

pTeX 系列以外による日本語文書作成

2018 年 4 月 15 日

Knuth 氏は、1990 年にこれ以上 T_EX^{*1} の機能拡張を行わないことを宣言した：

My work on developing T_EX, METAFONT, and Computer Modern has come to an end. I will make no further changes except to correct extremely serious bugs.

I have put these systems into the public domain so that people everywhere can use the ideas freely if they wish. ... Let us regard these systems as fixed points, which should give the same results 100 years from now that they produce today.

— Donald E. Knuth, *The future of T_EX and METAFONT*, TUGboat **11**, No. 4, 1990.

<http://tug.org/TUGboat/Articles/tb11-4/tb30knut.pdf>

しかし、これで「T_EX は終わった」わけではない。上の引用の第 2 段落にあるとおり、T_EX82 を改良して（別の名前で^{*2}）公開することは自由であり、今まで世界中で数々の拡張が試みられ、現在でも拡張の開発が進められている^{*3}。詳しくは

- ・ 八登崇之, 「日本人の知らない T_EX—T_EX の過去・現在・未来」, T_EX ユーザの集い 2010.
<http://zrbabbler.sp.land.to/texconf10.html>
- ・ 寺田侑祐, 「近年の T_EX の動向」, 数学ソフトウェアとフリードキュメント XVIII.
- ・ Arno Trautmann, *An overview of T_EX, its children and their friends* ...
<http://mirrors.ctan.org/info/tex-overview/tex-overview.pdf>
- ・ T_EX Wiki 中の「LuaT_EX」「X_YT_EX」「upT_EX」「ε-T_EX」等の各記事

などを参照。

pT_EX ももちろん T_EX82 の拡張であるのだが、本文章では、**pT_EX** 系列以外の代表的な **T_EX82** の拡張と、それによって日本語文書を作成する方法について述べる。取り扱うのは表 1 に載せた 3 つの方法である^{*4}。

Last update: 2018/4/15 17:41.

本文書は X_YT_EX と ZXjatype パッケージを用いて組版した。それ以外の「T_EX Live のインストール」「T_EX 実習」は、LuaT_EX-ja を用いて組版している。

^{*1} 本文書では、Knuth 氏の作成した（プログラムとしての）T_EX を **T_EX82** と呼ぶことにする。

^{*2} この「内容を改変した場合は別の名称にしなければならない」というのは T_EX 業界での特徴的な約束事である。

^{*3} 日本で主流の pT_EX, また upT_EX に関しても、バグ修正や不自然な挙動の改善が現在でも時々なされている。

^{*4} 表 1 には比較のため、「pT_EX + DVIPDFMx」も載せている。

表1 本文書で扱う p_TE_X 系列以外の T_EX

節	プログラム	パッケージ	ドライバ指定	
			graphicx	hyperref
§3	pdf _T E _X	CJK, bxcjkatype	pdftex (自動)	pdftex (自動)
§4	X _Y _T E _X	xeCJK, zxjatype	xetex (自動)	xetex (自動)
§5	Lua _T E _X	Lua _T E _X -ja	pdftex (自動)	pdftex (自動)
(p _T E _X + DVIPDFM _x	—	dvipdfmx	dvipdfmx)

目次

1	導入	3
2	予備知識：ε-T _E X 拡張	4
3	pdf _T E _X	6
3.1	pdf _T E _X とは	6
3.2	CJK パッケージと BXcjkatype パッケージ	7
4	X _Y _T E _X	8
4.1	X _Y _T E _X とは	8
4.2	ZXjatype パッケージ	8
4.3	IVS の利用	9
5	Lua _T E _X	10
5.1	Lua _T E _X とは	10
5.2	Lua _T E _X -ja パッケージ	11
6	fontspec パッケージ	13

1 導入

「 \TeX を使って日本語文書を作成する」と言った場合、「 \TeX 実習」で説明したように $\text{p}\text{\TeX}$ ・ $\text{up}\text{\TeX}$ を用いるのが一般的であろう。受講生の中には、「 $\text{p}\text{\TeX}$ ・ $\text{up}\text{\TeX}$ で十分。なぜ他の方法を気にするのか?」と思う方もいるかもしれない。しかし、あくまで筆者の私見だが、以下の 2 つの理由で重要である:

