulaire (n'oubliez
20 pas de confirmer avant le 30 avril 2001).
   Les champs indiqués \textcolor{red}{en rouge} sont
   obligatoires.
25 % Un seul formulaire par page autorisé.
```

---

6. Laissé à titre d'exercice au lecteur!

```

\begin{Form}
  \textbf{\textcolor{red}{Coordonnées}}
  \begin{tabular}[ht]{r1}
    % pour espacer les cellules
30  \Css{\#TBL-\TableNo-
      { padding\string: 4pt 12pt }}
    \Css{\#TBL-\TableNo- TD
      { padding\string: 4pt 12pt }}
    \textcolor{red}{Nom}
35    & \TextField[width=12cm,name=Nom]{} \\
    \textcolor{red}{Prénom}
      & \TextField[width=12cm,name=Prenom]{} \\
    \begin{minipage}[ht]{0.3\textwidth}
      \textcolor{black}{Labo \\ Établissement\\ Entreprise}
40    \end{minipage}
    & \TextField[width=12cm]{} \\
    \textcolor{red}{Adresse} & \TextField[width=12cm]{} \\
    \textcolor{red}{Code Postal}
      & \TextField[width=5cm]{} \\
45    \textcolor{red}{Ville} & \TextField[width=5cm]{} \\
    \textcolor{red}{Pays} & \TextField[width=5cm]{} \\
    \textcolor{black}{Téléphone}
      & \TextField[width=5cm]{} \\
    \textcolor{black}{Télécopie}
50    & \TextField[width=5cm]{} \\
    \textcolor{black}{Email} & \TextField[width=5cm]{} \\
    \end{tabular}

    \textbf{\textcolor{red}{TARIF}}

55  \begin{tabular}[ht]{p{0.4\textwidth}p{0.4\textwidth}}
    \Css{\#TBL-\TableNo- { padding\string: 4pt 12pt }}
    \Css{\#TBL-\TableNo- TD { padding\string: 4pt 12pt }}
    \begin{minipage}[ht]{0.4 \textwidth}
60    \raggedleft Adhérent de GUTenberg\\(ou d'un autre LUG)
    \end{minipage}
    &
    \begin{minipage}[ht]{0.4 \textwidth}
      % Un choix multiple par boutons radio. Il suffit de donner
65    % le même attribut 'name' pour que les boutons fassent
      % partie du même choix.
      \ChoiceMenu[radio,name=tarif]{}{Normal (700 FRF)} \\
      \ChoiceMenu[radio,name=tarif]{}{Étudiant(350 FRF)}
    \end{minipage} \\
70  \begin{minipage}[ht]{0.4 \textwidth}
    \raggedleft Non encore adhérent
  
```

---

```

\end{minipage}
&
\begin{minipage}[ht]{0.4\textwidth}
75 \ChoiceMenu[radio,name=tarif]{}
    {Normal + Adhésion (900 FRF)}\}
\ChoiceMenu[radio,name=tarif]{}
    {Étudiant + Adhésion (450 FRF)}
80 \end{minipage}
\end{tabular}

\textbf{Choix des tutoriels}

\begin{center}
85 X : XML -- H : TeX4ht -- O : Omega -- L: LaTeX
\end{center}

\begin{tabular}[ht]{p{0.2\textwidth}p{0.2\textwidth}
90     p{0.2\textwidth}p{0.2\textwidth}
        p{0.2\textwidth}p{0.2\textwidth}}
    \Css{\#TBL-\TableNo- { padding\string: 12pt }}
    \Css{\#TBL-\TableNo- TD { padding\string: 12pt }}
Tutoriels du 16 mai
& \ChoiceMenu[radio,name=tut16]{}{X}
95 & \ChoiceMenu[radio,name=tut16]{}{H}
& \ChoiceMenu[radio,name=tut16]{}{O}
& \ChoiceMenu[radio,name=tut16]{}{H}
& \ChoiceMenu[radio,name=tut16]{}{O}\}
Tutoriels du 17 mai
100 & \ChoiceMenu[radio,name=tut17]{}{L} & & &
\end{tabular}

\def\initval{Il est entendu que mon inscription ne sera
105     effective que lorsqu'elle sera confirmée par écrit et
        accompagnée d'un chèque ou d'un bon de commande.}

