
LEÇON 1

Où l'on parle de source ...

Les tris leçons qui suivent s'adressent aux débutants absolus ou bien aux lecteurs ayant déjà une expérience de $\text{T}_{\text{E}}\text{X}$ lui-même et souhaitant passer à $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$. Nous allons d'abord, en une dizaine de pages, un aperçu aussi complet que possible des capacités de $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ et de son mode de fonctionnement. Le premier contact est parfois déroutant mais nous verrons qu'en réalité, il suffit de très peu de temps pour s'habituer aux particularités de $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$. Après lecture de ces quelques pages, chacun sera déjà en mesure de préparer des documents complets. Les bases sont simples et le reste est affaire d'expérience.

À quoi sert $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$?

Tout simplement à créer des documents de la plus haute qualité typographique en libérant l'auteur de toutes les tâches de mise en pages et de construction : ces tâches peuvent aller de la simple structuration du document en chapitres, sections, paragraphes à la construction d'index et de tables des matières, à la tenue à jour des numérotations et des références croisées, la gestion de bibliographies, la construction de tableaux, l'inclusion de graphiques, la compilation de formules mathématiques, l'insertion de textes multilingues, etc.

Le principe de base consiste à donner des ordres à $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ au sein même du texte pour lui indiquer ce que l'on souhaite : $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ exécute les ordres qu'on lui donne et produit un document mis en pages.

Toutes ces notions vont être abordées sur un exemple concret. $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ étant aussi utilisé pour la construction de textes scientifiques que littéraires, nous allons nous mettre dans la situation d'un professeur de littérature préparant une édition critique des traités scientifiques de Blaise Pascal. Nous commencerons avec le *Traité sur la roulette*. La première chose à faire sera de créer un fichier que nous appellerons `roulette.tex`. Ce fichier constituera ce que l'on appelle habituellement le fichier *source* : c'est lui qui contient le texte que l'on veut éditer. Pour rédiger le fichier source, il faut être en possession d'un *éditeur de texte* : il s'agit d'un programme permettant de rédiger et de créer des fichiers au format *texte*, c'est-à-dire contenant exclusivement le texte qu'on a saisi, sans y ajouter des instructions cachées comme le font la plupart des traitements de texte. Il en existe d'excellents sur toutes les plates-formes : en particulier emacs sous UNIX, ou bien Alpha sous MACOS qui possède un mode spécialement dédié à $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$. L'usage d'un traitement de texte est déconseillé : si l'on n'a pas d'autres possibilités, il faudra veiller à sauvegarder le fichier en format texte.

Exécution d'un fichier source

Voici quel sera, pour commencer, le contenu de notre fichier `roulette.tex` :

```
\documentclass[a4paper]{article}
\begin{document}
```

La roulette est une ligne si commune, qu'après la droite et la circulaire, il n'y en a point de si fréquente ; et elle se décrit si souvent aux yeux de tout le monde qu'il y a lieu de s'étonner qu'elle n'ait point été considérée par les anciens, dans lesquels on n'en trouve rien : car ce n'est autre chose que le chemin que fait en l'air le clou d'une roue, quand elle roule de son mouvement ordinaire, depuis que ce clou commence à s'élever de terre, jusqu'à ce que le roulement continu de la roue l'ait rapporté à terre, après un tour entier achevé.

```
\end{document}
```

Cet exemple nous montre la structure adoptée par tout document source destiné à L^AT_EX : une première ligne contient ce que l'on appelle une déclaration de classe. Nous déclarons ici un document de classe *article* qui adoptera le format d'une feuille de papier A4. Tous les mots précédés par une crochets-blique \ sont des commandes : ce sont donc les ordres que l'on place dans le document et que L^AT_EX doit comprendre, interpréter et exécuter. La commande `\begin{document}` indique où commence effectivement le document. Ce dernier s'achève impérativement avec une commande `\end{document}`. Une fois que le fichier `roulette.tex` est prêt, il faut le faire exécuter par L^AT_EX. Suivant les systèmes, on devra écrire une commande telle que

```
latex roulette
```