- 「 \TeX が入っている」と「 $\text{p}\text{\TeX}$ が利用できる」とは同義ではない。

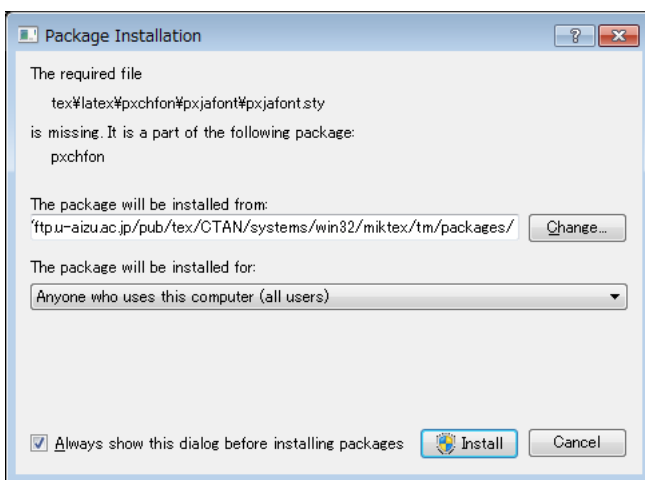
$\text{p}\text{\TeX}$ 自体は 1990 年代からあるが、それが「上流」の \TeX Live で利用できるようになったのは 2010 年というつい最近である。また、すぐ後に述べる $\text{MiK}\text{\TeX}$ という Windows 用の \TeX ディストリビューションでは、 $\text{p}\text{\TeX}$ は未だに含まれていない。

- 世界での \TeX の発展に $\text{p}\text{\TeX}$ は取り残されている

海外では、 $\text{pdf}\text{\TeX}$ (§3) などを利用した PDF 直接生成が当たり前になっている。さらには $\text{X}\text{\LaTeX}$ (§4), $\text{Lua}\text{\TeX}$ (§5) においては Unicode への本格的な対応や OpenType フォントの機能が利用できるようになっている。

一方、 $\text{p}\text{\TeX}$ は大雑把に言えば「 \TeX 82 を日本語拡張しただけ」である。実際には $\epsilon\text{-}\text{\TeX}$ 拡張がサポート ($\epsilon\text{-p}\text{\TeX}$) されていたり、 $\text{up}\text{\TeX}$ で「和文の部分」が Unicode 化されたりしているが、これ以上の機能拡張はもうあまり行われまいだろう。

例えば、 **$\text{MiK}\text{\TeX}$** という海外でよく使われている Windows 用の \TeX ディストリビューションがある。面白いのは、 $\text{MiK}\text{\TeX}$ のパッケージとしては存在するがシステムにまだインストールされていないファイルがタイプセット時に要求された場合、自動的に



のようなダイアログが出て、インターネットから取得することができることである。

しかし、 $\text{MiK}\text{\TeX}$ には $\text{p}\text{\LaTeX}$ が含まれていないので、日本語文書を作りたい場合は、本文書の残りで説明するような別の方法を用いないといけない。実は、本文書をつくるきっかけは、数年前の \TeX 実習における受講生の「 $\text{MiK}\text{\TeX}$ で日本語文書は作成できないのか」という質問であった。

2 予備知識： ϵ - $\text{T}_{\text{E}}\text{X}$ 拡張

表 1 に載せた 4 つのどの方法でも使えるのが、 ϵ - $\text{T}_{\text{E}}\text{X}$ 拡張である。これは The NTS team (Peter Breitenlohner et al.) によって開発された $\text{T}_{\text{E}}\text{X}82$ の拡張であり、一般利用者にとって特に有用なものを挙げると、次のようなものがある。

- `\numexpr 4-(3*2)+\count0` のように内部整数・長さなどを式の形で表現できる。
- 使用可能なレジスタが各種類ごとに 256 個から 32768 個まで激増した。