% Un champ de texte multilignes. Les dimensions sont
% ajustables, le statut (lecture seulement) également.
\textbf{Commentaires}
110 \begin{center}
\TextField[multiline,height=40pt,width=0.8\textwidth,
        value=\initval,name=commentaire]{}
\end{center}

115 % Les boutons pour soumettre ou annuler le formulaire
\begin{tabular}[ht]{cc}
    \Css{\#TBL-\TableNo- { padding\string: 12pt }}

```

```

        \Css{\#TBL-\TableNo- TD { padding\string: 12pt }}
        \Reset{Tout effacer} & \Submit{Envoyer}
120 \end{tabular}
        \end{Form}
        \end{document}

%%% Local Variables:
125 %%% mode: latex
        %%% TeX-master: t
        %%% End:

```

La figure 7 montre le résultat obtenu.

Formenligne.html - Microsoft Internet Explorer

File Edit View Favorites Tools Help

**GUTENBERG METZ 2001**

**Pré-inscription au congrès GUT2001**

Pour vous pré-inscrire complétez ce formulaire (n'oubliez pas de confirmer avant le 30 avril 2001).

Les champs indiqués **en rouge** sont obligatoires.

**Coordonnées**

Nom

Prénom

Adresse

Code Postal

Ville

Pays

**TARIF**

Adhérent de GUTenberg  
(ou d'un autre LUG)

Non encore adhérent

Normal (700 FRF)

Étudiant(350 FRF)

Normal + Adhésion (900 FRF)

Étudiant + Adhésion (450 FRF)

Done My Computer

FIGURE 7 – Le formulaire de préinscription à la conférence GUT2001.

### 3. Au delà de HTML...

#### 3.1. Conversion vers XML

HTML est une cible immédiate, mais XML est plus intéressant à long terme. Et c'est là que T<sub>E</sub>X<sub>4</sub>ht fait la différence : il maîtrise parfaitement le document L<sup>A</sup>T<sub>E</sub>X et donc sa structure.

La conversion vers XML met pleinement en œuvre le mécanisme de points d'entrée vers les commandes L<sup>A</sup>T<sub>E</sub>X. Pour plus de clarté, j'ai choisi de présenter un exemple très simple où le document L<sup>A</sup>T<sub>E</sub>X d'origine fait essentiellement appel à des commandes définies par l'utilisateur.

*Une recette de  
cuisine  
classique.*

```

1  \documentclass[a4paper]{article}
   \usepackage[latin1]{inputenc}
   \usepackage[T1]{fontenc}
   \usepackage[frenchb]{babel}
5
   \ifx\ifHtml\undefined
   \newcommand\PreParts{}
   \newcommand\PostParts{}
   \newcommand\PreCui{}
10  \newcommand\PostCui{}
   \newcommand\PreTh{}
   \newcommand\PostTh{}
   \newcommand\PreCal{}
   \newcommand\PostCal{}
15  \newcommand\PreDur{}
   \newcommand\PostDur{}
   \fi

   \newcommand{\parts}[1]
20   {Pour \PreParts #1 \PostParts personnes.\\}
   \newcommand{\calories}[1]{Calories : \PreCal #1 \PostCal.\\}
   \newcommand{\cuisson}[2]
   {\PreCui Cuisson : \PreDur #1 \PostDur
   à thermostat \PreTh #2 \PostTh. \PostCui\\}
25  \newenvironment{ingredients}
   {\begin{description}}{\end{description}}
   \newenvironment{instructions}
   {\begin{enumerate}}{\end{enumerate}}

30  \begin{document}

   \section{Quiche Lorraine}
   \parts{8}

```

```

\calories{150}
35 \begin{ingredients}
\item[250g] pâte à tarte
\item[150g] lardons
\item[3] {\oe}ufs
\item[100g] crème fraîche
40 \item[20cl] lait
\item[80g] gruyère
\end{ingredients}

\begin{instructions}
45 \item \'Etaler la pâte et garnir un moule à tarte.
\item Faire revenir les lardons.
\item Battre les {\oe}ufs, la crème et le lait.
\item Répartir les lardons sur la tarte,
\item puis le gruyère rapé.
50 \item Recouvrir avec le mélange d' {\oe}ufs battus.
\end{instructions}