ou bien lancer le programme T_EX et demander l'exécution à partir d'un article de menu qui pourrait s'appeler `Tex ...` ou `Typeset ...` ou `Exécuter ...`, etc. Chacun devra consulter la documentation de sa implémentation de T_EX. Il faut savoir en effet que L^AT_EX n'est en réalité qu'une forme dérivée de T_EX : lorsque l'on donne un fichier à exécuter à L^AT_EX, c'est en réalité T_EX qui fait le travail. Il va lire la signification des commandes qu'il rencontre dans ce que l'on appelle un *format* : L^AT_EX est en fait un *format* pour T_EX. Il existe souvent un menu, dans le programme T_EX, qui permet de choisir le format souhaité. Dans notre cas ce sera toujours `latex`. Ne confondez pas, d'autre part, le programme T_EX avec l'éditeur de texte qui vous aura permis de créer le fichier `roulette.tex`. Ce sont des programmes distincts et on passe sans cesse de l'un à l'autre. Le programme TeXtures sur Macintosh est un cas un peu particulier qui réunit les deux programmes en un.

L^AT_EX, après exécution de `roulette.tex`, produit un fichier nommé `roulette.dvi`. C'est ce fichier qui contient le document mis en forme et prêt à l'édition. Là encore, il peut y avoir de petites différences suivant les systèmes mais le principe reste toujours le même : on doit visualiser le fichier DVI pour juger du résultat. Cela se fait soit à partir d'un article de menu *ad hoc*, soit en appelant un programme dit prévisualisateur de DVI comme `xdvi`, `dvipreview` etc. On verra alors à l'écran notre texte comme ceci :

La Roulette est une ligne si commune, qu'après la droite et la circulaire, il n'y en a point de si fréquente ; et elle se décrit si souvent aux yeux de tout le monde qu'il y a lieu de s'étonner qu'elle n'ait point été considérée par les anciens, dans lesquels on n'en trouve rien : car ce n'est autre chose que le chemin que fait en l'air le clou d'une roue, quand elle roule de son mouvement ordinaire, depuis que ce clou commence à s'élever de terre, jusqu'à ce que le roulement continu de la roue l'ait rapporté à terre, après un tour entier achevé.

Le résultat produit est un paragraphe parfaitement assemblé dont les lignes sont justifiées à gauche et à droite. On peut d'ores et déjà noter plusieurs points importants :

- le texte source peut être saisi sans se préoccuper le moins du monde des coupures de lignes car c'est L^AT_EX qui s'en charge.
- on n'a pas à se préoccuper non plus de l'espace entre les mots : celui-ci est *calculé* automatiquement par L^AT_EX de façon à obtenir un paragraphe équilibré au mieux. Il n'y a pas de blanc fixe : l'espace entre les mots peut être amené à varier effectivement dans certaines circonstances.
- certains mots peuvent être coupés en fin de ligne par L^AT_EX.
- il se peut que le résultat que vous obtenez ne soit pas tout à fait identique au texte ci-dessus : cela dépend évidemment des marges de votre document. Il se peut aussi que les lettres accentuées aient été supprimées si votre installation n'est pas encore adaptée à la langue française : le problème des accents sera abordé par la suite et peut être résolu de diverses manières. Nous ne nous en préoccupons pas pour le moment. C.f. § 3.5.1.

La notion de commande

Nous allons voir maintenant comment introduire des commandes dans le texte. Les commandes, aussi appelées macros, sont toujours préfixées par une contre-blique `\` : c'est comme ça que L^AT_EX les distingue du reste du texte. Par exemple, L^AT_EX ne tenant pas compte des fins de ligne dans le document source, il doit exister des commandes permettant d'aller à la ligne : le plus simple est d'utiliser la commande `\par` qui signifie « fin de paragraphe ». Lorsqu'il la rencontre, L^AT_EX compresse le paragraphe qu'il était en train de lire puis commence un nouveau dès qu'il rencontre le mot suivant. Voici l'exemple suivant : la colonne de gauche contient le texte source et celle de droite ce que l'on obtient après exécution par L^AT_EX :

```
\noindent Le feu P. Mersenne proposa
donc la recherche de la nature de cette
ligne à tous ceux de l'Europe qu'il en
crût capables, et entre autres à
Galilée. Mais aucun n'y put réussir,
et tous en désespérèrent.\par\noindent
Plusieurs années se passèrent de cette
sorte jusqu'en 1694.
```

```
Le feu P. Mersenne proposa donc
la recherche de la nature de cette
ligne à tous ceux de l'Europe qu'il
en crût capables, et entre autres à
Galilée. Mais aucun n'y put réus-
sir, et tous en désespérèrent.
Plusieurs années se passèrent de
cette sorte jusqu'en 1694.
```

La commande `\noindent` sert à empêcher l'indentation des paragraphes. Signalons qu'une autre façon de passer à la ligne consiste à laisser une ligne blanche dans le fichier source : autrement dit, deux sauts à la ligne dans le fichier source provoquent un passage à la ligne dans le document produit.