これにより、たくさんパッケージ読み込んだときに発生しうる

```
! No room for a new \count .
```

がほとんど起こらなくなった^{*5}。

- 数式中の「サイズが自動的に変わる括弧」に使われる `\left`・`\right` に加えて、`\middle` が利用できるようになった。例えば量子力学の Dirac 記法、例えば

$$\left\langle \phi \left| \frac{\partial^2}{\partial t^2} \right| \psi \right\rangle$$

を、

```
\left\langle \phi \middle| \frac{\partial^2}{\partial t^2} \middle| \psi \right\rangle
```

と簡単に組めるようになった^{*6}。

ϵ - $\text{T}_{\text{E}}\text{X}$ 拡張は現在開発されている様々な $\text{T}_{\text{E}}\text{X}$ の拡張において標準的な地位を占めており、 ϵ - $\text{T}_{\text{E}}\text{X}$ 拡張を要求されるパッケージは年々増加している。次期 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ である $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}3$ では、 ϵ - $\text{T}_{\text{E}}\text{X}$ 拡張を必要とするので、必須の拡張といえよう。

しかし心配することはない。 $\text{T}_{\text{E}}\text{X}$ Live 2011 以降では、意識しなくても標準で ϵ - $\text{T}_{\text{E}}\text{X}$ 拡張が有効になっているからである：

- `latex` と打つと、`dvi` 出力モード・ ϵ - $\text{T}_{\text{E}}\text{X}$ 拡張有効の `pdf $\text{T}_{\text{E}}\text{X}$` が起動する。
- `pdflatex` と打つと、PDF 出力モードの `pdf $\text{T}_{\text{E}}\text{X}$` が起動する。
- 日本語が処理できる `p $\text{T}_{\text{E}}\text{X}$` , `up $\text{T}_{\text{E}}\text{X}$` に関しても例外ではない。「 $\text{T}_{\text{E}}\text{X}$ 実習」の PDF ではごまかしてきたが、`platex`, `uplatex` と打って実際に起動されるプログラムは、 ϵ - $\text{T}_{\text{E}}\text{X}$ 拡張が取り込まれた `ϵ -p $\text{T}_{\text{E}}\text{X}$` , `$\epsilon$ -up $\text{T}_{\text{E}}\text{X}$` である：

```
> platex
This is e-pTeX, Version 3.14159265-p3.5-130605-2.6 (sjis) (TeX Live 2014
/W32TeX)
...
> uplatex
```

^{*5} 実際には、このエラーはレジスタ確保用のマクロが出しているため、 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ では `etex` パッケージなどを読み込むなどしてマクロ側で対応させる必要がある。

^{*6} なお、集合の内包的記法の時に `\middle` を使うときには周囲の空白に注意が必要である。
八登氏のブログ記事「`\middle` であり `\mid` であるもの」を参照すること。

This is e-upTeX, Version 3.14159265-p3.5-u1.11-130605-2.6 (utf8.uptex) (
TeX Live 2014/W32TeX)
...

3 pdf \TeX

3.1 pdf \TeX とは

pdf \TeX は, Hàn Thế Thành 氏による \TeX の拡張であり, 海外で主に利用されている. pdf \TeX 上で \LaTeX を動作させたものは pdf \LaTeX と呼ばれる. pdf \TeX には, 以下のような特徴がある:

- \TeX 82 や p \TeX が dvi ファイルを出力するのに対して, pdf \TeX は PDF ファイルを \TeX ソースから直接生成できる. これにより, PDF のさまざまな機能を直に利用できる.
- より高品質な組版を行う micro-typography という機能が搭載された. 例えば
 - 各行の両端を視覚的に揃えるため, いくつかの記号類を版面の外に少しだけ突き出させる.
 - 単語間の空白が小さく/大きくなりすぎるのを防ぐため, 各文字の幅をわずかに収縮/拡張させる.

といったものである. micro-typography については,

- Hàn Thế Thành, *Micro-typographic extensions to the \TeX typesetting system*, dissertation. <http://www.pragma-ade.com/pdfTeX/thesis.pdf>
- R. Schlicht, *The microtype package*. ([microtype パッケージ](#)のドキュメント)

に詳しい.