\cuisson{40mn}{210 degrés}

55 \end{document}

```

Il s'agit d'une recette de cuisine. On imaginerait sans peine un style plus riche pour typographier ce type de recette. Des points d'entrée `\Pre...` et `\Post...` ont été volontairement introduits dans les commandes de mise en forme. Ces points d'entrée serviront à  $\TeX$ 4ht lors de la traduction vers XML. Le fichier de paramétrage est donné ci-dessous :

```

Paramétrage 1 \Configure{html}{xml}
pour        \Preamble{html,0.0,ref-,fonts}
conversion vers \Configure{HTML}{\IgnorePar{Tg<?xml version="1.0"?>
XML         \Tg<document>}{\Tg</document>}
5          \Configure{section}{\Tg<recette>}{\Tg</recette>}
           {\Tg<titre>}{\Tg</titre>}

\ConfigureList{ingredients}
           {\Tg<ingredients>}{\Tg</ingredients>}
10        {\Tg<quantite>}{\Tg</quantite>}

\ConfigureList{instructions}
           {\EndP \Tg<instructions>
           \def\EndItem{\def\EndItem{\EndP\Tg</p>}}
15        {\EndItem \Tg</instructions>}{\EndItem \DeleteMark}{\Tg<p>}

\NewConfigure{parts}[2]{

```



```

    \newcommand\PreParts{#1}%
    \newcommand\PostParts{#2}}
20 \Configure{parts}{\Tg<parts>}{\Tg</parts>}

\NewConfigure{cal}[2]{
    \newcommand\PreCal{#1}%
    \newcommand\PostCal{#2}}
25 \Configure{cal}{\Tg<cal>}{\Tg</cal>}

\NewConfigure{cui}[6]{
    \newcommand\PreCui{#1}%
    \newcommand\PostCui{#2}
30 \newcommand\PreDur{#3}%
    \newcommand\PostDur{#4}%
    \newcommand\PreTh{#5}%
    \newcommand\PostTh{#6}}
35 \Configure{cui}{\Tg<cuisson>}{\Tg</cuisson>}
    {\Tg<duree>}{\Tg</duree>}
    {\Tg<thermostat>}{\Tg</thermostat>}

\begin{document}

40 \EndPreamble

```

On voit que la génération de code XML consiste essentiellement à émettre des balises (*tags*) à l'aide de la commande `\Tg{}`. Ce qui ne veut pas dire que ce soit toujours très simple!

Le résultat obtenu se présente sous cette forme :

```

1  <?xml version="1.0"?> <document>
    <recette>
    <titre>
      <
5     "x1-10001"></>Quiche Lorraine</titre>
      Pour <parts>8 </parts>personnes.
      Calories : <cal>150 </cal>.
        <ingredients><quantite>
10     250g </quantite>p(circ a)te (grave a) tarte
        <quantite>
        150g </quantite>lardons
        <quantite>
        3 </quantite>&#0339;ufs
        <quantite>
15     100g </quantite>cr(grave e)me fra(circ i)che
        <quantite>

```

*Résultat de la  
conversion vers  
XML*

```

20cl </quantite>lait
      <quantite>
80g </quantite>gruy(grave e)re</ingredients>
20  <instructions>
      <p>(acute E)taler la p(circ a)te
          et garnir un moule (grave a) tarte.
      </p>
      <p>Faire revenir les lardons.
25  </p>
      <p>Battre les &#0339;ufs, la cr(grave e)me et le lait.
      </p>
      <p>R(acute e)partir les lardons sur la tarte,
          puis le gruy(grave e)re rap(acute e).
30  </p>
      <p>Recouvrir avec le m(acute e)lange d&#8217;&#0339;ufs
battus.</p>
      </instructions>
<cuisson>
35  Cuisson : <duree>40mn </duree>(grave a) thermostat
          <thermostat>210degr(acute e)s </thermostat>.
      </cuisson>
</recette>
</document>
40

```

### 3.2. La prise en compte de MathML

$\text{\TeX}$  et  $\text{\LaTeX}$  étant spécialement efficaces dans le traitement des mathématiques,  $\text{\TeX4ht}$  peut traduire des formules mathématiques avec la syntaxe  $\text{\TeX}$  en MathML. En voici un exemple.

```

Exemple de 1  \documentclass[a4paper]{article}
document à  \usepackage[latin1]{inputenc}
convertir vers \usepackage[T1]{fontenc}
MATHML      \usepackage[frenchb]{babel}
5
      \begin{document}

      \section{Test MathML}

10  $$ g(x) = \int_{1}^x \frac{1}{1 + y^2} dy $$

      \end{document}

```

Après avoir lancé la commande :

```
d:\>htlatex exemple5 xhtml,fonts,mathml
```

on obtient sans peine le fragment MathML présenté ci-dessous.