Avec L^AT_EX, tout fonctionnellement au moyen de commandes qui émaillent ainsi le texte : certaines commandes ont un effet permanent comme les deux que nous venons de mentionner, d'autres un effet plus durable. Par exemple, la commande `\large` permet de passer à un corps de caractères un peu plus grand. Si ce changement de corps ne concerne qu'une partie du texte, il faudra la délimiter au moyen d'accolades, ce qui nous conduit à la notation de groupe ou de zone : si une commande est placée dans une

régi n délimitée par une paire d'acc lades { et }, sa p rtée sera limitée à cette z ne, sin n elle restera valable jusqu'à ce que l' n renc ntre une n uvelle macr définissant un autre réglage. Aj ut ns ainsi un titre à n tre texte :

<pre>{\large HISTOIRE DE LA ROULETTE} \par appelée autrement trochoïde ou cycloïde \par {\small où l'on rapporte par quels degrés on est arrivé à la connaissance de la nature de cette ligne.} \par La Roulette est une ligne\dots.</pre>	<p>HISTOIRE DE LA ROULETTE appelée autrement tr ch ïde u cycl ïde ù l' n rapp rte par quels degrés n est arrivé à la c nnaissance de la nature de cette ligne. La R ulette est une ligne ...</p>
--	--

La notion d'environnement

La n ti n d'envir nnement vient c mpléter celle de gr upe délimité par des acc lades : un envir nnement délimite une régi n dans laquelle ser nt appliquées certaines règles particulières. Il en existe de très variés. Un envir nnement c mmence t uj urs par une c mmande `\begin` et se termine par une c mmande `\end` : ces deux c mmandes s nt suivies du n m de l'envir nnement entre acc lades. Si, par exemple, n us v uli ns centrer n tre titre, il suffirait de le placer dans un envir nnement `\begin{center}` ...`\end{center}` c mme ceci :

```
\begin{center}
{\large HISTOIRE DE LA ROULETTE}
\par appelée autrement trochoïde ou cycloïde
\par {\small où l'on rapporte par quels degrés on est arrivé
à la connaissance de la nature de cette ligne.}
\end{center}
\par La Roulette est une ligne si commune, qu'après la droite et la
circulaire, il n'y en a point de si fréquente ;
```

Ce qui d nnerait à présent :

```

HISTOIRE DE LA ROULETTE
appelée autrement tr ch ïde u cycl ïde
ù l' n rapp rte par quels degrés n est arrivé à la c nnaissance de la nature
de cette ligne.
La R ulette est une ligne si c mmune, qu'après la dr ite et la circulaire,
il n'y en a p int de si fréquente ;
```

Les envir nnements f ncti nnent c mme les acc lades du p int de vue de la p rtée des macr s : une macr placée à l'*intérieur* d'un envir nnement sera valable seulement jusqu'à la fin de cet envir nnement. En revanche, une macr placée *avant* le début d'un envir nnement sera valable en principe aussi à l'intérieur.

Les envir nnements peuvent être imbriqués les uns dans les autres mais ne d ivent jamais être entrecr isés :

<pre>\begin{center} \begin{figure} \end{figure} \end{center}</pre>	est correct mais	<pre>\begin{center} \begin{figure} \end{center} \end{figure}</pre>	ne l'est pas.
--	------------------	--	---------------

La notion d'argument

Les macros sont parfois indispensables, pour fonctionner, de connaître certains renseignements. Par exemple, la macro `\hspace` est utilisée pour créer un espace blanc horizontal mais il faut évidemment lui indiquer la dimension de l'espace souhaité. Elle attend donc qu'on lui passe une valeur qui constituera ce qu'on appelle son *argument*.

Si on écrit par exemple `a\hspace{15mm}b` on obtiendra `a b`. Un argument est toujours encadré par des accolades et suit immédiatement le nom de la macro.

