

# TeX Live 2016 における pTeX 系列のプリミティブ

北川 弘典

2016/11/05

TeX ユーザの集い 2016

# 本発表の概要

- T<sub>E</sub>X Live 2016 の pT<sub>E</sub>X 系列 ((ε-)(u)pT<sub>E</sub>X) で利用可能なプリミティブを解説.
- 全部を扱うことは時間的に不可能

省略したものは帯色が緑のスライドへ



# 本発表の概要

- T<sub>E</sub>X Live 2016 の pT<sub>E</sub>X 系列 ((ε-)(u)pT<sub>E</sub>X) で利用可能なプリミティブを解説.
- 全部を扱うことは時間的に不可能

**強調するが、この配色のスライドは発表では省略**





ASCIIMW 公式<sup>1</sup>内にいくつかのプリミティブの説明があるが、それ以外に……

- 省略 組方向 dtou
- 組方向判定 ~2011
- 数式中のベースライン補正 2016
- 省略 \kcatcode
- 省略 \mathchoice の不具合修正 2016

---

<sup>1</sup> 「アスキー日本語 TeX (pTeX)」, <http://ascii.asciimw.jp/pb/ptex/>

# 組方向 dtou

	横組	縦組	「dtou 方向」
組方向変更	<code>\yoko</code>	<code>\tate</code>	<code>\dtou</code>
組版例	……共に ASCII、シフト JIS の組み合わせを用いた。ただし、DVI ファイル内の漢字コードだけは、JIS コード……	……共に ASCII、シフト JIS の組み合わせを用いた。ただし、DVI ファイル内の漢字コードだけは、JIS コード……	……共に ASCII、シフト JIS の組み合わせを用いた。ただし、DVI ファイル内の漢字コードだけは、JIS コード……
字送り方向	→ (右向き)	↓ (下向き)	↑ (上向き)
行送り方向	↓ (下向き)	← (左向き)	→ (右向き)
和文フォント	横組用	縦組用	横組用 (回転)

## ■ 現在の組方向を判定<sup>2</sup>

- ▶ `\ifydir` (横組)
- ▶ `\iftdir` (縦組)
- ▶ `\ifmdir` (数式ディレクション)

Hy  
あ  
い  
う

Hy  
あ  
い  
う

縦組

縦数式

## ■ `\box <num>` の組方向を判定

- ▶ `\ifybox <num>`, `\iftbox <num>`

横組

縦組

<sup>2</sup> 「 $\TeX$  の出版への応用 — 縦組み機能の組み込み —」には記述あり。

## ■ 現在の組方向を判定<sup>2</sup>

- ▶ `\ifydir` (横組)
- ▶ `\iftdir` (縦組)
- ▶ `\ifmdir` (数式ディレクション)
- ▶ `\ifddir` (dtou 方向) 2011

Hy  
あ  
い  
う

Hy  
あ  
い  
う

縦組    縦数式

## ■ `\box <num>` の組方向を判定

- ▶ `\ifybox <num>`, `\iftbox <num>`, `\ifdbbox <num>`  
横組                      縦組                      dtou 方向 2011

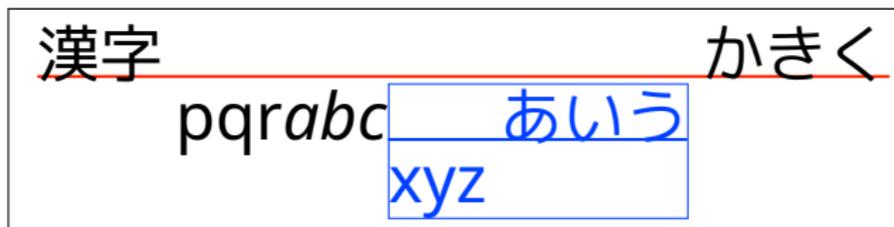
<sup>2</sup> 「 $\TeX$  の出版への応用 — 縦組み機能の組み込み —」には記述あり。

# 数式中のベースライン補正（従来）

**従来** 文中数式は全体にベースライン補正がかかる  
→数式中のボックス中の欧文文字は二重に補正される

```
\ybaselineshift=1zw
```

```
漢字pqr$abc\hbox{xyzあいう}$かきく
```



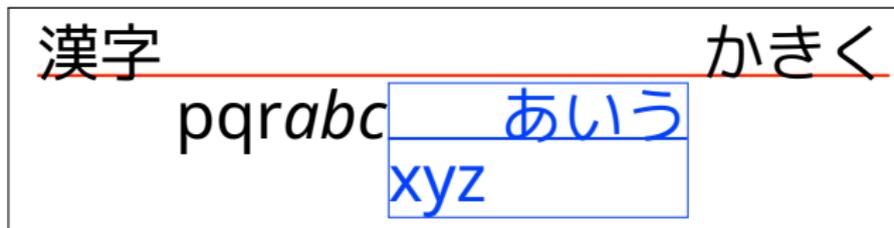
**TL2016** 数式中のボックスを**逆方向に補正**.

