コンピュータリテラシ＃10 – マークアップによるテキスト整形

久野 靖 (電気通信大学)

2017.5.19

1　今回の目標

今回の目標は次の通りです。

- マークアップ方式によるテキスト整形の概念を理解する — ワープロソフトよりも効率よく作業できることを理解しておくと、効率を高める工夫がしやすくなります。
- LaTeX による文書の作成方法について理解する — 理系では LaTeX によってレポート・論文を書くことが普通です。
- 数式や表なども含む、LaTeX のさまざまなマークアップについて学ぶ — 数式や表はレポート・論文で非常によく使います。

2　文書作成

2.1　テキストと文書の違いと位置付け

コンピュータは情報を処理する装置ですが、人間の知的活動 (情報) のアウトプットは大半が**文書** (書籍、レポート等) です。このため、文書を作成する**ワープロソフト**や**文書整形系** (formatter、後述) は、コンピュータで最も活躍するアプリケーションの 1 つです。ここで、**テキスト**と文書の違いについて触れておきます。テキスト情報は「文字で表された情報」で、テキストを扱うソフトの代表テキストエディタです。エディタでは文字の並びは自由に修正できますが、作成したファイルの文字はどれも同じ大きさで、美しく読みやすいとは言えませんね。我々は大量の情報を取り入れるために文書を読むので、それが美しく読みやすいことは大切です。その具体例に入る前に、次の問いを考えてみましょう。

- 美しい文書とはどういうものを言うのか?
- なぜ美しい文書が望ましいのか?

1 番目の問いに対して「文字がきれいな形である」「多様な字形 (フォント) が使われてる」などが浮かんだ人もいそうですが、文字がきれいでもでたらめな規則で並んでいたら嬉しくないですね。

そこで 2 番目の問いですが、なぜ美しい文書が望ましいのでしょう? それは、文書が美しいと、内容を読み取る効率がよいからです。具体的には、文字の並び方が読みやすく、「ここは見出し」「これは図」などの構造が把握しやすく、必要な箇所が探しやすいこと、つまり次のことです。

- 文書の構造が読み手に的確に伝わること。

逆にいえば、「美しい文書」では、内容であるテキスト (文字) に加え、文書のどこが何であるの**付加情報**も含まれています。これによって「見出しだから大きく」などの処理が可能になるわけです。

2.2 文書整形のアルゴリズム

ワープロソフト (や文書整形系) が「どのように」各文字を紙面に配置しているかを言葉で説明できますか? たとえば、初期のワープロソフトなどでは「原稿用紙」モデル、つまり画面や紙の上に縦横のサイズが決まったます目があり、そこに 1 文字ずつ文字を詰めて行く、というアルゴリズムが使われていました。それだとどんな美しくないことが起きるか分かりますか?

- 禁則処理 (行頭に「。」などが来ないための処理) の結果、右端が「でこぼこ」になる (図 1 上)。
- 英字を日本語と同じ文字間隔で詰めるとえらく間延びになる。かといって日本語 1 文字ぶんに英字 2 文字を詰めると窮屈 (しかも英字が奇数だと半分のアキが…)。
- 箇条書きなど字下げの段落で文字詰め幅を変更すると空白がずれて悲惨になる (図 1 下)。

