

# Formation T<sub>E</sub>X : plain T<sub>E</sub>X

Denis ROEGEL

[roegel@loria.fr](mailto:roegel@loria.fr)

1999

1	Exemple simple en <i>plain</i> T <sub>E</sub> X . . . . .	2
2	Premières observations . . . . .	4
3	Ligatures automatiques . . . . .	5
4	T <sub>E</sub> X et <i>plain</i> T <sub>E</sub> X . . . . .	6
5	Commandes . . . . .	7
6	Structuration d'un document source . . . . .	8
7	Divers . . . . .	9
8	Redéfinition de commandes par L <sup>A</sup> T <sub>E</sub> X . . . . .	10
9	Correction italique . . . . .	11
10	Changement de police de bas niveau . . . . .	12
11	Les groupes . . . . .	14
12	Les caractères spéciaux . . . . .	15
13	Affichage de caractères spéciaux . . . . .	17
14	Analyse lexicale . . . . .	18
15	Dimensions . . . . .	19
16	Boîtes . . . . .	20
17	Espacements flexibles (ressorts) . . . . .	23
	1 Généralités . . . . .	23
	2 Arithmétique des ressorts . . . . .	25
	3 Ressorts infinis . . . . .	26
	4 Ressorts doublements infinis . . . . .	26

5	Pavages ou continuations ( <i>leaders</i> ) . . . . .	27
18	Mode horizontal, vertical, mathématique . . . . .	28
19	Découpage des paragraphes en lignes . . . . .	29
1	Justification . . . . .	29
2	Indentation . . . . .	30
3	E spacements . . . . .	30
4	Pénalités . . . . .	31
20	Rassemblement de lignes en pages . . . . .	33
21	Macros . . . . .	34
22	Macros du T <sub>E</sub> Xbook . . . . .	35

## 1. Exemple simple en plain TeX

English words like 'technology' stem from a Greek root beginning with the letters `\tau\epsilon\psi\iota\lambda\omicron\chi\iota\ldots\,\$`; and this same Greek word means `{\sl art\}` as well as technology. Hence the name `\TeX`, which is an uppercase form of `\tau\epsilon\psi\iota\lambda\omicron\chi\iota\$`.

Insiders pronounce the `\chi` of `\TeX` as a Greek chi, not as an 'x', so that `\TeX` rhymes with the word `blecchh`. It's the 'ch' sound in Scottish words like `{\sl loch\}` or German words like `{\sl ach\}`; it's a Spanish 'j' and a Russian 'kh'. When you say it correctly to your computer, the terminal may become slightly moist.

The purpose of this pronunciation exercise is to remind you that `\TeX` is primarily concerned with high-quality technical manuscripts: Its emphasis is on art and technology, as in the underlying Greek word. If you merely want to produce a passably good document---something acceptable and basically readable but not really beautiful---a simpler system will usually suffice. With `\TeX` the goal is to produce the `{\sl finest\}` quality; this requires more attention to detail, but you will not find it much harder to go the extra distance, and you'll be able to take special pride in the finished product.

`\bye`

(**extrait du TeXbook**)

**Exercice :**

*Compiler le fichier précédent.*

## 2. Premières observations

- Texte plus une commande de fin (`\bye`) ;
- `$...$` pour entrer dans le mode mathématique ;
- Commandes : `\tau`, `\epsilon`, `\chi` (lettres grecques), `\ldots` (ellipse), `\,` (petite espace), `\sl` (changement de police), `\/` (correction italique), `\TeX` (logo  $\text{T}_{\text{E}}\text{X}$ ), `\`, `\bye` (fin) ;
- Groupes délimitant une action : `{...}` ;
- Tirets : `-`, `--`, `---` ;
- Espaces multiples ignorés ;
- Changement d’alinéa par une ligne blanche.

### **3. Ligatures automatiques**

En écrivant `fi`, on obtient « fi ».

Les ligatures (ff, fi, etc.) et crénages sont définies dans les polices. Des polices existantes peuvent être modifiées grâce au concept de polices virtuelles.