```
1  <math display="block">
    <mi>g</mi>
    <mrow><mo>( </mo><mi>x</mi><mo> )</mo></mrow>
    <mo>=</mo>
5  <msubsup> <mo><imgsrc="cmex10-5a.gif"
        alt=" integral "
        class="cmex-10--5a"
        align="middle" /></mo>
    <mrow><mn>1</mn></mrow>
10 <mrow><mi>x</mi></mrow>
    </msubsup>
    <mfrac>
        <mrow><mn>1</mn></mrow>
        <mrow><mn>1</mn> <mo>+</mo> <msup><mi>y</mi></mrow>
15 <mn>2</mn></mrow></msup></mrow>
    </mfrac>
    <mi>d</mi><mi>y</mi>
    </math>
```

*Résultat de la  
conversion vers  
MATHML*

20

### 3.3. Plus loin vers XML ?

Il existe un début de support pour certaines DTD standards et Eitan Gurari fournit une page [5] d'où l'on peut télécharger des scripts pour appeler T<sub>E</sub>X<sub>4</sub>ht avec des paramètres permettant de cibler une sortie non seulement vers HTML ou XHTML, mais aussi vers Docbook, eBook ou TEI. Ceci peut s'effectuer en combinaison avec une génération des mathématiques sous forme bitmap ou MATHML.

A titre d'exemple, le fichier de la figure 3 converti vers la DTD Docbook en utilisant MATHML fournit le résultat ci-dessous<sup>7</sup>.

```
1  <?xml version="1.0"?>
    <!DOCTYPE article [<!ELEMENT equation (math)>
    <!ENTITY % docbook.dtd
        PUBLIC "-//OASIS//DTD DocBook XML V4.0beta6//EN"
5  "http://www.oasis-open.org/docbook/xml/4.0/docbookx.dtd">
```

*Exemple de  
conversion vers  
DocBook +  
MATHML*

---

7. Après indentation !

```

%docbook.dtd;
<!ENTITY % mathml.dtd PUBLIC "-//W3C//DTD MathML 2.0//EN"
                    "mathml.dtd">
%mathml.dtd;
10 ] >

<!--translated from exemple2.tex, on 2000-06-02 14:16:00,
      by TeX4ht
      (http://www.cis.ohio-state.edu/~gurari/TeX4ht/mn.html)-->
15
<article>
  <title>Quelques notions de mathématiques</title>
  <!--authorP. Olynome-->
  <!--date2 juin 2000-->
20  <section role="likesection">
    <title><anchor id="x1-1000"  ></anchor>
      Table des matières
    </title>
    <toc>
25      <tocpart>
        <tocentry role="sectionToc">
          1&#x00A0;<anchor id="QQ2-1-2"></anchor>
          <xref linkend="x1-20001"  ></xref>
          Première section
30        </tocentry>
        <tocentry role="subsectionToc">
          1.1&#x00A0;<anchor id="QQ2-1-3"  ></anchor>
          <xref linkend="x1-30001.1"  ></xref>
          Première sous-section
35        </tocentry>
        <tocentry role="subsectionToc">
          1.2&#x00A0;<anchor id="QQ2-1-4"  ></anchor>
          <xref linkend="x1-40001.2"  ></xref>
          Deuxième sous-section
40        </tocentry>
        <tocentry role="sectionToc">
          2&#x00A0;<anchor id="QQ2-1-5"  ></anchor>
          <xref linkend="x1-50002"  ></xref>
          Deuxième section
45        </tocentry>
      </tocpart>
    </toc>
    <para></para>
  </section><!--end likesection-->
50  <section role="section">
    <title><anchor id="x1-20001"  ></anchor>

```