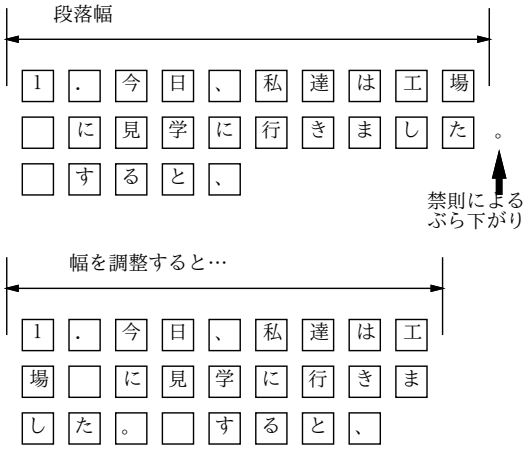

図 1: 「原稿用紙方式」の困った点

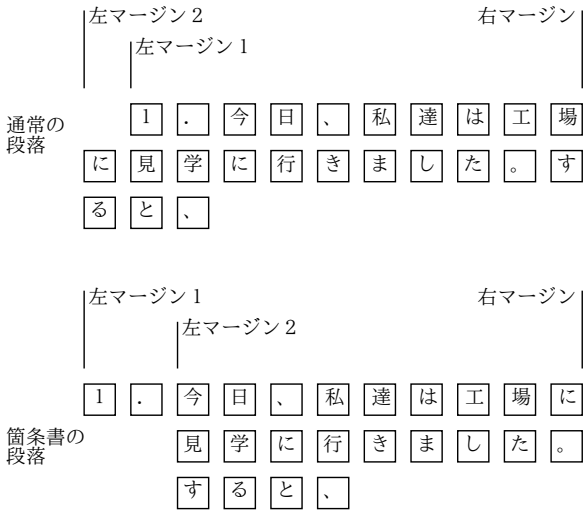

図 2: マージン指定による段落の詰め合わせ

今日のソフトではこのようなことはなく、段落は「1 行目の左マージン」「2 行目以降の左マージン」「右マージン」の位置で形が指定されており、文字を普通に打って行くことでこれらのマージン内で自動的に文字が詰め合わされて行くようになっています (図 2)。ただし、これをちゃんと使うには、段落の形を変える時に「ルーラ」を出してマージンを設定する必要があります (これを知らずに字下げを空白で調整している人もまだ多いようです)。

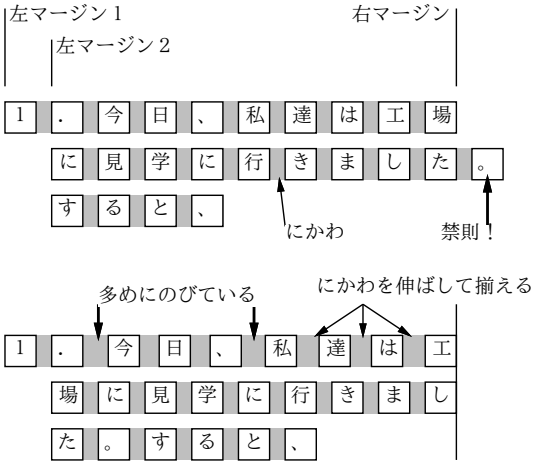

図 3: 整形のアルゴリズム

さらに禁則なども考慮して整形するため、現在は次のようなアルゴリズムが使われます (図 3)。

- 段落ごとに「詰め合わせる幅」があり、それぞれの文字は「四角い箱」と考えてその箱を幅一杯まで詰めていく。
- 文字と文字の組み合わせごとに適切な「あき」が変わってくるので、組み合わせごとに適切な「にかわ (のり)」を詰めていく。
- 禁則処理などでその位置で行かえできないなら、少し余分に詰めるか詰めたものを取り除く。
- 右端が揃うように全体を詰める/伸ばする (この時にかわ部分が伸び縮みする)。
- すべてのにかわが均等に伸び縮みするわけではなく、「(」の左、「)」の右、「。」の右など、伸びてもおかしくないところが余分に伸びるようにする。
- 各行が出来てくると、今度はページに行を上から詰めていく。ここでも「にかわ」がはさまれ、ページの切れ目の禁則 (節タイトルがページ下端に来ない等) に応じて同様の処理が行われる。

ここで重要なのは、「文字どうしのアキ (にかわの幅や伸び)」「段落間のアキ」「マージンの値」などをどう調整すれば見やすい文書になるか、ということです。これについては、長い歴史のある印刷・出版界が多くのノウハウを持っており、それを駆使して読みやすい書籍を作っています。そこで、LaTeX などの整形システムでも、これらのノウハウを借りて来てパラメタを調整することで、できるだけ見やすい整形となるように努めているわけです。