#### 4. T<sub>E</sub>X et *plain* T<sub>E</sub>X

- T<sub>E</sub>X est le moteur, c'est-à-dire le programme qui est effectivement exécuté ; c'est un programme écrit en PASCAL. Le moteur est accessible avec la commande `virtex`.
- *plain* T<sub>E</sub>X (souvent appelé T<sub>E</sub>X) est une couche au-dessus de T<sub>E</sub>X écrite en T<sub>E</sub>X. C'est ce que l'on obtient avec la commande `tex`. Il y environ 600 commandes dans *plain* T<sub>E</sub>X, qui se rajoutent aux 300 commandes de T<sub>E</sub>X.
- L<sup>A</sup>T<sub>E</sub>X est essentiellement une couche au-dessus de *plain* T<sub>E</sub>X.



## 5. Commandes

Les commandes sont de deux natures :

1. Caractère de contrôle (*control character*) : un `\` suivi d'un caractère qui n'est pas une lettre. Exemples : `\,`, `\\`, `\&`, etc.
2. Commandes de contrôle (*control commands*) : un `\` suivi d'une ou plusieurs lettres. Le nom de la commande s'achève avec la dernière lettre. Exemple : `\TeX`, `\bye`, `\sl`, etc.

Les espaces sont ignorés après une commande de contrôle, mais pas après un caractère de contrôle.

## 6. Structuration d'un document source

Un gros document est en général scindé en plusieurs fichiers. Il y a un fichier principal (sur lequel est appelé `tex`) et des fichiers inclus par `\input`.

Exemple :

```
% fichier principal
\input preface.tex
\input chapitre1.tex
\input chapitre2.tex
...
\bye
```

Remarques :

- L'extension `.tex` est implicite ;
- Avec  $\LaTeX$ , la syntaxe est `\input{fichier}`.
- $\LaTeX$  possède aussi une commande `\include`.

## 7. Divers

- `\show` permet de voir la définition d'une commande (et voir en passant si c'est une primitive). Exemples :

```
*\show\bye
```

```
> \bye=\outer macro:
```

```
->\par \vfill \supereject \end .
```

```
*\show\TeX
```

```
> \TeX=macro:
```

```
->T\kern -.1667em\lower .5ex\hbox {E}\kern -.125emX.
```

```
*\show\string
```

```
> \string=\string.
```

## 8. Redéfinition de commandes par $\LaTeX$

La plupart des commandes de *plain*  $\TeX$  fonctionnent en  $\LaTeX$ , mais pas nécessairement toutes de la même manière.  $\LaTeX$  modifie certaines commandes (par exemple `\input`) et en supprime certaines. En principe, un utilisateur de  $\LaTeX$  normal (pas un concepteur de classes) ne devrait utiliser que les commandes documentées dans le manuel de  $\LaTeX$  de Lamport. Mais ce n'est pas toujours possible.

## 9. Correction italique

Lorsqu'un caractère italique est suivi d'un caractère romain, il peut y avoir collision ou bien tout simplement un effet désagréable.

Exemple :

***/b*** { \it l } b

L'ajout de la correction italique (avec `\/`) après le dernier caractère penché résout ce problème :

***/b*** { \it l \/ } b

Avec  $\text{\LaTeX}$ , les commandes `\textit`, etc., intègrent déjà la correction italique.

## 10. Changement de police de bas niveau

T<sub>E</sub>X ne fait que positionner des caractères, mais il ne les affiche pas (c'est le travail de `dvips`, `xdvi`, etc.). T<sub>E</sub>X a uniquement besoin des dimensions des caractères et celles-ci se trouvent dans les fichiers `.tfm` (*T<sub>E</sub>X Font Metrics*). Pour faire utiliser une nouvelle police par T<sub>E</sub>X, il faut donc en principe uniquement obtenir ou créer le ou les fichiers `.tfm` adaptés.

Lorsqu'on parle de police en T<sub>E</sub>X, c'est donc en général un abus de langage pour « fichier `.tfm` ». La visualisation des caractères se fait en dehors de T<sub>E</sub>X.

