

Cahiers **GUT**enberg

☞ LE COIN DES GOUROUS

☞ Georges WEIL, Raymond SÉROUL

Cahiers GUTenberg, n° 3 (1989), p. 55-59.

<http://cahiers.gutenberg.eu.org/fitem?id=CG_1989__3_55_0>

© Association GUTenberg, 1989, 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.

Le coin des GOUROUS

Responsable de la rubrique : Georges WEIL

CCS - CNRS, 23 rue du Loess, 67200 Strasbourg ; EARN : C01104@FRCCSC21

1. Macros pour entourer texte ou formule

et l'épaisseur du trait. Ces trois variables sont initialisées par défaut respectivement à 6 points, 6 points et 0,4 points.

Par Georges WEIL, CNRS - Strasbourg.

1.1. But des macros

Ces macros \TeX ¹ tracent un cadre autour d'un texte ou d'une formule mathématique.

1.2. Appel des macros

`\entoure{texte}`

ou

`$$\entoureformule{mtexte}$$`

avec :

`texte` = texte devant être entouré ;
`mtexte` = formule en mode mathématique centré devant être entourée.

1.3. Fonctionnement

La lecture du membre crée trois variables de type dimension `\entourehor`, `\entourever` et `\entouregra` permettant de définir respectivement l'espacement horizontal et vertical entre la boîte et le cadre

¹ou \LaTeX puisque le présent texte a été composé en \LaTeX avec ces macros !

1.4. Premier exemple

```
\setbox0=\vbox{\hsize=5cm
\parindent=20pt \parskip=1pt
Le pr'ésent texte est rang'e
dans une bo'i te de
5 centim'etres de large,
entour'ee d'un cadre
'a .5cm de chaque c'ot'e.\par
Ceci est un nouveau paragraphe.
Remarquez que l'on a
modifi'e dans cette {\tt vbox} les
valeurs de {\tt hsize},
{\tt parskip} et {\tt parindent}.
\par}
\entourehor=0.5cm \entourever=0.5cm
\entouregra=2pt
\centerline{\entoure{\box0}}
```

Le résultat est :

Le présent texte est rangé dans une boîte de 5 centimètres de large, entourée d'un cadre à .5cm de chaque côté.

Ceci est un nouveau paragraphe. Remarquez que l'on a modifié dans cette vbox les valeurs de hsize, parskip et parindent.

1.5. Autres exemples

```

$$\entoureformule{A=\pmatrix{a&z&e&r&t&y\cr
q&s&d&f&g&h\cr
w&x&c&v&b&n\cr}}$$

```

$$A = \begin{pmatrix} a & z & e & r & t & y \\ q & s & d & f & g & h \\ w & x & c & v & b & n \end{pmatrix}$$

```

$$A=\entoureformule{\pmatrix{a&z&e&r&t&y\cr
q&s&d&f&g&h\cr
w&x&c&v&b&n\cr}}$$

```

$$A = \begin{pmatrix} a & z & e & r & t & y \\ q & s & d & f & g & h \\ w & x & c & v & b & n \end{pmatrix}$$

On a $\frac{f(a+h)-f(a)}{h} \approx f'(a)$ pour h petit.

On a $f(a+h) - f(a) \approx hf'(a)$ pour h petit.

Dans ce dernier exemple, la boîte créée par la macro `\entoure` a été baissée afin que la formule soit alignée avec le reste de la ligne. En outre on remarquera que les modifications de `\entourehor` et `\entourever` sont dans un groupe pour qu'elles restent locales.

1.6. Les macros

```

% MACRO ENTOURE
% pour entourer une vbox par un cadre
%
% \entoure#1
%
% \entourehor et \entourever
% sont 2 variables de type dimen
% permettant de definir l'espaceent
% horizontal et vertical
% entre la vbox et les traits.
%
%
% \ifx\mbrentoure\relax\endinput
% \else\let\mbrentoure=\relax\fi
%
% \newdimen\entourehor \entourehor=6pt
% \newdimen\entourever \entourever=6pt
% \newdimen\entouregra \entouregra=0.4pt
% \long\def\entoure#1{\vbox{\hrule
% height\entouregra
% \hbox{\vrule width\entouregra
% \kern\entourehor
% \vbox{\kern\entourever%
% \hbox{#1}\kern\entourever}
% \kern\entourehor\vrule
% width\entouregra}
% \hrule height\entouregra}
%
% \long\def\entoureformule#1{\vcenter{\hrule
% height\entouregra
% \hbox{\vrule width\entouregra
% \kern\entourehor
% \vbox{\kern\entourever%
% \hbox{\displaystyle #1}
% \kern\entourever}
% \kern\entourehor\vrule
% width\entouregra}
% \hrule height\entouregra}