2.3 見たまま方式とマークアップ方式 exam

美しい文書を作成するためには、テキストと付加情報をともに入力したり修正するなどの必要があることまでは分かりました。ではそれを具体的にどのようにして行なったらよいでしょうか? コンピュータの世界では、その方法として次の 2 種類が考えられ、現に使用されています (図 4)。

- 見たまま方式、**WYSIWYG**(What You See Is What You Get — 「あなたが見るものがあなたの得るものである」) 方式。この方式では、実際にプリンタで印刷したイメージにできるだけ近いものを生成し、画面表示します。そしてマウスやメニューなどを用い、画面上に見える文書を対象に修正を施します。配置やフォントを変更したり、文章を直したりすると、その結果は直ちに画面に反映されます。Word などもこの仲間です。
- **マークアップ** (markup) 方式 — テキストエディタを用い、ファイル中にテキストに混ぜて特別な「印」(マークアップ) を入れ、整形系と呼ばれるソフトウェアがこれに従ってレイアウトを

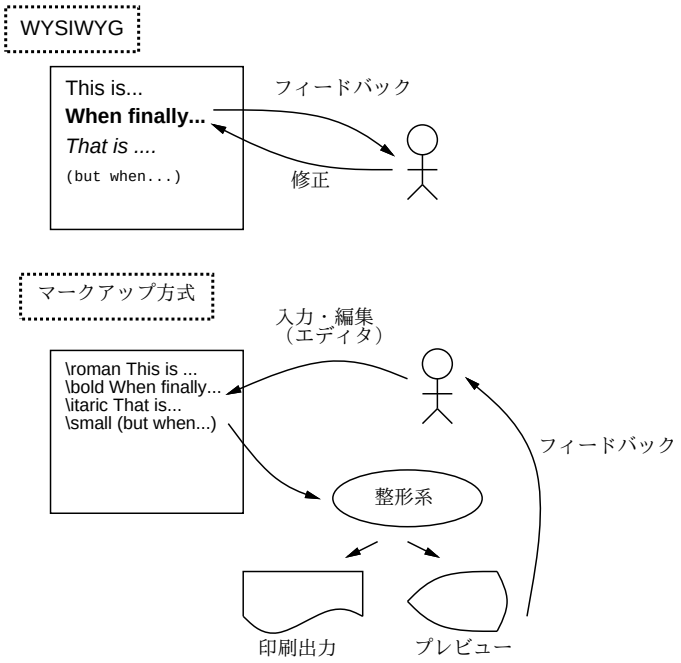

図 4: WYSIWYG 方式とマークアップ方式

行います。今日のマークアップは「ここは見出し」「ここは引用」など、個々の部分の意味を指定する**意味づけ方式** (semantic encoding) を採用しています。マークアップではコマンドによって細かい制御が行えますが、レイアウトをチェックする際に「整形+プレビュー」操作をおこなう必要があります。

世の中のワープロソフトでは見たまま方式が主流なのに、ここではマークアップ方式を取り上げるわけですが、その理由について説明しておきましょう。

- 見たまま方式は、個所ごとに「このスタイル」という指定を繰り返し行い、そのマウスやメニュー操作は時間が掛かります。マークアップなら、どこはどういうマークアップをするか覚えればあとは「打ち込む」作業の一部であり、時間が掛かりません。
- 見たまま方式は (多くの場合)「こういう体裁に」ということを人間がデザインし制御するので、センスがないと美しくできません。意味づけ方式では、スタイルは標準化されているので、そのようなことに頭を悩ます必要はありません。[1]
- 見たまま方式は、1 つの体裁に合わせて文書を作るため、後で体裁を変更するのは大変です。意味づけ方式なら、整形用スタイルを複数用意することで、多様な体裁に対応できます。
- 見たまま方式は、ファイル形式が特定ソフトに依存しがちで、コンテンツの流用が難しくなります。[2] マークアップはもともと全部テキストファイルなので、(マークアップの統一的な書き換えなどの処理を行うなどでして) 流用が容易です。