Pour accéder à une police, disons `cmr10` (en fait `cmr10.tfm`), T<sub>E</sub>X utilise la commande `\font` :

```
\font\test=cmr10
{\test texte en cmr10}
```

Ces commandes sont de bas niveau et donc très rigides. Il n'est pas facile de passer facilement d'une police à son équivalent gras, etc.

D'un autre côté, en bonne typographie il ne faut pas chercher à utiliser toutes les polices disponibles.

L<sup>A</sup>T<sub>E</sub>X offre davantage de souplesse.

La commande `\char` permet d'accéder à un caractère d'une police. Par exemple, `\char98` donne le caractère 98 de la police courante, en général un « b ».

Certaines commandes sont définies avec `\char`.

`\%`, par exemple, est défini comme étant égal à `\char'\%`, ce qui signifie en fait le caractère de la police courante à la position indiquée par le code de %, lequel est donné avec `'\%`.

## 11. Les groupes

Un groupe (indiqué par « { » et « } ») est une construction permettant de restreindre l'effet de certaines commandes. Exemple :

```
texte romain {\sl texte pench\'e\/} texte romain
```

Les commandes `\beginngroup` et `\endgroup` peuvent aussi être utilisées.

On ne peut pas écrire :

```
{ \beginngroup ... } \endgroup
```

(il ne doit pas y avoir de chevauchement)



## 12. Les caractères spéciaux

Les caractères du fichier que T<sub>E</sub>X va voir se répartissent en catégories. La catégorie usuelle est celle des *lettres* (A,...,Z,a,...,z). La plupart des autres caractères (par exemple les chiffres ou la ponctuation) rentrent dans la catégorie *autre*. Lorsque les lettres ou les caractères de la catégorie *autre* font partie d'un texte, elles font apparaître les caractères auxquels on s'attend. Les 14 autres catégories ont un comportement différent.

Au début d'une exécution de T<sub>E</sub>X, les caractères sont répartis dans certaines catégories, mais on peut changer la catégorie d'un caractère en cours de route, et donc faire en sorte que certains caractères se comportent comme d'autres caractères. En général, c'est dangereux et à éviter par les débutants.

- Catégorie 0 : échappement (exemple : \)
- Catégorie 1 : début de groupe (exemple : {)
- Catégorie 2 : fin de groupe (exemple : })
- Catégorie 3 : passage ou sortie du mode mathématique (exemple : \$)
- Catégorie 4 : alignement (exemple : &)
- Catégorie 5 : fin de ligne
- Catégorie 6 : paramètre (exemple : #)
- Catégorie 7 : exposant (exemple : ^)
- Catégorie 8 : indice (exemple : \_)
- Catégorie 9 : caractère ignoré
- Catégorie 10 : espace
- Catégorie 11 : lettre
- Catégorie 12 : autre caractère
- Catégorie 13 : caractère actif (exemple : ~)
- Catégorie 14 : commentaire (exemple : %)
- Catégorie 15 : caractère non valide

### 13. Affichage de caractères spéciaux

Aux caractères spéciaux standard correspondent des commandes permettant de les afficher :

- `\backslash` produit `\` ; on peut aussi utiliser `{\tt \char`\\}`
- `{\tt $\{$}` pour `{`
- `{\tt $\}$}` pour `}`
- `\$` pour `$`
- `\&` pour `&`
- `\#` pour `#`
- `{\tt \char`^}` pour `^`
- `\_` pour `_`
- `{\tt \char`~}` pour `~`
- `\%` pour `%`

$\LaTeX$  offre d'autres commandes pour accéder à ces caractères.

## 14. Analyse lexicale

Les codes des caractères déterminent comment le texte d'un fichier est compris par T<sub>E</sub>X et plus précisément comment se fait l'analyse lexicale (identification des *unités* du texte).