`\textbaselineshiftfactor` で補正割合指定.

`\textbaselineshiftfactor=0`    % 従来の挙動

`\ybaselineshift=1zw`

漢字pqr\$abc\hbox{xyzあいう}\$かきく



省略 `\scriptstyle, \scriptscriptstyle` 用の命令も

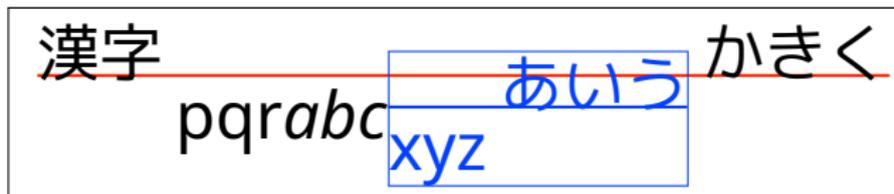
**TL2016** 数式中のボックスを**逆方向に補正**.

`\textbaselineshiftfactor` で補正割合指定.

```
\textbaselineshiftfactor=500 % 0.5倍
```

```
\ybaselineshift=1zw
```

```
漢字pqr$abc\hbox{xyzあいう}$かきく
```



**省略** `\scriptstyle`, `\scriptscriptstyle` 用の命令も



**TL2016** 数式中のボックスを**逆方向に補正**.

`\textbaselineshiftfactor` で補正割合指定.

`\textbaselineshiftfactor=1000` % 初期値, 「1倍」

`\ybaselineshift=1zw`

漢字pqr\$abc\hbox{xyzあいう}\$かきく



**省略** `\scriptstyle`, `\scriptscriptstyle` 用の命令も

# `\scriptbaselineshiftfactor`

初期値 : 700 (0.7 倍)

→ `\scriptstyle` 下の明示的ボックスでは  
ベースライン補正が 0.7 倍であるのがよい

1 `\ybaselineshift=1zw`

2 一 : `pqr$\scriptstyle abc\text{xyzあいう}$`

3 二 : `pqr$\scriptstyle abc\text{%`

4 `\ybaselineshift=0.7\ybaselineshift xyzあいう}$`

`\scriptscriptbaselineshiftfactor` の  
初期値は 500 (0.5 倍).

一 :	<code>pqrabc<sub>xyz</sub></code>	あいう
二 :	<code>pqrabcxyz</code>	あいう

# pTeX における \kcatcode

pTeX でも \kcatcode 命令は使用可能.

- アスキー版 pTeX では, **内部コード**の上位バイトごとに設定可能

内部コードがシフトJIS のときは,  $(2n - 1)$  区と  $2n$  区が同じ \kcatcode を持つことになる.

- T<sub>E</sub>X Live 中の pTeX では, **JIS コード**の上位バイトごと (区ごと) に設定可能

# `\mathchoice` の不具合修正

寺田さんにより，次のソースが segmentation fault  
で落ちることが報告 (2015/09/10) :

```
1  $\mathchoice{\あ}{}{}{}{\$}$   
2 \bye
```

**原因** `\mathchoice` は，各数式スタイルごとに  
全部組んだ後適切なスタイルの結果のみ残す  
→ 不要な結果を廃棄する過程に不具合があった

- 省略  $\epsilon$ -TeX の機能全部

$\epsilon$ -TeX のマニュアル (etex\_man.pdf) 参照

- 省略 無限大レベル fi の追加

2011

- 256 個の数式ファミリのサポート

2011

- pdfTeX の一部機能

- `\lastnodechar`

2015

- `\epTeXinputencoding`

2016

$\Omega$  と同様に、有限と fil の間に無限大レベル fi を新設

- \hfил 等の fi の対応物は  $\epsilon$ -pT<sub>E</sub>X 独自
  - ▶ \hfi, \vfi, \pagefistretch
- \gluestretchorder の値

エンジン	有限	fi	fil	fill	filll
$\epsilon$ -T <sub>E</sub> X, pdfT <sub>E</sub> X, X <sub>3</sub> T <sub>E</sub> X	0	—	1	2	3
X <sub>2</sub> , LuaT <sub>E</sub> X	0	1	2	3	4
<b><math>\epsilon</math>-pT<sub>E</sub>X</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>