L'argument d'une macro peut aussi être du texte. Ainsi la macro `\emph` sert à mettre une partie du texte en italiques. Elle ne porte donc que sur le texte qui lui est passé en argument. Par exemple :

<p>Pour déguiser un peu les choses, il changea les premiers noms de <code>\emph{Roulette et Trochoïde}</code> en celui de <code>\emph{Cycloïde}</code>.</p>	<p>Pour déguiser un peu les choses, il changea les premiers noms de <i>Roulette et Trochoïde</i> en celui de <i>Cycloïde</i>.</p>
---	---

Une macro peut fort bien avoir plusieurs arguments : chaque argument sera placé entre accolades et ils devront se suivre dans l'ordre prévu par la macro en question. Connaître la syntaxe d'une macro, c'est connaître l'ordre et la signification de ses arguments. Par exemple, la macro `\raisebox` permet de déplacer verticalement quelques mots : elle attend un premier argument qui indique la valeur du déplacement vertical et un second qui contient les mots à déplacer. D'où l'exemple suivant qui montre par ailleurs que les macros peuvent en contenir d'autres. Ici la commande `\footnotesize` qui sert à obtenir des caractères encore plus petits qu'avec `\small`, est placée à l'intérieur du deuxième argument de la macro `\raisebox` :

B. Pascal`\raisebox{5pt}{\footnotesize (1623-1662)}` B. Pascal⁽¹⁶²³⁻¹⁶⁶²⁾

Il existe d'autre part des arguments optionnels : ce sont des arguments qu'il n'est pas obligatoire de préciser mais qui peuvent apporter une information supplémentaire modifiant le comportement d'une macro. Nous en avons déjà vu un exemple avec la commande initiale de notre fichier `roulette.tex`. Nous avons écrit au début de ce fichier :

```
\documentclass[a4paper]{article}
```

Ici le format `a4paper` est une valeur optionnelle. La macro aurait fonctionné aussi bien si nous avions simplement écrit `\documentclass{article}` mais elle aurait adopté par défaut un autre format (`letterpaper`). Un argument optionnel est toujours encadré par des crochets tandis qu'un argument obligatoire est toujours encadré par des accolades.

LEÇON 2

Où l'on parle de composition ...

Maintenant que nous avons vu les principes fondamentaux du fonctionnement de \LaTeX , nous allons donner un aperçu de ses capacités à prendre en charge réellement la structure d'un document. C'est à proprement parler un travail de composition puisque l'on transmet à \LaTeX un matériau souvent brute afin qu'il le mette en forme. Il s'agit ici non pas seulement de créer les paragraphes et les pages mais bien de structurer entièrement le document, d'en gérer les numérotations, de construire des tableaux ou des formules complexes, etc.

Structuration d'un document

Notre fichier source sur le *Traité de la Roulette* de Pascal a été déclaré comme un document en classe *article* au moyen de la commande initiale `\documentclass{article}`. Il existe également deux autres classes appelées *book* et *report*. En classe *article*, on peut subdiviser le document en sections, sous-sections, sous-sous-sections, paragraphes et sous-paragraphes. \LaTeX dispose de macros telles que `\section`, `\subsection`, etc. qui assurent à la fois la mise en pages et la numérotation de ces subdivisions. Il suffirait ainsi d'écrire simplement dans le fichier source (faites-en l'expérience)

```
\section{Traité de la roulette}
```

pour que \LaTeX crée, dans le document produit, une nouvelle section comme ceci :

1 Traité de la roulette

Cette section a automatiquement reçu un numéro (1 puisque c'est la première), son titre a été écrit en caractères gras et se trouve séparé du texte qui précède ou qui suit par un certain espacement vertical. Cette section pourrait contenir ensuite des sous-sections comme par exemple

```
\subsection{Présentation du texte}
```

qui donnera :

. Présentation du texte

La numérotation est maintenant différente, le titre est en gras mais dans un corps plus petit et l'espacement vertical est légèrement moindre. En classes *book* et *report*, on dispose en plus d'une commande `\chapter` qui permet de créer des chapitres. \LaTeX se charge alors de commencer une nouvelle page, d'y inscrire le mot chapitre suivi de son numéro, puis le titre du nouveau chapitre (qui est l'argument de la commande `\chapter`) en grands caractères et composer le texte plus bas sur la page.

On peut ainsi s'appuyer entièrement sur les réglages internes de \LaTeX pour assurer une mise en pages optimale. Nous verrons d'ailleurs par la suite que \LaTeX fait beaucoup plus que ce qui vient d'être décrit et qu'il s'efforce en particulier de ne pas couper les pages n'importe où : il est capable, quand besoin est, d'augmenter ou de diminuer l'espacement vertical qu'il insère entre les sous-titres, etc.