L'analyse lexicale produit des *tokens*. Il y a deux sortes de *tokens* : ceux qui correspondent à des caractères et ceux qui correspondent à des commandes. Seuls ceux qui correspondent à des caractères appartiennent à l'une des 16 catégories précitées.

```
{\hskip 36 pt}
```

produit les *tokens* suivants :

```
{1 hskip 312 612 ␣10 p11 t11 }2
```

## 15. Dimensions

- pt point
- pc pica (1pc=12pt)
- in pouce (1in=72.27pt)
- bp gros point (72bp=1in)
- cm centimètre (2.54cm=1in)
- mm millimètre (10mm=1cm)
- dd point didot (1157dd=1238pt)
- cc cicéro (1cc=12dd)
- sp *scaled point* (65536sp=1pt)

Ces dimensions peuvent être utilisées pour ajouter des espaces ou régler la taille de certains objets. Pour introduire un blanc (horizontal ou vertical) de 3pc, il suffit d'écrire : `\kern 3pc`. Le fait que le blanc est horizontal ou non dépend du mode dans lequel on se trouve.

Des dimensions négatives peuvent être employées.

## 16. Boîtes

Une boîte est une structure permettant de contenir des données affichables. Une boîte peut contenir des caractères, des lignes, d'autres boîtes, des espacements, etc. Lorsque des objets sont mis en boîte, ils forment au sein de celle-ci une structure rigide.

Il y a deux grandes sortes de boîtes : les boîtes horizontales et les boîtes verticales. La différence tient dans la manière avec laquelle la boîte est remplie.

Exemples de création de boîte horizontale :

```
\hbox{ABC} \hbox{A\hbox{BC} D}
```

Une boîte est affichée avec les commandes précédentes et le résultat est le suivant :

ABC      ABC D

Exemples de création de boîte verticale :

```
X \vbox{\hbox{A}\hbox{B}}  
  \vbox{\hbox{A}\hbox{BC}\hbox{D}}
```

L'affichage donne :

          A  
          A   BC  
          X B   D

La boîte verticale est donc un empilement vertical, alors que la boîte horizontale est un enfilement horizontal.

Une boîte peut être stockée dans une variable et réutilisée ensuite :

```
\newbox\boite
```

```
\setbox\boite=\hbox{A\vbox{\hbox{B}\hbox{C}} D}
```

(Commandes  $\text{\LaTeX}$  : `\newsavebox`, `\savebox`, `\sbox`)

Cette boîte est utilisée avec `\box\boite` ou `\copy\boite`. `\box` vide la boîte et ne permet pas de la réutiliser (ou afficher) plusieurs fois, ce que permet `\copy`.

`\vtop` est une variante de `\vbox` où l'alignement se fait avec la première ligne :

```
X \vtop{\hbox{A}\hbox{B}\hbox{C}}
   \vtop{\hbox{A}\hbox{BC}\hbox{D}}
```

donne

```
 X A   A
   B   BC
   C   D
```

Le contenu d'une boîte peut être examiné avec `\showbox`. Si l'on fait :

```
\newbox\test
\setbox\test=\hbox{\TeX}
\showboxdepth\maxdimen
\showboxbreadth\maxdimen
\showbox\test
```

on obtient :

```
\hbox(6.83331+2.15277)x18.6108
.\tenrm T
.\kern -1.66702
.\hbox(6.83331+0.0)x6.80557, shifted 2.15277
..\tenrm E
```

```
.\kern -1.25  
.\tenrm X
```

Les caractères sont aussi des boîtes particulières. Comme elles, elles ont trois dimensions : largeur, hauteur, profondeur. Un caractère a toutefois encore d'autres attributs, comme la correction italique.

Une page est aussi une boîte verticale.

Au fur et à mesure qu'une page est construite, elle est accumulée dans `\box255`.

Boîtes noires, filets : `\hrule` et `\vrule`.



## 17. Espacements flexibles (ressorts)

### 17.1. Généralités

Dans le langage de T<sub>E</sub>X : *glue*.

Un ressort peut être vu comme un objet ayant une dimension au repos et pouvant être compressé ou étiré en fonction des contraintes de son environnement. Ce sont en d'autres termes des blancs de dimension variable.