- ϵ - \TeX 拡張 (§2) も搭載されている.
- その他各種便利な機能, 例えば

乱数 `\pdfuniformdeviate` 等. アルゴリズムは「引き算法」($x_n = x_{n-55} - x_{n-24}$).

文字列比較 `\pdfstrcmp` により文字列の大小を比較できる. 次期 \LaTeX である $\text{\LaTeX}3$ では, これに相当する機能が ϵ - \TeX 拡張とともに必須になった.

PDF 中の絶対位置の取得 `\pdfsavepos` 等を利用する. 実際に用いる際には, 補助ファイル経由 (つまり, 2 回のタイプセットが必要) となるだろう.

実は, 単に `latex` とコマンドプロンプトで入力すると, 起動するプログラムは, dvi 出力モードの (ϵ - \TeX 拡張が有効な) pdf \TeX である:

```
> latex
This is pdfTeX, Version 3.14159265-2.6-1.40.14 (TeX Live 2014/W32TeX)
...
```

3.2 CJK パッケージと BXcjkjatype パッケージ

CJK パッケージは、欧文用の L^AT_EX で中国語・日本語・韓国語の入った文書を組版するためのパッケージである。CJK パッケージで利用可能な日本語フォントとしては ipaex-type1 というものがあり、それをインストールした上で次の cjktest.tex を UTF-8^{*7}で作り、

```
1 \documentclass{article}
2 \usepackage{CJK}
3 \begin{document}
4 \begin{CJK*}{UTF8}{ipxm}
5   日本語の明朝体と， {\CJKfamily{ipxg}ゴシック体}.
6   和文と alphabet の間の，「四分空き」は入らない.
7 \end{CJK*}
8 \end{document}
```

このソースファイルを

```
> pdflatex cjktest
```

でタイプセットすると、cjktest.pdf が直接生成される。しかし、結果を見ると、和文文字と欧文文字の間に入る（通常 1/4 全角の）空白が挿入されなかったり、全角コンマと開き鍵括弧の間が間延びしていたりと正しく組版されていない。

日本語がメインの文書を作成する場合、[T_EX Wiki 中のページ](#)によると、八登崇之氏による

- **BXjscls パッケージ**：「pL^AT_EX 2_ε用新ドキュメントクラス」からの派生
- **BXcjkjatype パッケージ**：日本語組版に適した CJK パッケージの設定と、上記の問題点の修正

を利用して

```
\documentclass[pdflatex,jadriver=standard]{bxjsarticle}
\begin{document}
...
\end{document}
```

のようにすればよい、とある。だが、残念ながら MikT_EX 2.9 では BXjscls のバージョンが古いためにこれではうまくいかず、

```
\documentclass{article} % 欧文用クラスファイル！
\usepackage[whole]{bxcjkjatype}
\begin{document}
...
\end{document}
```

のようにならないといけないようである。

^{*7} バイトオーダーマーク (BOM) をつけるとタイプセットに失敗するので、つけないように注意。

4 XeTeX

4.1 XeTeX とは

XeTeX (XeTeX) は、TeX の Unicode 拡張であるとともに、OpenType などの最新のフォント技術に統合させたものであり、次の特徴がある：

- pdfTeX と同様に、PDF ファイルを直接生成^{*8}し、また ϵ -TeX 拡張も搭載。
 - フォントの扱いが大幅に拡張された。
 - OS 側で使用出来るフォント、例えば「MS 明朝」、を直に扱える^{*9}。
 - **HarfBuzz**という OpenType レイアウトエンジンを採用しており、これによって OpenType フォント自身に含まれる合字・アクセント位置などの情報が利用できる。
- X_qLaTeX (X_qTeX 上で LaTeX を動かしたもの) 上でこれらの機能を扱うために作られたのが、fontspec パッケージ (§6) である。

日本語で読める XeTeX に関する情報としては、本文書の最初に挙げたもの以外に

- 八登崇之, 「X_qLaTeX で日本語する件について」,
<http://zrbabbler.sp.land.to/xelatex.html>
- 清水美樹, 「はじめての LaTeX」著者サポートページ中の「MiKTeX で XeTeX で日本語を」