```

1.7. Explications sur les macros

Le principe consiste à mettre le texte à encadrer dans une boîte et de tracer un cadre autour de la boîte

La macro `\entoureformule` utilise la commande `\vcenter` afin que la boîte créée soit centrée. La formule mathématique sera en *displaystyle*.

2. \edef : Définition avec développement immédiat – Application aux fractions continues

Par Raymond SEROUL *Laboratoire de Typographie Informatique, Université Louis Pasteur, Strasbourg*

2.1. Introduction

La définition avec développement immédiat (`\edef` : *expanded definition*) permet d'accumuler petit à petit du matériel dans une même macro. On se sert de `\edef` selon le schéma suivant :

```
\def\accumule{a}
\edef\accumule{\accumule b}
\edef\accumule{\accumule c}
```

Après cette suite de `\edef`, un `\show\accumule` montre que le texte de remplacement de `\accumule` est 'abc'. Voici une application : il s'agit d'écrire une macro `\toto` ayant $n \geq 1$ arguments (nombre non défini à l'avance) et qui doit se comporter ainsi :

`$$\toto[1]$$ (1)`

`$$\toto[1,2]$$ (1(2))`

`$$\toto[\alpha,\beta,\gamma]$$`

`..... ($\alpha(\beta(\gamma))$)`

Ce problème n'est pas si académique qu'il ne le paraît : il s'agit, dans un premier temps, d'étudier le codage des fractions continues.

Il est facile d'écrire une macro qui lit un par un tous les tokens jusqu'à une

balise imposée à l'avance (ici, elle s'appelle `\endessai`) :

```
\def\essai#1{\message{#1}%
\ifx #1\endessai \let\next=\relax
\else\let\next=\toto\fi
\next}
```

Par exemple, la séquence `\essai abc \endessai xyz` écrit abc dans le fichier log puis imprime 'xyz'.

Remarque : il est essentiel d'utiliser le test `\ifx`. Si on change ce test en `\if` (histoire de voir), \TeX proteste en déclarant qu'il ne connaît pas la macro `\endtoto`. Avec un `\ifx`, `\endtoto` est parfaitement accepté comme *token*.

Maintenant, nous sommes prêts. Nous accumulons les parenthèses ouvrantes et les arguments de `\toto` dans la macro `\tete` et les parenthèses fermantes dans la macro `\queue` :

```
\def\tete{()} \def\queue{}}
\def\TOTO#1{%
\ifx #1]\def\next{\tete\queue}%
\else \ifx #1,
\edef\tete{\tete}%
\edef\queue{\queue}%
\else \edef\tete{\tete#1}\fi
\let\next=\TOTO\fi
\next}
```

La conception de la macro `\TOTO` est assez simple. On teste le token `#1` qui suit cette macro :

- si `#1` n'est ni une virgule ni un

crochet fermant, on l'incorpore à `\tete` et on ne modifie pas `\queue`;

- si #1 est une virgule, on ajoute une parenthèse ouvrante à `\tete` et une parenthèse fermante à `\queue`;
- enfin, si #1 est un crochet fermant, on sait que l'on a fini de lire les arguments. On termine alors en écrivant `\tete\queue` dans le texte.