```

    Première section
  </title>
  <section role="subsection">
55  <title><anchor id="x1-30001.1" ></anchor>
    Première sous-section
  </title>
  <para>
    Une équation&#x00A0;: <!--l. 21-->
60  <equation role="inline">
    <math>
      <msup><mi>x</mi><mrow><mn>2</mn></mrow></msup>
      <mo>+</mo>
      <msup><mi>y</mi><mrow><mn>2</mn></mrow></msup>
65  <mo>=</mo>
      <msup><mi>z</mi><mrow><mn>2</mn></mrow></msup>
    </math>
  </equation>
  </para>
70  <para>Une autre équation&#x00A0;: <!--l. 24-->
  <equation role="block">
    <math><mrow>
      <mfrac>
        <mrow><mi>&#x2202;</mi><mi>f</mi></mrow>
75  <mrow><mi>&#x2202;</mi><mi>x</mi></mrow>
      </mfrac>
      <mo>=</mo>
      <mfrac>
        <mrow><mi>&#x2202;</mi><mi>f</mi></mrow>
80  <mrow><mi>&#x2202;</mi><mi>y</mi></mrow>
      </mfrac>
    </mrow></math>
  </equation>
85  </para>
  </section><!--end subsection-->

  <section role="subsection">
    <title><anchor id="x1-40001.2" ></anchor>
90  Deuxième sous-section
  </title>
  <para> Une liste en rouge&#x00A0;: </para>
  <itemizedlist>
    <listitem> <para>premier point </para></listitem>
95  <listitem> <para>deuxième point</para></listitem>
  </itemizedlist>

```

```

    </section>
    <!--end subsection-->
100 </section>
    <!--end section-->

    <section role="section">
      <title><anchor id="x1-50002" ></anchor>
105   Deuxième section
      </title>
    </section>
    <!--end section-->
    </article>
110

```

L'implantation complète de la conversion vers ces DTD n'est pas terminée. Selon Eitan Gurari, le principal objectif était de prouver que la conception de son outil était adaptée à ce travail. Une définition plus complète pour ces DTD viendra lorsque la demande se fera sentir plus fortement.

En attendant, pour l'utilisateur qui en aurait besoin, il est possible de pousser un peu plus loin la définition actuelle en listant les points de configuration au moyen de l'option `info` et en complétant au moyen de commandes `Configure{ }` les parties qui ne seraient pas ou mal représentées.

## 4. Conclusion

Il reste bien des aspects de ce système à explorer, que ce soit au niveau des polices virtuelles, de JavaScript<sup>8</sup>, des *images maps* avec le système DraTeX<sup>9</sup>...

J'espère que cet avant-goût des possibilités de T<sub>E</sub>X<sub>4</sub>ht aura convaincu le lecteur de l'intérêt de cet outil. Le meilleur moyen d'en savoir plus reste encore de l'utiliser. Il manque certainement un manuel d'utilisation extensif, mais les références fournies ci-dessous devraient combler en partie cette lacune.

### A. Quelques difficultés

Si les css sont un outil puissant pour rendre plus attrayants les documents affichés, il faut se méfier de la manière dont les navigateurs les interprètent. Par exemple, Netscape

---

8. Voir à ce sujet le support de `exerquiz.sty` qui permet de générer des questionnaires à choix multiples!

9. Système de composition graphique, par le même auteur Eitan Gurari!

4.7 étend la portée d'une balise `<span>` au-delà de ce qui est attendu, ce qui peut provoquer des effets de bord désastreux. Imaginons le code HTML généré suivant :

```
<div class="author" align="center"><a href="mailto:foo"><span
class="cmr-12">Mr Foo</span></a>
<span class="cmr-12">and </span>
<a href="mailto:bar"><span class="cmr-12">Mrs Bar</span></a>
</div>
```

et dans la feuille de style :

```
.cmr-12{font-size:120%;}
```

Netscape va étendre la portée des `<span>` jusqu'à `</div>`, donc chacun des champs va bénéficier d'une augmentation de taille de 120% *par rapport au précédent* ! On peut contourner le problème dans ce cas spécifique en empêchant T<sub>E</sub>X<sub>4</sub>ht de générer des balises `<span>` localement :

```
\Configure{thanks author date and}
  {\HCode{<span class="thanks">}}
  {\HCode{</span>}}
  {\HCode{<div class="author" align="center">}\NoFonts}
  {\EndNoFonts\HCode{</div>}}
  {\HCode{<div class="date" align="center">}}
  {\HCode{</div>}}
  {\HCode{<span class="and">}}~~~\HCode{</span>}}
  {\HCode{<br />}}
\Css{.author {font-size:120\%;}}
```

Une manière plus globale de contourner le problème consiste à appliquer un filtre à l'ensemble des pages produites. T<sub>E</sub>X<sub>4</sub>ht le prévoit et il est possible de demander à t4ht d'appliquer une commande externe sur chaque page. À défaut d'écrire une transformation XSLT ad-hoc, un script Perl tel que celui qui suit peut convenir.