などがある。なお、XeTeX で hyperref パッケージを使用する際には dvipdfmx オプションを指定してはならず、代わりに unicode オプションを指定する。

4.2 ZXjatype パッケージ

X_qTeX 自体は Unicode 対応しているので、原理的には fontspec パッケージで日本語フォントを指定すれば日本語は出る……のだが、標準の X_qLaTeX は「日本語の組み方を知らない」。

pLaTeX に匹敵する品質の日本語組版を行うには、**xeCJK パッケージ** (Wen Chang Sun 氏ら) や、それに日本語用の設定を追加する **ZXjatype パッケージ** (八登氏) を利用するのが良い。TeX Wiki 中の「xeCJK/ZXjatype」ページには、日本語の情報がある。

クラスファイルも先に紹介した八登氏作の BXjscls を用いて、**UTF-8** で以下のように記述するのが手取り早い：

```
1 \documentclass{bxjsarticle}% 日本語用クラスファイル
2 \usepackage{zxjatype}
3 \setjamainfont{IPA明朝}% \rmfamily に対応。本文など
4 \setjasansfont{IPAゴシック}% \sffamily に対応。見出し、強調など
5 \setjamonofont{IPAゴシック}% \ttfamily に対応。
6 \usepackage{xltextra}% XeLaTeX ロゴ
```

^{*8} 実際には、DVI を拡張した xdv 形式で出力し、それを xdvipdfmx によって PDF に変換している。

^{*9} 従来、TeX で使用出来るフォントは、OS 側で利用できるフォントとは全く独立であり、新たなフォントを TeX で扱うようにするためには面倒な作業が必要であった。


```

7 \begin{document}
8 こんにちは, \XeLaTeX の世界へ!
9 \[
10 \int_0^{\infty} e^{-x^2} \backslash, dx = \frac{\sqrt{\pi}}{2}.
11 \backslash]
12 \textsf{sans serif} に対してはゴシック体が使われる. }
13 誰かが, 「あっ」と言った.
14 \end{document}

```

タイプセットは

```
> xelatex xelatex_test.tex
```

のようにする. `xelatex_test.pdf` が生成されるはずである.

`ZXjatype` パッケージは内部で `fontspec` パッケージを読み込んでいるため, 欧文フォントもフォント名で指定できる.

```

1 \documentclass{bxjsarticle}
2 \usepackage{zxjatype}
3 \setjamainfont{IPA明朝}% \rmfamily に対応. 本文など
4 \setjasansfont{IPAゴシック}% \sffamily に対応. 見出し, 強調など
5 \setjamonofont{IPAゴシック}% \ttfamily に対応.
6 \setmainfont[Ligatures=TeX]{Times New Roman}% \rmfamily, Times 相当
7 \setsansfont[Ligatures=TeX]{Arial}% \sffamily, Helvetica 相当
8 \setmonofont[Ligatures=TeX]{Courier New}% \ttfamily, Courier 相当
9 \begin{document}
10 Times New RomanとIPA明朝.
11 \textsf{ArialとIPAゴシック. }
12 \end{document}

```

4.3 IVS の利用

IVS (Ideographic Variation Sequence) は, Unicode において同じコードに割り当てられてしまうような細かい字形の差を使い分けるための仕組みであり, 文字の直後に `U+E0100–U+E01EF` のいずれかを続けたものである.

`XYTeX` では利用するフォントが IVS に対応していれば, それ以外には何もしなくても IVS が利用できる. 例えば, `U+E01xy` を $\langle xy \rangle$ と表記した時,

```
\Large 東京都葛<01>飾区, 奈良県葛<00>城市
```

からは

東京都葛飾区, 奈良県葛城市

が得られ, きちんと字形の使い分けができています.