「文書を作るのにコマンドを覚えるなど耐えられない」「そんな面倒な方式は旧態依然だ」と思ったでしょうか? しかし実は、大量に文書を作成するプロほど、LaTeX のような整形系をメインに使っています。それは大量に文書を作る人にとってはコマンドの方が楽で効率的だからです。また、LaTeX はスタイルのチューニングが徹底しているので、「おまかせ」で比較的美しい文書ができます。

[1] WYSIWYG でもスタイルシート機能を活用すれば意味づけ方式を GUI でやる形になりますからスタイル設計者次第にできますが、その分マークアップに近くなり難しくなるので使わない人も多いです。

[2] テキストファイルに変換すれば他のソフトに持っていけますが、付加情報は失われてます。

2.4 LaTeX を動かしてみる

ではまず LaTeX を「とにかく」体験してみて頂きます。コマンドの書き方は後回しにして、先にサンプルファイルを「そのまま」整形してみましょう。LaTeX の入力ファイルは「.tex」で終わる名前をつけます。以下ではそれが「sample.tex」であるものとして説明します。整形のためのコマンドは platex ですので、入力ファイルを指定してこれを起動します。

```
...$ platex sample.tex
This is e-pTeX, Version 3.1415926-p3.4-110825-2.6 (utf8.euc) (TeX Live 2013)
 restricted \write18 enabled.
entering extended mode
(./sample.tex
pLaTeX2e <2006/11/10> (based on LaTeX2e <2011/06/27> patch level 0)
Babel <3.9f> and hyphenation patterns for 78 languages loaded.
(/export/opt2/texlive/2013/texmf-dist/tex/platex/base/jarticle.cls
Document Class: jarticle 2006/06/27 v1.6 Standard pLaTeX class
(/export/opt2/texlive/2013/texmf-dist/tex/platex/base/jsize12.clo))
(./sample.aux) (/export/opt2/texlive/2013/texmf-dist/tex/latex/base/omscmr.fd)
[1] [2] [3] (./sample.aux) )
Output written on sample.dvi (3 pages, 7524 bytes).
Transcript written on sample.log.
```

自分でファイルを作った際は、エラーがあると途中で「?」というプロンプトが出て止まります。その時は「x[RET]」と打って実行を終わらせ、エラーメッセージを見て間違いを直し、再度実行してください。整形が成功すると.dvi で終わる名前のファイル (この場合だと sample.dvi) ができます。