当初は  $\epsilon$ -T<sub>E</sub>X に値を合わせたが……どうするのがよい？

T<sub>E</sub>X Live 2016 以降は正式に  $\epsilon$ -pT<sub>E</sub>X の一部に  
(それまで「FAM256 パッチ」と呼ばれ別扱い)

▶ `\omathcode`, `\odelcode`, `\omathchar`, `\odelimiter`,  
`\omathaccent`, `\omathchardef`, `\oradical`

■ `\odelcode`, `\odelimiter` の設定では

```
\odelimiter "4120034 "130045
```

のように整数を 2 つ並べる

■ `\odelcode` の値**取得**は不可能

(51 bit 整数をどう返す)

# 256 個の数式ファミリのサポート (2)

2015 年以降の L<sup>A</sup>T<sub>E</sub>X では……

- **数式英字フォント**<sup>3</sup>の 16 種以上同時使用は容易.  
以下を記述する (or 10/20 以降の `expl2e.sty`).

1 `\makeatletter`

2 `\mathchardef\@mathgroup@top=256`

3 `\makeatother`

- **数式記号フォント**<sup>4</sup>については  
それほど楽ではない (マクロ再定義が必要)

---

<sup>3</sup>`\DeclareMathAlphabet` で定義される `\math??`.

<sup>4</sup>`\DeclareSymbolFont` で定義.



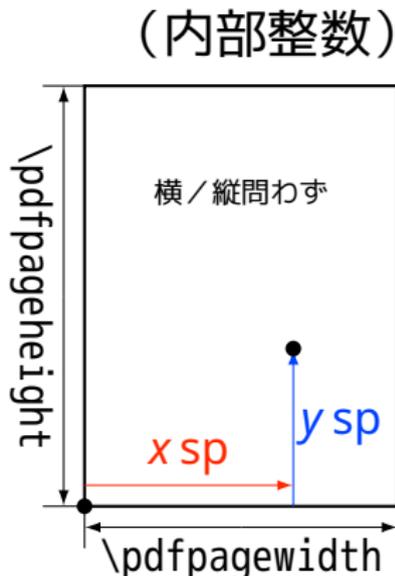
pdfT<sub>E</sub>X の機能が要求されるパッケージが近年増加  
→それに伴い, ε-pT<sub>E</sub>X にも少しずつ実装

- L<sup>A</sup>T<sub>E</sub>X3 省略 \pdfstrcmp 2011
- linegoal 省略 \pdfsavepos 他 2011
- standalone 省略 \pdffilemoddate 他 2014
- bmpsizе 省略 \pdffiledump 2014
- T 氏のリクエスト 省略 \pdfshellescape 2015
- movie15 \pdfmdfivesum 2016
- L<sup>A</sup>T<sub>E</sub>X3 周り \pdfprimitive 他 2016

.....

- ▶ `\pdfstrcmp <text1> <text2>`
  - `<text1>`, `<text2>` をバイト列として辞書式比較
    - `<text1> <text2>` なら `-1`
    - `<text1> = <text2>` なら `0`
    - `<text1> > <text2>` なら `1`
  - `<text>` 中の和文文字は UTF-8 で符号化  
→ 下は内部コードに関わらず `0`
    - 1 `\pdfstrcmp{あabc}{^^e3^^81^^82abc}`

- ▶ `\pdfsavepos`  
実行箇所の絶対位置を `\shipout` 時に保存
- ▶ `\pdflastxpos`, `\pdflastypos`  
`\pdfsavepos` で保存した座標
  - `\pdflastxpos` : 図中の  $x$
  - `\pdflastypos` : 図中の  $y$
- ▶ `\pdfpagewidth`, `\pdfpageheight`  
原点位置<sup>5</sup>を指定する内部長さ



<sup>5</sup>pdfTeX では, 出力ページの寸法もこれらで指定.

- ▶ \pdfcreationdate  
dvi 生成時刻を PDF 標準日付形式  
(例: D:20161108202752+09'00') で返す  
→ T<sub>E</sub>X82 ではできなかった**秒**の取得に有用
- ▶ \pdffilemoddate *<filename>*  
*<filename>* の更新時刻を \pdfcreationdate と  
同じ書式の文字列で返す
- ▶ \pdffilesize *<filename>*  
*<filename>* のファイルサイズを返す

- ▶ \pdffiledump [offset <offset>]  
length <length> <filename>

ファイル <filename> 中の <length> バイトを取得

- 出力形式は大文字 16 進文字列
- <offset> で開始位置を指定可能

```
1 \pdffiledump offset 1000 length 10 {\jobname.tex}
2 % ==> A4E38389E381B87D0A20
```

## 外部コマンドの実行 (shell-escape, \write18)

の状態を示す読み取り専用整数

- 0: 利用不可能
- 1: 利用可能
- 2: **制限付き**で利用可能

「安全」と認められたもの<sup>6</sup>のみ実行可能T<sub>E</sub>X Live 2015 から extractbb が加わる

pmpost は含まれていない

---

<sup>6</sup>texmf.cnf 中の shell\_escape\_commands で指定.