```
1  open IN, "<$ARGV[0]";
   while (<IN>) {
       $s = $s . $_;
   }
5  close IN;

   $s_prev = "";
   while ($s_prev ne $s) {
       $s_prev = $s;
10  # ATTENTION: les trois lignes qui suivent doivent
       # être rassemblées sur une seule et même ligne;
       # des raisons de présentation on nécessité de
       # scinder l'expression.
       $s =~ s@<span\s*class=(\[^\"]*\)\>([\^<]*)
```

*Script Perl  
pour filtrer les  
balises SPAN  
redondantes.*

```

15      </span>\s*<span\s*class=\1>([\^<]*)
      </span>@<span class=\1> \2\3 </span>@gm;
    }
    print $$;
    0;

```

Les utilisateurs auront également remarqué un autre défaut flagrant lors du traitement d'un texte français par exemple : le titre qui apparaît dans la fenêtre du navigateur est erroné. En effet, il ressemble à :

```

Quelques notions de math\let \protect \relax \let \protect
\relax ...

```

Il est évident que le traitement particulier du caractère « é » sous forme de macro est à l'origine de ce résultat.  $\TeX$ 4ht fournit une solution élégante pour contourner ce problème. Il est possible de redéfinir localement les *contributions* de macros à la traduction de blocs. Il suffit d'ajouter dans le fichier de paramètres :

```
\Contribute{TITLE}{\def é{\&eacute;}}|
```

pour corriger l'effet indésirable.

## Bibliographie

- [1] Luc MARANGET & al. Site de référence pour Hevea.  
<http://para.inria.fr/~maranget/hevea/>.
- [2] Ross MOORE & al. Site de référence pour  $\LaTeX$ 2HTML.  
<http://saftack.fs.uni-bayreuth.de/~latex2ht/>.
- [3] W3 Consortium. Documents de référence pour HTML, XML, MATHML, feuilles de style CSS et XSL... <http://www.w3.org/>.
- [4] Eitan GURARI et Sebastian RAHTZ. 20<sup>e</sup> Conférence du  $\TeX$  Users Group, Vancouver, Colombie Britannique, 1999.  
<http://www.cis.ohio-state.edu/~gurari/tug99/>.
- [5] Eitan GURARI. Une liste de scripts pour appeler  $\TeX$ 4ht.  
<http://www.cis.ohio-state.edu/~gurari/TeX4ht/bugfixes2.html>.
- [6] Eitan GURARI. Pointeur vers les mises à jour pour  $\TeX$ 4ht.  
<http://www.cis.ohio-state.edu/~gurari/TeX4ht/bugfixes.html>.
- [7] Eitan GURARI. Site principal de  $\TeX$ 4ht.  
<http://www.cis.ohio-state.edu/~gurari/TeX4ht/mn.html>.
- [8] Eitan GURARI. Une liste de pointeurs vers d'autres traducteurs.  
<http://www.cis.ohio-state.edu/~gurari/TeX4ht/mn.html#resources>.
- [9] Eitan GURARI. 18<sup>e</sup> Conférence du  $\TeX$  Users Group, San Francisco, Californie, 1997.  
<http://www.cis.ohio-state.edu/~gurari/tug97/tug97-d.html>.



- 
- [10] Steve MAYER. Comparaison des résultats produits par plusieurs convertisseurs vers HTML.  
<http://www.mayer.dial.pipex.com/samples/example.htm>.
  - [11] Steve MAYER.  $\TeX$  Converter, interface pour des convertisseurs vers HTML.  
<http://www.mayer.dial.pipex.com/tex.htm>.
  - [12] Michel GOOSSENS & Sebastian RAHTZ, avec Eitan GURARI, Ross MOORE et Robert SUTOR. *The  $\LaTeX$  Web Companion*. Addison-Wesley, 1999.
  - [13] Sebastian RAHTZ. Manuel de référence du paquetage `hyperref`.  
<http://www.tug.org/applications/hyperref/manual.html>.
  - [14] H. WILLIAMSON. *Writing cross-browser dynamic HTML*. Springer-Verlag, 2000.