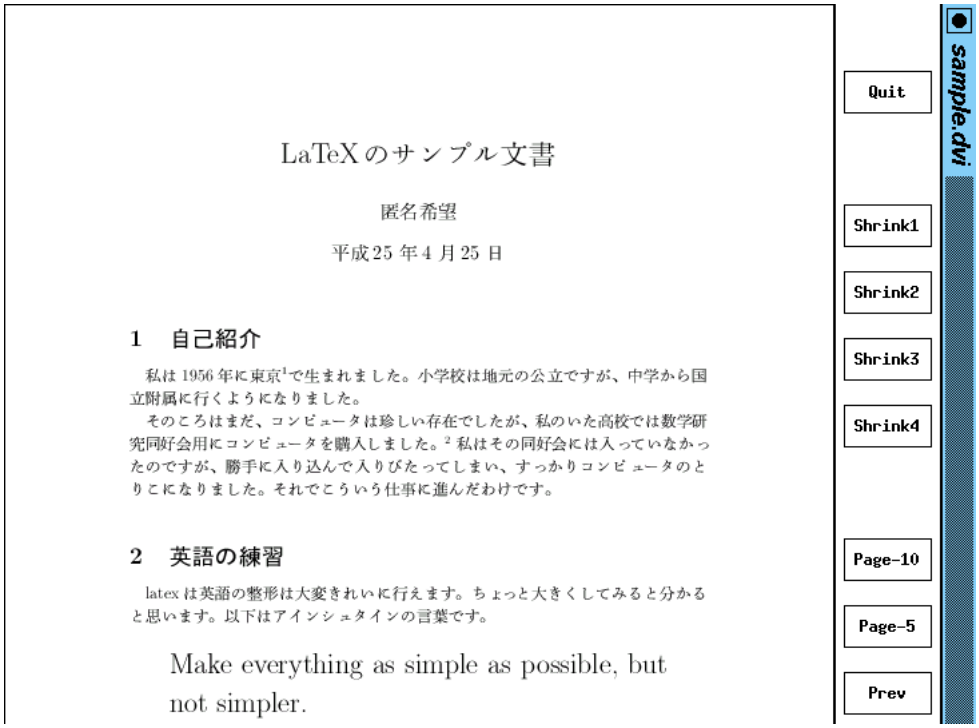

図 5: pxdvi によるプレビュー

整形結果はまず画面で確認します。それにはプレビューア pxdvi を「pxdvi sample.dvi &」で起動します。すると、図 5 のような窓が現れます。右側に縮尺やページめくりを指定するボタンがある

ので、これらをマウスで操作し、各ページが意図どおりか確認します。なお、他の人に渡したりプリンタに出すには「dvipdfmx sample.dvi」により PDF 形式を生成してください。

大変だったでしょうか? 確かに最初はとっつきにくいですが…では、先頭の行を次のように直すとどうなるか、試してみてください。

```
\documentclass[12pt,a4j]{jarticle} ←文字サイズ変更
\documentclass[twocolumn,a4j]{jarticle} ← 2 段組み
```

このような取り換えが簡単なのが意味づけ方式の利点だということが理解して頂けるかと思います。

演習 1 sample.tex を入手して整形し、結果を画面で確認してみなさい。うまく行ったら、冒頭の documentclass 指定について次の変更を 1 つ以上 (できれば全部) 行ってみなさい。

a. twocolumn 指定の有無による違いを比べてみる。
b. フォントサイズ指定 (9pt、10pt、11pt、12pt のどれか) を変えた時の違いを調べる。
c. 紙サイズ指定 a5j、b5j、a4j、b4j のどれか) を変えた時の違いを調べる。

いずれも、指定を変更した時にきれいに見えるようにどのような工夫が見られるか、文字サイズに比べて整形幅が狭くなった場合に整形時にどのようなメッセージが出てどのような整形結果になるかを見ること。

3 文書整形系 LaTeX

3.1 文書の基本構造 exam

ここでは、意味付け方式の文書整形システム **LaTeX** を実例中心で一通り解説します。まず、LaTeX の文書に最低限必要な基本構造の例を見てみます。次のものは、一つの完結した LaTeX 文書です (内容も読んでください)。

```
\documentclass{jarticle}
\begin{document}
ここに示すように、LaTeX の文書はまずこの文書がどんなスタ
イルの文書であるかを宣言する部分から始まる。スタイルの例
としては「記事 (jarticle)」、「本 (jbook)」、「報告
(jreport)」などがある。ここでは一番簡便な jarticle を例に
使用している。これに加えて字の大きさ、図形の取り込み機能
などをオプションとして指定できる。

続いて「文書開始」の宣言があり、この中に文書の本体が入る。
この部分には様々な記述が可能だが、一番簡単にはここに示す
ように段落ごとに 1 行あけて次々に文章を書いて行くだけでも
地の文が普通にできる。つまり、「特に指定がない」ならば
「地の文」である。

あとは文書の本体が終わったら最後に必ず「文書終了」の宣言
がある。最低限必要なのはこれだけである。
\end{document}
```

これを LaTeX に掛けて整形すると図 6 になります。ここで少し補足しすると、まず宣言 (指令) は

```
\指令名 [オプション指定]{パラメタ}
```