5 Lua_TE_X

5.1 Lua_TE_X とは

現在開発中の、pdf_TE_X 2.0 とでも言うべきものが **Lua_TE_X** である。いつものように、Lua_LA_TE_X は Lua_TE_X 上で L_AT_EX を動作させたものであり、また、Lua_TE_X で `hyperref` パッケージを使用する際には `dvipdfmx` オプションを指定してはならず、代わりに `unicode` オプションを指定する。

Lua_TE_X は、X_Y_TE_X と共に Unicode を直接扱え、**T_EX** ソースの文字コードは **UTF-8** であることが期待されている。Lua_TE_X の最大の特徴は内部にスクリプト言語 Lua が組み込まれていることで、例えば

```
$x=\cos x$ の解を数值的に計算すると、
$x=\backslash\mathrm{directlua}\{
  \mathrm{local}\ b,e=0,1
  \mathrm{while}\ (e-b)>1\mathrm{e-}15\ \mathrm{do}
    \mathrm{local}\ h=(e+b)/2
    \mathrm{if}\ h>\mathrm{math}.\cos(h)\ \mathrm{then}\ e=h\ \mathrm{else}\ b=h\ \mathrm{end}
  \mathrm{end}
  \mathrm{tex}.\mathrm{write}((e+b)/2)
\}$ である。
```

から

$x = \cos x$ の解を数值的に計算すると、 $x = 0.73908513321516$ である。

が得られる。途中「`\directlua{ ... }`」の中身が Lua で書かれたスクリプトであり、二分法を用いて $x = \cos x$ の数値解を求めている。

さらに、この Lua スクリプトを使い、従来では T_EX ソースではいじれなかった（プログラムとして T_EX を改造するしかなかった）内部の挙動をカスタマイズできるという特徴がある。以下のような活用例がある：

lua-visual-debug パッケージ 見た目ではわからないグルー（伸縮可能な空白）やカーン（幅一定の詰め物）などを PDF 上で確認できる。

fontspec パッケージ と **luaotfload パッケージ** fontspec パッケージは X_Y_LA_TE_X だけではなく Lua_LA_TE_X でも使用可能である。

Lua_TE_X 本体には X_Y_TE_X のような OpenType レイアウトエンジンは搭載されていないが、それに相当する部分が luaotfload パッケージである。luaotfload パッケージは、巨大な Lua スクリプトを用いて、Lua_TE_X において OpenType の各種機能を扱えるようにしている。

Lua_TE_X-ja パッケージ p_TE_X 並み、あるいはそれ以上の品質・自由度の日本語組版を目指したパッケージであり、以下で詳しく解説する。

5.2 LuaTeX-jā パッケージ

LuaTeX-jā は、pTeX と同等、あるいはそれ以上の品質の日本語組版を TeX マクロと Lua スクリプトによって、つまりプログラム側に変更を加えずに、実現しようという試みである。

- pTeX と完全な互換は目指さない（そもそも不可能）。pTeX の仕様が不便・不都合だと感じた場合には、LuaTeX-jā では積極的に改める。
- また、LuaTeX 側の都合により、現在は横組みのみの対応となっている。
- しかし、組版速度は pTeX や XeTeX などより遅い。徐々に高速化をしているのだが……。

LuaTeX-jā を利用して日本語文書を作るには、pLaTeX 2_ε 新ドキュメントクラスを LuaTeX-jā 用に改変した ltjsarticle, ltjsbook をクラスファイルとして使うのが楽である^{*10}。例えば UTF-8 で以下の lualatex_test.tex を記述し、

```
1 \documentclass{ltjsarticle}
2 \begin{document}
3 ようこそ、Lua\TeX の世界へ！
4 \[
5 \int_0^{\infty} e^{-x^2} \, dx = \frac{\sqrt{\pi}}{2}.
6 \]
7 ちょっとチェックしちゃった。
8
9 $\pi \simeq \directlua{tex.print(math.pi())}$
10 \end{document}
```

その後

```
> lualatex lualatex_test.tex
```

でタイプセットすると、lualatex_test.pdf が生成されるはずである。



TeX Live ではこれでよいが、MiKTeX では実はうまくいかない。

LuaTeX-jā は、fontspec パッケージも自動で読み込まず、また標準では和文フォントは埋め込まない（pLaTeX の標準と同じ）ことになっている。だが、「TeX 実習」で述べたように、表示の面から PDF 内に日本語フォントを埋め込む方がよい場合がある。LuaTeX-jā に関しては、もう一つの理由からも和文フォント埋め込みを推奨したい。それは、フォント内部に含まれている各種の組版情報を利用するためである。