La macro `\TOTO` ne s'occupe pas du crochet ouvrant et elle n'initialise pas `\tete` et `\queue`. Il est facile de remédier à ces petits défauts à l'aide d'une nouvelle macro `\toto` :

```
\def\toto#1{\def\tete{()
\def\queue{}}
\def\TOTO##1{\ifx##1
\def\next{\tete\queue}
\else\ifx##1,
\edef\tete{\tete(}
\edef\queue{\queue)}
\else\edef\tete{\tete##1}\fi
\let\next=\TOTO\fi
\next}\TOTO}
```

Commentaires : le #1 qui suit `\toto` n'est pas utilisé dans la suite. Il s'agit simplement d'une ruse classique qui oblige T_EX à « consommer » le crochet ouvrant qui doit obligatoirement suivre `\toto`. Les double-dièses ##1 qu'utilise `\TOTO` sont nécessaires puisque maintenant la définition de `\TOTO` figure à l'intérieur de la définition de `\toto`. Vérifions que tout se passe bien :

```

$\toto[]$ ..... ()
$\toto[1]$ ..... (1)
$\toto[1,2]$ ..... (1(2))
```

```

$\toto[1,22,333]$ ..... (1(22(333)))
```

```

$\toto[\alpha_1,\beta_2,\gamma_3]$
..... (\alpha_1(\beta_2(\gamma_3)))
```

Maintenant que nous avons notre macro, il n'est pas difficile de passer au codage d'une fraction continue. Ce que nous voulons ici est une macro `\fraccont` telle que les codages

```

$x=\fraccont[\alpha_0,\alpha_1,\alpha_2]$ et $y=\fraccont[\alpha,\beta,\gamma,\delta,\varepsilon]$
```

nous fabrique automatiquement les fractions continues

$$x = \alpha_0 + \frac{1}{\alpha_1 + \frac{1}{\alpha_2}}$$

$$y = \alpha + \frac{1}{\beta + \frac{1}{\gamma + \frac{1}{\delta + \frac{1}{\varepsilon}}}}$$

Pour cela, il suffit de modifier un tout petit peu la macro précédente. Chaque fois que nous rencontrerons une virgule, nous ajouterons `{1\over` à `\tete` et `}` à `\queue` à la place des parenthèses ouvrantes et fermantes de l'exemple précédent.

```

\def\fraccont#1{\def\tete{%
\group\displaystyle}
\def\queue{\egroup}
\let\oldstrut=\strut\def\strut{}
\def\FRACCONT##1{\ifx##1
\def\next{\tete\queue}
\else\ifx##1,
\edef\tete{\tete+\bgroup%
```

```
\strut1\over\displaystyle}  
\edef\queue{\queue\egroup}  
\else\edef\tete{\tete##1}\fi  
\let\next=\FRACONT\fi  
\next}  
\let\strut=\oldstrut\FRACONT}
```

Notez les réglages indispensables pour un meilleur rendu typographique (les `\displaystyle`, trop souvent oubliés par les débutants, qui permettent d'obtenir des gros caractères et le `\strut` pour éloigner les '1' du plafond constitué par les traits de fractions). Notez enfin les acrobaties nécessaires pour utiliser un `\strut` dans une `\edef` (`\let\oldstrut=\strut\def\strut{}` suivi de `\let\strut=\oldstrut`). C'est désagréable, mais \TeX n'aime pas les `\strut` dans une `\edef`.

3. Question à des gourous

L'édition de textes pour les sciences humaines est une vraie source de problèmes que pratiquement aucun système de PAO ne résoud pleinement. En particulier les historiens des textes ont des besoins très forts de deux choses qu'aucun système WYSIWYG ne permet :

- Numérotation des lignes d'un texte (par exemple toutes les 5 lignes). Ceci a déjà été fait en \TeX (voir par exemple *TUGBoat*, vol. 5, num. 1, May 1984, p.25).
- Utilisation de plusieurs types de notes. Dans la publication d'un cartulaire, l'usage veut que l'on sépare les notes d'apparat critique (numérotées par exemple *a*, *b*) et les notes historiques (numérotées par exemple *1*, *2*). On en trouvera, par exemple,

un modèle dans *Le Médiéviste et l'ordinateur*, num. XVI, p.19.

Question : Existe-t-il un gourou capable décrire les macros \TeX ou \LaTeX pour faire ceci? Écrire à la rédaction qui transmettra.

Une date à retenir :

15–16 Mai
1990

Toulouse

Journées GUTenberg :
 \TeX &
Les Industries des Arts Graphiques

Renseignements :
Association GUTenberg
BP 21
F-78350 Jouy-en Josas
France