ここに示すように、LaTeX の文書はまずこの文書がどんなスタイルの文書であるかを宣言する部分から始まる。スタイルの例としては「記事 (jarticle)」、「本 (jbook)」、「報告 (jreport)」などがある。ここでは一番簡便な jarticle を例に使用している。これに加えて字の大きさ、図形の取り込み機能などをオプションとして指定できる。

続いて「文書開始」の宣言があり、この中に文書の本体が入る。この部分には様々な記述が可能だが、一番簡単にはここに示すように段落ごとに 1 行あけて次々に文章を書いて行くだけでも地の文が普通にできる。つまり、「特に指定がない」ならば「地の文」である。

あとは文書の本体が終わったら最後に必ず「文書終了」の宣言がある。最低限必要なのはたったこれだけである。

図 6: pLaTeX の出力

の形をしています。オプション指定がないときは `[...]` はなく、またパラメタがないときは`{...}`も不要です。ということは「\」はそのままでは文書に含められないわけです。LaTeX では、このような特別な (そのままでは使えない) 記号として次のものがあります。[3]

```
# $ % & ~ _ ^ \ { }
```

3.2 表題、章、節 exam

本文だけの文書では見づらいので、表題と章立てをつけましょう。その場合の例を示します。

```
\documentclass{jarticle}
\begin{document}
\title{あなたがつけた表題}
\author{書いた人の名前}
\maketitle

\section{表題について}
表題をつけるには title、author など必要な事項を複数指定した後、最後に maketitle というとそれらの情報をもとに表題が生成される。その際指定されなかった事項は出力されないかまたは適当なものが自動生成される (たとえば日付を指定しないと整形した日付が入る。)

\section{章立てについて}
ここにあるように section 指令を使って各節の始まり、およびその表題を指定する。節の中でさらに分ける場合には section というのも使え、さらに subsection というのまで可能である。ちなみに jbook/jreport スタイルでは section の上位に chapter があるが jarticle では section からである。ところで、節番号 etc. は自動的に番号付けされるのに注意。

\section{より細かいことは}
より細かいことは、参考書を参照してください。
\end{document}
```

[3] この他に<、>など記号自体に特別な意味はないのですが、TeX の標準フォントの関係で出力すると別の字になってしまうものがいくつかあります。

これを整形すると、文書の冒頭にタイトル、著者、日付 (整形した日付) [4] がそれなりの形式で用意され、見出しも前後にアキを取ってそれなりのフォントで出力されます。なお、section より小さいレベルの見出しとして subsection、subsubsection も使えます。

3.3 いくつかの便利な環境 exam

表題と章建てがあれば普通の文書がかなり書けますが、全部地の文ではめりはりがないですね。それに、プログラム例など行単位でできているものまで詰め合わされては困ります。このように、部分的にスタイルが違う範囲は次のような形 (**環境**と呼びます) を使います。

```
\begin{環境名}
....
....
\end{環境名}
```

では、よく使う環境について、説明していきましょう。まず、**verbatim 環境** (そのまま) というのは文字通り入力をそのまま整形せずに埋め込みます。たとえば

```
\begin{verbatim}
This is a pen.
 That is a dog.
\end{verbatim}
```

は、つぎのような出力になります。プログラム例などを示すのに最適です。

```
This is a pen.
 That is a dog.
```

ちなみに、独立した行にする代わりに、文章のなかに一部「そのまま」を埋め込みたい場合もあります。そのような時には verbatim の類似品で\verb|....|という書き方を使うことができます。これで縦棒にはさまれた部分がそのまま出力できます。中に縦棒を含めたい時は、両端に縦棒以外の適当な記号を使います。なお、verbatim も verb も脚注 (後述) の中には入れられません。

つぎに、**itemize 環境** (箇条書き) について説明しましょう。この場合は、環境のなかに複数「\item ....」というものが並んだ格好になっていて、その一つずつが箇条書きの 1 項目になります。たとえば

```
\begin{itemize}
\item あるふぁはギリシャ文字の一番目です。
\item ベータはギリシャ文字の二番目です。
\item ガンマはギリシャ文字の三番目です。
\end{itemize}
```