Une autre expérience instructive consisterait à écrire maintenant à un endroit arbitraire dans le fichier source `roulette.tex` la commande suivante :

```
\tableofcontents
```

En exécutant deux fois de suite (nous verrons plus tard pourquoi) le fichier, on obtiendra alors une table des matières parfaitement formatée reprenant les titres des sections et sous-sections que nous avons créées, comme ceci :

1	Traité de la roulette	8
1.1	Présentation du texte	8

Notion de boîtes

La notion de boîte est, elle aussi, fondamentale. À tous les niveaux, \LaTeX place les éléments qu'il manipule dans des boîtes qu'il place ensuite côte à côte ou bien l'une au-dessus de l'autre. Ainsi les lettres elles-mêmes sont vues comme de petites boîtes élémentaires. Lorsqu'il compile une ligne, \LaTeX place les boîtes-lettres les unes à côté des autres pour créer des boîtes-mots qu'il juxtapose à leur tour en insérant entre elles l'espacement qu'il juge nécessaire ; il obtient alors des lignes qui sont elles-mêmes considérées comme des boîtes qu'il empile pour en faire des pages.

Il est très facile de créer soi-même des boîtes. Supposons que l'on veuille placer l'un à côté de l'autre le texte français et le texte latin du début du *Traité sur la Roulette*. Il suffirait de mettre chacun dans une boîte et de juxtaposer ces boîtes. Faisons l'expérience en nous servant d'une macro appelée `\parbox` dont la fonction est justement de mettre des paragraphes dans une boîte. Celle-ci attend deux arguments qui sont respectivement la largeur de la boîte et son contenu. Écrivons donc, dans le fichier `roulette.tex`, les instructions suivantes (nous abrégons le texte français qui a été donné plus haut) :

```
\parbox{5cm}{La Roulette est une ligne si commune, qu'après
la droite et la circulaire, il n'y en a point de si fréquente ;
et elle se décrit si souvent aux yeux de tout le monde etc.}
\hspace{1cm}
\parbox{5cm}{Inter infinitas linearum curvarum species, si
unam circularem excipias, nulla est quae nobis frequentius
occurrat quam Trochoides (gallice \emph{la Roulette}) : ut
mirum sit quod illa priscorum seculorum geometras latuerit,
apud quos de tali linea nihil prorsus reperiri certum est.}
```

Nous avons placé entre les deux macros `\parbox` une commande `\hspace{1cm}` afin de les espacer d'un centimètre.

Le résultat est alors :

La Roulette est une ligne si commune, qu'après la droite et la circulaire, il n'y en a point de si fréquente; et elle se décrit si souvent aux yeux de tout le monde qu'il y a lieu de s'étonner qu'elle n'ait point été considérée par les anciens, dans lesquels on n'en trouve rien.

Inter infinitas linearum curvarum species, si unam circularem excipias, nulla est quae nobis frequentius occurrat quam Trichides (gallice *la Roulette*): ut mirum sit quod illa priscorum seculorum geometras latuerit, apud quos de tali linea nihil prorsus reperiri certum est.

Notion de liste

Il est une autre façon courante de structurer un texte qui consiste à faire des listes. \LaTeX dispose de trois environnements qui se chargent de la mise en forme de telles listes. Nous allons utiliser ici l'environnement `\begin{enumerate} ... \end{enumerate}` qui crée des listes numérotées. Chaque article de la liste est introduit par une commande `\item` et \LaTeX se charge de tout le reste (numérotation, indentation, espacements). Il suffit de taper dans le fichier `roulette.tex` le texte de gauche pour produire ce qui se trouve dans la colonne de droite :

Les problèmes que je proposai au commencement d'octobre dans l'histoire de la roulette sont ceux-ci :

```
\begin{enumerate}
\item Trouver le centre de gravité de la
ligne courbe.
\item Trouver la dimension et le centre
de gravité de la surface de son demi-solide
autour de la base.
\item Trouver la dimension et le centre
de gravité de la surface de son demi-solide
autour de l'axe.
\end{enumerate}
```

Les problèmes que je proposai au commencement de cet ouvrage dans l'histoire de la roulette sont ceux-ci :

1. Trouver le centre de gravité de la ligne courbe.
2. Trouver la dimension et le centre de gravité de la surface de son demi-solide autour de la base.
3. Trouver la dimension et le centre de gravité de la surface de son demi-solide autour de l'axe.