Intérêt : résolution du problème de l'alignement. Ainsi, tous les blancs inter-mots sont en fait de petits ressorts et ceux-ci sont compressés ou étirés pour permettre la justification. Le blanc inter-mots est un ressort implicite, défini dans la police courante de T<sub>E</sub>X.

Des ressorts peuvent être utilisés explicitement :

```
\hbox to 10cm{A\kern2cm B}
```

produit un message « Underfull \hbox » parce que ce que l'on a demandé est impossible :

A            B

Pour mettre A et B aux deux extrémités d'une distance de 10cm, on peut mettre un vrai ressort, de dimension au repos de 5cm, pouvant s'allonger de 5cm et pouvant se raccourcir de 5cm :

```
\hbox to 10cm{A\hskip5cm plus 5cm minus 5cm B}
```

ce qui donne :

A B

et pas d'erreur.

Là encore, en écrivant :

```
\hbox to 10cm{A\hskip5cm plus 1cm minus 5cm B}
```

cela va passer (et sans erreur !).

A B

On ne peut donc jamais vraiment limiter l'extensibilité. Par contre, il n'y a pas indéfiniment de compressibilité :

```
\hbox to 2cm{A\hskip5cm plus 5cm minus 0cm B}
```

donne

A B

et un message `Overfull \hbox.`





## 17.5. Pavages ou continuations (*leaders*)

Avec une boîte et du blanc (extensible ou non), on peut faire un pavage de ce blanc avec cette boîte. Exemples :

```
*\leaders\hbox{\TeX}\hskip5cm*
```

qui donne :

```
* TEXTTEXTTEXTTEXT *
```

```
*\leaders\hbox{.}\hskip5cm*
```

donne :

```
* ..... *
```

C'est le principe des lignes pointillées dans les tables des matières.

```
\hbox to 7cm{A\leaders\hbox{*}\hfil  
B\leaders\hbox{*}\hfil C}
```

donne :

```
A*****B*****C
```

## 18. Mode horizontal, vertical, mathématique

Il y a trois principaux modes dans lesquels peut se trouver T<sub>E</sub>X à un moment donné.

1. Mode horizontal : c'est le mode le plus fréquent, celui correspondant à du texte dans un alinéa. Le mode horizontal commence dès qu'un caractère ou certaines commandes propres à ce mode sont rencontrées (`\indent`, `\noindent`, `\vrule`, `\hskip`, `\leavevmode`, etc.)
2. Mode vertical : c'est le mode dans lequel on se trouve au début de la compilation et entre les alinéas. Certaines commandes font revenir au mode vertical (`\par`, `\hrule`, `\vskip`, etc.).
3. Mode mathématique : on passe à ce mode depuis le mode horizontal ou vertical.

## 19. Découpage des paragraphes en lignes

Un seul alinéa (*paragraph* en anglais) est traité à la fois par T<sub>E</sub>X avant qu'il ne soit mis en forme et justifié. Un alinéa est traité globalement ce qui permet d'égaliser les espaces excédentaires ou manquants sur l'ensemble de l'alinéa pour obtenir un résultat supérieur à un traitement ligne par ligne.

Un alinéa est en général un morceau de texte entre deux lignes blanches. Une commande `\par` force une fin d'alinéa.

### 19.1. Justification

La justification d'un alinéa n'est faite que par *modification des espaces inter-mots* ou par *l'emploi de coupures*. En aucun cas n'est-il fait d'interlettrage (espacement dans un mot).

Le texte est justifié sur la largeur courante du texte qui est `\hsize`.

Un alinéa est *toujours* justifié, mais il est possible d'insérer des ressorts à droite ou à gauche de chaque ligne, ce qui peut donner un effet non justifié si on le souhaite.

Ces ressorts de gauche et de droite sont `\leftskip` et `\rightskip`. La commande `\raggedright` qui correspond à un texte non justifié à droite modifie la valeur de `\rightskip`.

## 19.2. Indentation

Un alinéa est indenté en outre de `\parindent` qui est une dimension pouvant être modifiée. À la fin d'un alinéa,  $\text{T}_{\text{E}}\text{X}$  ajoute un ressort `\parfillskip` qui évite ainsi de justifier la dernière ligne. En donnant une valeur différente à `\parfillskip`, on peut forcer la justification de la dernière ligne ou l'indenter à droite d'un montant quelconque.