は、つぎのような出力になります。

- あるふぁはギリシャ文字の一番目です。
- ベータはギリシャ文字の二番目です。
- ガンマはギリシャ文字の三番目です。

次に、この itemize を **enumerate 環境** (数え上げ) に変更すると、出力の際に項目ごとに 1, 2, 3... と自動的に番号付けされるようになります。入力はほとんどまったく同じだから出力のみ示します。

[4] 日付を自分で記入したければ「\date{ 日付表記 }」を author の後あたりに入れます。

1. あるふぁはギリシャ文字の一番目です。
2. ベータはギリシャ文字の二番目です。
3. ガンマはギリシャ文字の三番目です。

点や番号でなくタイトルをつけたい場合には **description 環境** (記述) を使います。これは

```
\begin{description}
\item[あるふぁ] これはギリシャ文字の一番目です。
\item[ベータ] これはギリシャ文字の二番目です。
\item[ガンマ] これはギリシャ文字の三番目です。
\end{description}
```

のように、各タイトルを [] の中に指定するもので、上の整形結果は次のようになります。

あるふぁ これはギリシャ文字の一番目です。

ベータ これはギリシャ文字の二番目です。

ガンマ これはギリシャ文字の三番目です。

この他にもいくつか環境がありますが、とりあえずこれくらいで十分使えると思います。

3.4 脚注

脚注の作り方は、単に好きなところに`\footnote{.....}`という形で注記をはさんでおけばそれがページの下に集められて脚注になり、そこへの参照番号は自動的につけられます。[5]

3.5 文字サイズ

文字サイズ指定は、`{\LARGE 大きく}`のように書くと{\LARGE 大きく}なります。このように LaTeX では`{...}`が範囲を区切り、字体や文字サイズを変更した効果はその範囲を出ると終わります。文字サイズとしては{\huge huge}、{\LARGE LARGE}、{\Large Large}、{\large large}、normalsize、{\small small}、{\footnotesize footnotesize}、{\scriptsize scriptsize}、{\tiny tiny} が指定できます。字体は`\textbf{Bold}`で**Bold**(ボールド体)、`\texttt{Typewriter}`で`Typewriter`(タイプライタ書体)、`\textit{Italic}`で*Italic*(イタリック)、そして`\textrm{Roman}`でRoman(立体 — 普通の書体) です。

3.6 表 exam

表は情報を整理して伝えられまです。LaTeX では表は**tabular 環境**で作り出します。その先頭では、

- 表のカラム数
- それぞれのカラムを左/中央/右揃えのどれにするか
- 各カラムの境界および左右に罫線を引くかどうか

を文字列で指定します。1 つのカラムごとに揃え方を `l`/`c`/`r` のうち 1 文字で指定し `l`/`c`/`r` の文字数がカラム数になります)、これらの文字の間や左右端に「`|`」を挿入するとそこに縦罫線が入ります。具体例で見てみましょう (**center 環境**は表全体を中央揃えするものです)。

[5] たとえば、こんな具合になりますね。

```
\begin{center}          ←見ばえのため
\begin{tabular}{c|ll}
AND 演算 & 0 & 1 \\
\hline                  ←横罫線
0       & 0 & 0 \\
1       & 0 & 1 \\
\end{tabular}
\end{center}
```

上のようにすると、次のように表示されます。見れば分かる通り、表の内部では「&」がカラムの区切り、「\\」が 1 行終わり、「\hline」が横罫線を表します。

AND 演算	0	1
0	0	0
1	0	1

ところで、表の一部について「セルを結合」したいことがよくあります。横方向の結合は multicolumn コマンドを使ってできます (縦結合もできますがここでは省略)。たとえば、先の例の 2 つ並んだ「0」を 1 個にまとめたい場合は、0 の行を次のようにします。

```
0 & \multicolumn{2}{|c}{0} \\ ← 2 セル結合、内容は「0」
```