Notion de tableaux

La constitution de tableaux est aussi une des grandes forces de \LaTeX , suivant le principe maintenant bien compris que l'auteur se contente de fournir le matériau brut et que \LaTeX assure lui-même une composition optimale.

C'est un environnement `\begin{tabular} ... \end{tabular}` qui réalise les tableaux : les données y sont inscrites ligne par ligne, séparées par le symbole `&` et chaque ligne se termine par une double barre `\\`. En voici un exemple tiré du *Traité du triangle arithmétique* de Pascal :


```

\begin{tabular}{|l|cccc|}
\hline
Unités & 1 & 1 & 1 & 1 & 1 \\
\hline
Naturels & 1 & 2 & 3 & 4 & 5 \\
\hline
Triangulaires & 1 & 3 & 6 & 10 & 15 \\
\hline
Pyramidaux & 1 & 4 & 10 & 20 & 35 \\
\hline
\end{tabular}

```

Unités	1	1	1	1	1
Naturels	1	2	3	4	5
Triangulaires	1	3	6	10	15
Pyramidaux	1	4	10	20	35

L'argument `|l|cccc|` de l'environnement signifie que la première colonne doit être alignée à gauche (pour *left*) et que les cinq suivantes seront centrées (pour *center*) tandis que les barres verticales indiquent l'endroit où l'on veut des lignes verticales dans le tableau. Les commandes `\hline` permettent de tracer les lignes horizontales. On peut les modifier à volonté. Le plus remarquable est que L^AT_EX a calculé lui-même la largeur des colonnes de telle sorte que le matériau s'y insère exactement.

Formules mathématiques

Notre spécialiste de Pascal aimerait maintenant citer une formule mathématique liée à la théorie du Triangle Arithmétique. Il est bien connu que les nombres figurant sur les lignes du triangle sont les coefficients des puissances du binôme, comme par exemple les chiffres 1 3 3 1 dans l'identité :

$$(a + b)^3 = 1a^3 + 3a^2b + 3ab^2 + 1b^3$$

Pour indiquer à L^AT_EX qu'il faut compiler une formule mathématique, on utilise celle-ci de signes \$. Tout ce qui se trouve encadré entre deux signes \$ est considéré par L^AT_EX comme des mathématiques. La formule ci-dessus est ainsi obtenue tout simplement en écrivant :

```
$(a+b)^3=1a^3+3a^2b+3ab^2+1b^3$
```

Il suffisait ici de savoir que les exposants sont indiqués au moyen d'un accent circonflexe. Pour le reste, L^AT_EX s'est chargé lui-même de mettre les lettres en italiques, d'introduire du blanc autour des signes + et = et de placer les exposants.

LEÇON 3

Où l'on parle d'édition ...

Cette troisième et dernière leçon est consacrée aux fonctions plus avancées de \LaTeX et achèvera sans aucun doute de convaincre de son efficacité : il s'agit principalement des fonctions de référencement et d'indexation, c'est-à-dire de toute ce qui s'apparente à la phase post-rédactionnelle et donc à l'édition. Là encore, l'utilisateur n'a à connaître qu'une poignée de macros et c'est \LaTeX qui accomplira tout le travail. Comme dans les leçons qui précèdent, nous nous contentons de décrire le code et de le commenter. Il est important d'essayer soi-même de reproduire ce code dans le fichier source que nous avons appelé `roulette.tex` et de le faire exécuter par \LaTeX : pour les bibliographies et les références croisées, il faut obligatoirement exécuter le fichier *deux fois* de suite pour obtenir le résultat définitif.

Notes

Une étude critique sur Pascal nécessitera certainement des notes de bas de page. Celles-ci sont obtenues avec la macro `\footnote` qui prend le texte de la note comme argument. Par exemple :

```
Expériences nouvelles touchant le
vide\footnote{Cet opuscul e a été
publié en octobre 1647 chez Pierre
Margat à Paris. On le trouve à la
bibliothèque de Clermont-Ferrand
avec quelques lettres autographes.}
```

Expériences nouvelles
touchant le vide¹

Traité
du vide

Les notes sont numérotées séquentiellement et sont complétées en bas de page dans un cadre plus petit. On place tout aussi facilement des notes dans la marge avec la macro `\marginpar`. La note ci-citée a été obtenue en écrivant l'instruction suivante dans le texte de ce paragraphe :

```
\marginpar{\tiny Traité du vide}
```