和文フォントを埋め込む設定にする場合、luatexja-fontspec パッケージを読み込むのが簡便である。このパッケージを読み込むと、fontspec パッケージとその和文への拡張が使用可能となる：

```
1 \documentclass{ltjsarticle}
2 \usepackage{luatexja-fontspec}
3 \setmainfont{IPAMincho}% \rmfamily に対応。本文など
```

^{*10} beamer 等の欧文用クラスファイルを使う場合はブリアンブルに\usepackage{luatexja}を挿入する必要があるが、このltjsarticleクラスでは自動的に読み込まれるので不要である。

```

4 \setsansfont{IPAGothic}% \sffamily に対応．見出し，強調など
5 \setmainfont[Ligatures=TeX]{Times New Roman}% \rmfamily, Times 相当
6 \setsansfont[Ligatures=TeX]{Arial}% \sffamily, Helvetica 相当
7 \setmonofont[Ligatures=TeX]{Courier New}% \ttfamily, Courier 相当
8 \begin{document}
9 Times New RomanとIPA明朝．
10 \textsf{ArialとIPAゴシック．}
11 \end{document}

```

§4.2 で述べた Xe_{La}TeX と Xe_{La}TeX パッケージを用いた例とは，日本語フォントを指定する 3, 4 行目の部分が若干異なっている．

- 命令名称が異なっている．
- Lua_{La}TeX では，「IPA 明朝」といった日本語のフォント名は利用できないようである．そのため，ここでは“IPAMincho”という英語でのフォント名を使っている．
- \setmonofont, \setjamonofont に対応する命令は存在しない．これは元々 p_{La}TeX で標準に用いる書体が明朝・ゴシックの 2 つだけだったことによる^{*11}．

ちなみに，Lua_{La}TeX 本体は IVS (§4.3) にはまだ未対応のようであるが，現行の Lua_{La}TeX-ja では

```

\usepackage{luatexja-otf}
\directlua{luatexja.otf.enable_ivs()}

```

をプリアンブルに書き加えることで IVS が使用可能になる．

^{*11} 開発版では一定の条件下で \setmonofont が使用可能になっている．

6 fontspec パッケージ

[fontspec パッケージ](#)は、 $\text{X}\text{\LaTeX}$ と $\text{Lua}\text{\TeX}$ において簡単に OpenType フォントや TrueType フォントを扱うためのパッケージである。§5.1で述べたように、 $\text{Lua}\text{\TeX}$ では内部処理に [luaotfload パッケージ](#)を使っている。

fontspec パッケージで標準のフォントを指定するには次の 3 命令を使う：

```
\setmainfont[<option>]{<font>} \textrm, \rmfamily で指定される, 標準のローマン体.  
\setsansfont[<option>]{<font>} \textsf, \sffamily で指定される, 標準のサンセリフ体.  
\setmonofont[<option>]{<font>} \texttt, \ttfamily で指定される, 標準のタイプライタ体.  
    \verb や verbatim 環境でもこのフォントが用いられる.
```

何も指定しないと、それぞれ順に “Latin Modern Roman”, “Latin Modern Sans”, “Latin Modern Mono” が使われる。これらは $\text{T}\text{\E}\text{X}$ 標準の Computer Modern をベースにした OpenType フォントである。

また、 $\text{X}\text{\LaTeX}$ における日本語組版に使う ZXjatype パッケージや、 $\text{Lua}\text{\TeX}$ における $\text{Lua}\text{\TeX}$ -ja パッケージには、上記 3 命令の「日本語版」の命令が準備されている（各々の解説を参照）。

上記 *<option>* に指定できる設定はかなり多い。詳細は fontspec パッケージのマニュアルや以前に挙げた八登氏の「 [\$\text{X}\text{\LaTeX}\$ で日本語する件について](#)」中の記述を参照して欲しい。