`\hangindent` et `\hangafter` sont en général utilisés pour modifier la forme de l'indentation initiale qui peut être faite sur plusieurs lignes, par exemple pour y insérer une lettrine ou une figure. Des formes plus complexes peuvent être données à un alinéa avec `\parshape`. Le  $\text{T}_{\text{E}}\text{X}$ book donne par exemple un alinéa en forme de disque.

## 19.3. Espacements

Les espacements après la ponctuation peuvent être rendus tous égaux par la commande `\frenchspacing`. `\nonfrenchspacing` produit des espaces plus longues après les ponctuations qu'entre les mots. Ces commandes fonctionnent en modifiant une propriété des caractères de ponctuation appelée le facteur d'espace (*space factor*).



## 19.4. Pénalités

La mise en forme d'un alinéa peut encore être modulée par des « pénalités ». Si la pénalité est forte, une coupure sera sanctionnée. Si elle est faible, une coupure pourra être encouragée. La valeur 10000 interdit une coupure. La valeur  $-10000$  la force. Les pénalités peuvent être indiquées directement avec `\penalty`, par exemple `\penalty100`, ou bien par des commandes comme `\break`, `\nobreak` (`\pagebreak` en  $\LaTeX$ ). En mode horizontal, les pénalités vont agir sur les coupures de ligne alors qu'en mode vertical elles vont agir sur les coupures de page.

Lorsqu'un alinéa est mis en forme,  $\TeX$  se base sur un certain critère de qualité et va essayer de produire un alinéa dont la qualité est la meilleure possible.  $\TeX$  essaie d'abord de faire une mise en page sans coupure et son critère de qualité est `\pretolerance`. Si la mise en page échoue,  $\TeX$  essaie à nouveau en faisant des coupures et avec pour nouvel objectif de qualité `\tolerance`. Si `\pretolerance` est négatif, la première étape sera sautée.

Des pénalités pour certains cas particuliers peuvent être ajoutées :

- `\hyphenpenalty` : coupure
- `\adjdemerits` : lignes adjacentes visuellement incompatibles
- `\doublehyphendemerits` : deux coupures consécutives
- etc.

On peut forcer un alinéa à avoir une ligne de plus que sa taille optimale en indiquant `\looseness=1` au début de l'alinéa.

Les messages « `Underfull \hbox` » peuvent être contrôlés en modifiant `\hbadness`. Mais attention, ceci ne supprime pas les mauvaises lignes, seulement les messages !

Les messages « `Overfull \hbox` » peuvent être contrôlés en modifiant `\hfuzz` qui est le dépassement autorisé sur une ligne.

Enfin, une troisième passe sur l'alinéa est possible si les deux premières passes ont échoué et si `\emergencystretch` est positif.

## 20. Rassemblement de lignes en pages

Entre les alinéas est ajouté le ressort `\parskip`.

Le choix de la coupure de page se fait aussi en fonction de pénalités. Il y a une pénalité particulière pour la coupure entre la première et la deuxième ligne d'un alinéa : `\clubpenalty`. Entre l'avant-dernière et la dernière, c'est `\widowpenalty`. Ceci permet donc d'interdire les veuves et les orphelins.

Les messages « `Underfull \vbox` » peuvent être contrôlés en modifiant `\vbadness`. Mais attention, ceci ne supprime pas les mauvaises pages, seulement les messages !

Les messages « `Overfull \vbox` » peuvent être contrôlés en modifiant `\vfuzz` qui est le dépassement autorisé sur une page.

La page formée se trouve dans la boîte 255 et la routine de sortie (`\output`) est appelée.

## **21. Macros**

Voir article « Anatomie d'une macro ».

## 22. Macros du T<sub>E</sub>Xbook

Étude de certaines parties du fichier `manmac.tex` qui sont les macros de Knuth pour mettre en page le T<sub>E</sub>Xbook.