Références

Nous voudrions maintenant renvoyer le lecteur à la sous-section 1.1 qui figure sur la page 8. Or, tant que le document n'a pas été compilé, nous ne savons ni le numéro que portera cette sous-section ni la page où elle sera placée. Ce sont d'ailleurs des données qui peuvent être amenées à changer, si nous modifions le plan du document ou si nous ajoutons ou retirons du texte par la suite. Le moyen fourni par \LaTeX consiste à

1. Cet opuscul e a été publié en octobre 1647 chez Pierre Margat à Paris. On le trouve à la bibliothèque de Clermont-Ferrand avec quelques lettres autographes.

placer une étiquette à l'endroit en question avec la commande `\label`. Cette étiquette devra porter un nom et c'est grâce à ce nom qu'on pourra ensuite faire référence à cet endroit du document. Le nom de l'étiquette est arbitraire : nous avons ainsi placé une étiquette que nous avons appelée *josephine* au début de la sous-section 1.1 en écrivant la commande `\label{josephine}`. Avec les commandes `\ref` et `\pageref`, on pourra produire à tout moment respectivement le numéro de la sous-section ainsi que sa page. Par exemple :

La sous-section `\ref{josephine}` se trouve sur la page `\pageref{josephine}`. La sous-section 1.1 se trouve sur la page 8.

Bibliographies

La gestion des bibliographies est élémentaire avec L^AT_EX mais peut être poussée très loin en leur attribuant des styles très élaborés. Nous allons nous contenter d'un exemple simple. Grâce à l'environnement `\begin{thebibliography} ... \end{thebibliography}` nous allons constituer une liste de deux ouvrages. Cet environnement fonctionne comme les listes que nous avons vues dans la leçon précédente, si ce n'est que chaque article de la liste est introduit par une commande appelée ici `\bibitem` au lieu de `\item`. Ainsi, avec les instructions suivantes :

```
\begin{thebibliography}{9}
  \bibitem{mair} Albert Maire (1925-1927) \emph{Bibliographie
    générale des œuvres de Blaise Pascal}, 5 vol., Paris,
    Giraud-Badin.

  \bibitem{bibn} \emph{Catalogue des ouvrages de Pascal conservés
    au Département des Imprimés} (1935), Bibliothèque Nationale,
    Paris, Imprimerie Nationale.
\end{thebibliography}
```

nous obtenons le résultat que voici :

Bibliographie

- [1] Albert Maire (1925-1927) *Bibliographie générale des œuvres de Blaise Pascal*, 5 vol., Paris, Giraud-Badin.
- [2] *Catalogue des ouvrages de Pascal conservés au Département des Imprimés* (1935), Bibliothèque Nationale, Paris, Imprimerie Nationale.

La numérotation a été faite automatiquement par L^AT_EX. Le chiffre {9}, dans cet exemple, indique la largeur réservée à cette numérotation comme nous le réexpliquerons par la suite : les étiquettes auront la largeur du chiffre 9.

Chacune des macros `\bibitem` s'est vue attribuer un mot-clé (`mair` pour le premier ouvrage et `bibn` pour le second) qui permettra par la suite d'y faire référence. Ils fonctionnent comme des étiquettes. Pour citer un de ces ouvrages dans le texte du document, on utilisera ensuite la commande `\cite` qui prend en argument le mot-clé de l'ouvrage que l'on veut citer.

Par exemple :

```
Consulter le catalogue \cite{bibn}
```

pr duira :

```
Consulter le catalogue [2]
```

Index

\LaTeX est capable de constituer des index du type de ceux que l'on place en fin d'ouvrage pour renvoyer le lecteur aux pages traitant d'un sujet particulier. Supposons que notre professeur de littérature veuille indexer le nom de Galilée qui a été mentionné à la page 5. Il suffit de prendre soin, chaque fois que ce nom est mentionné, de le faire suivre de la commande `\index{Galilée}`. C'est ce que nous avons fait. Le lecteur pourra vérifier dans l'index à la fin de cet ouvrage que Galilée y figure bien.

La commande `\index` sert à relever tous les mots à indexer. La prudence de l'index lui-même est un peu plus longue à expliquer : on se reportera au paragraphe 6.2.4.