結果は次のようになります。

AND 演算	0	1
0	0	
1	0	1

3.7 数式 exam

TeX 属の整形系はもともとは**数式**を美しく打ち出したい、という目的のもとに開発されたものなので、数式機能がとても充実しています (そのため凝り出すと大変ですから、ここでは簡単に説明します)。まず、数式を文中にはさむときは$で囲みます。たとえば、

…と書くと$x^2 - a_0$のような具合になります。

と書くと $x^2 - a_0$ のような具合になります。ここで出てきたように、肩字は^、添字は_で表せます。その他、数学に出てくる記号はたとえば

```
\equiv \partial \subset \bigcap \cdots \sum \int
```

のように書くと $\equiv \partial \subset \bigcap \cdots \sum \int$ のように出てきます。また、文中ではなく独立した行にしたければ、$...$の代わりに\[...\] ではさむか、\begin{displaymath} … \end{displaymath}で囲みます (どちらでも同じ意味)。たとえば

…と書くと\[f(t) = \sum_{j=1}^m a_j e^{i\lambda_j} t\] になります。

と書くと

$$f(t) = \sum_{j=1}^{m} a_j e^{i\lambda_j} t$$

になります。このように、肩字や添字のグループ化には{}を使います (丸かっこは数式本体のために使います)。もう 1 つよく使うのは分数で、$\frac{a}{b}$は $\frac{a}{b}$ となります。もっと網羅的な記号一覧などは TeX の入門書などを参照してください。

演習 2 LaTeX の数式機能を使って、次のようなものを 1 つ以上 (できれば全部) 指定してみなさい。

a. $a_0 + a_1 + \cdots + a_n$

b. $\sum_{i=1}^{N} \int_0^i f(x)dx$

c. $x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3}}}$

できれば、それぞれについてもう少し複雑な「技」を追加してみるとなおよいです。

演習 3 LaTeX を使って自分の自己紹介の文書を作成してみなさい。タイトルと章立ては必ずおこなうこと。加えて次の機能から 1 つ以上 (できれば全部) を活用すること。文書内容は「あああ」みたいな意味のないものではなく、それらの機能を使う必然性のある形で使ってください。

a. 箇条書きや引用。できれば、箇条書の中に箇条書を入れて何が起きるか観察できるとなおよいでしょう。
b. 脚注。できれば、脚注の中で使えない機能として何があるか試してみるとなおよいでしょう。
c. 表。できれば、複数カラムの結合も使ってみられるとなおよいでしょう。

本日の課題 **10A**

本日の課題は「演習 1」「演習 2」「演習 3」に含まれる小問 (合計で 9 個) の中から 1 つ以上を選択し、結果をレポートとして報告して頂くことです。ただし、「演習 3」から必ず 1 つ以上選ぶこと (演習 1、演習 2 の内容を含める場合はこの中に一緒に書き込む)。そして、LaTeX による整形結果を PDF ファイルにして、LMS の「レポート＃ 10」の箇所からアップロードしてください。以下の内容がこの順に含まれるようにしてください。

- 題名「コンピュータリテラシレポート＃ 10」、学籍番号と氏名、提出日付を書く。(グループでやったものはグループのメンバー全員の氏名も脚注などで別途書く。)
- 課題の再掲を書く (どんな課題であるかをレポートを読む人が分かる程度に要約する)。
- レポート本体の内容 (やったこととその結果) を書く。
- 考察 (課題をやった結果自分が新たに分かったことや考えたこと) を書く。
- 以下のアンケートに対する回答。

Q1. 普段どのくらい文書を作成していますか。またそのときに使うソフトや心がけていることなどを教えてください。
Q2. LaTeX のようなマークアップ型の文書作成系についてどのように感じましたか。
Q3. リフレクション (今回の課題で分かったこと)・感想・要望をどうぞ。

なお、課題はグループでやって構いません。その場合も、(メンバー氏名を明記した上で) レポートは必ず各自で執筆してください。レポート文面が同一 (コピー) と認められた場合は同一であると認めた全員について点数にペナルティを科すことがあります。