Macros personnelles

Pour terminer, voici certainement l'aspect le plus satisfaisant de l'utilisation de \LaTeX : il s'agit de la possibilité de créer ses propres macros. Il existe une commande de \LaTeX appelée `\newcommand` permettant de déclarer et définir de nouvelles macros. Voyons-en tout de suite un exemple : supposons que notre professeur soit lassé d'écrire perpétuellement le titre «*Traité du Triangle Arithmétique*». Il va simplifier son travail de saisie en définissant une macro qui exécutera cela à sa place. Nous baptiserons cette nouvelle macro `\ttra` et placerons sa définition au début du fichier source (après la commande `\documentclass` et avant la commande `\begin{document}`) :

```
\newcommand{\ttra}{Traité du Triangle Arithmétique}
```

Désormais il suffira d'écrire `\ttra` chaque fois que l'on voudra obtenir l'expression entière. Par exemple :

```
Le \ttra{} date de 1654.
```

```
Le Traité du Triangle Arithmétique
date de 1654.
```

L'intérêt de cette méthode, outre qu'elle facilite ici la saisie, est que l'on peut décider par la suite de modifier la définition de cette macro : la modification de répercutera alors immédiatement à travers tout le document. Si nous voulons que le titre *Traité du Triangle Arithmétique* soit mis en italiques il suffirait de changer la définition comme ceci :

```
\newcommand{\ttra}{\emph{Traité du Triangle Arithmétique}}
```

Nous pourrions demander par la même occasion que ce titre soit indexé systématiquement. On changerait la définition en :

```
\newcommand{\ttra}{\emph{Traité du Triangle Arithmétique}}
\index{Triangle Arithmétique}
```

La macro `\ttra` accomplirait ainsi une double tâche : écrire le titre dans le style demandé et l'indexer.

Résumé des principes de base

Ces tris leçons ont donné un aperçu des principales fonctionnalités de \LaTeX mais le panorama n'est pas exhaustif. Il y a encore bien d'autres aspects à explorer. Nous allons cependant résumer ici les traits caractéristiques qui distinguent radicalement \LaTeX des traitements de texte ordinaires. Il s'agit de points qui paraissent surprenants au départ mais dont on comprendra et appréciera très rapidement la pertinence :

- il y a une distinction radicale entre fichier source et fichier produit ;
- dans le fichier source, les sauts de lignes sont ignorés par \LaTeX . Des lignes successives font toute partie d'un même paragraphe. \LaTeX remplace un passage à la ligne dans le fichier source par une espace. Pour obtenir un saut de ligne dans le fichier produit, il faut placer une commande de fin de paragraphe ;
- des blancs successifs sont ignorés par \LaTeX : ils sont considérés comme un seul et unique blanc ;
- les blancs au début d'une ligne dans le fichier source sont ignorés. Cela permet parfois de décaler des éléments vers la droite dans le fichier source afin de le rendre plus lisible sans que cela ait la même incidence sur le document produit ;
- toutes les actions que l'utilisateur veut réaliser sont obtenues au moyen de commandes ou macros que l'on place à l'endroit requis dans le fichier source ;
- on peut délimiter la portée des macros en définissant des zones au moyen de paires d'accrochages `{ et }` ou bien en utilisant des environnements ;
- certains symboles ont une signification particulière dans les commandes du fichier source :
 - la commande `\` au début d'un mot caractérise les macros,
 - les accrochages `{ et }` délimitent des groupes : une commande placée dans un groupe n'agit que jusqu'à l'accrochage fermante de ce groupe,
 - le symbole `&` est utilisé dans les tableaux pour séparer les entrées devant figurer sur un même rang du tableau,
 - le symbole `%` sert à placer des commentaires dans le fichier source : tout ce qui se trouve à droite d'un signe `%` jusqu'au bout de la ligne est ignoré par \LaTeX . On peut ainsi placer des annotations ou des explications qui figureront dans le fichier source mais n'apparaîtront pas dans le document final. Chaque ligne de commentaire est introduite par un `%`,
 - nous verrons aussi que le symbole `@` ne peut pas être utilisé tel quel dans le fichier source : il caractérise les commandes internes de \LaTeX et les protège en les rendant inaccessibles,
- le fichier source est un fichier de données brutes qui doit être traité par \LaTeX : à la sortie de ce traitement, on obtient le document final sous la forme d'un fichier dit DVI. Pour juger le résultat, il faut visualiser le fichier DVI à l'écran ou bien l'imprimer. On n'a donc pas une visualisation directe du fichier source au fur et à mesure de sa rédaction. Ce point paraît décourageant au début mais s'explique par le mode de fonctionnement de \LaTeX et est vite contrebalancé à la fois par la puissance du système dans son ensemble et par la qualité typographique du résultat obtenu.