▶ \pdfmdfivesum  $\langle text \rangle$

引数  $\langle text \rangle$  の MD5 チェックサムを計算

- $\langle text \rangle$  中の和文文字は UTF-8 で符号化

→ 下の 2 行は内部コードに関わらず同じ結果

```
1 \pdfmdfivesum{あabc}
```

```
2 \pdfmdfivesum{^^e3^^81^^82abc}
```

▶ \pdfmdfivesum file  $\langle filename \rangle$

ファイル  $\langle filename \rangle$  の MD5 チェックサムを計算

L<sup>A</sup>T<sub>E</sub>X3 2015/07/15 で突如前提扱いに！

→議論の末, L<sup>A</sup>T<sub>E</sub>X3 2015/07/28 では使わないように

▶ `\pdfprimitive <cs>`

■ `<cs>` と同名のプリミティブがあれば,  
そのプリミティブを実行

■ そうでなければエラー

! Missing primitive name.

▶ `\ifpdfprimitive <cs>`

`<cs>` の意味が同名のプリミティブであるときに真

現在構築中のリストの最後が文字ノードであれば、  
その内部コードを返す (読み取り専用整数)

- 最後が文字ノードでなければ -1 を返す
- $\epsilon$ -pTeX が自動的に挿入する、メトリック由来の  
グルーや禁則用ペナルティは透過

- 1 あ「\message{\the\lastnodechar}% ==> 41430
- 2 あ』\message{\the\lastnodechar}% ==> 41431

- 合字は最後の構成要素を返す

- 1 ffi\message{\the\lastnodechar}% ==> 105

# \lastnodechar の応用例

- 1 これは\textgt{『ほげ党宣言』}の\\ % OK
- 2 これは, \textgt{『ほげ党宣言』} (ホゲ\\ % NG
- 3 これは\mytextgt{『ほげ党宣言』}の\\ % OK
- 4 これは, \mytextgt{『ほげ党宣言』} (ホゲ % OK

これは『ほげ党宣言』の  
これは, 『ほげ党宣言』 (ホゲ  
これは『ほげ党宣言』の  
これは, 『ほげ党宣言』 (ホゲ

ここで使った \mytextgt の定義は以下のページにある :

<https://ja.osdn.net/projects/epTeX/wiki/lastnodechar>



▶ \epTeXinputencoding  $\langle encoding \rangle$

現在のファイルの文字コードを（次の行から）変更

■ 指定可能な  $\langle encoding \rangle$  :

```
default jis          euc      sjis      utf8  uptex
          BINARY      EUC-JP  Shift_JIS UTF-8
          ISO-2022-JP
```

■ \input 等で読み込まれるファイルには影響なし

■ ログやファイルへの出力には影響なし

（上記2点が $\TeX$ の\kanjifiletypeとの違い）

## 参考：\kanjifiletype 他 (jT<sub>E</sub>X)

- ▶ \kanjifiletype (内部整数)  
ファイルとの入出力, ログへの出力の漢字コード
  - \input 等で読み込まれるファイルにも影響
  - ログの出力コード：「ASCII → 漢字」ごと
  - ファイルへの出力コード：\openout 時
- ▶ \kanjiterminaltype (内部整数)  
端末との入出力の漢字コード
  - 入力コードは各行ごとに更新.



TeX Live 2012 に取り込まれて以降, upTeX エンジンの変更は Unicode 更新への対応がほとんど.

- ▶ `\kchar <num>, \kchardef<cs>=<num>`
- ▶ `\ucs <num>`
- ▶ `\enablecjktoken, \disablecjktoken,`  
`\forcecjktoken`

# `\kchar`, `\kchardef`, `\ucs`

- ▶ `\kchar`  $\langle num \rangle$ , `\kchardef`  $\langle cs \rangle = \langle num \rangle$

文字コード  $\langle num \rangle$  の値に関わらず, 常に和文動作

`\char`, `\chardef` は文字コードが 256 以上のとき和文動作

- ▶ `\ucs`  $\langle num \rangle$

Unicode から内部コードへの変換

- 内部コードが `uptex` なら, 単なる恒等写像

- それ以外の場合は JIS X 0208 内の文字

(のコード) のみ正常動作

- ▶ `\enablecjktoken`  
`\kcatcode` (15 : 欧文, 16-19 : CJK) に従う
- ▶ `\disablecjktoken`  
常に欧文 (単なるバイト列) 扱い

# U+0080 以降の Unicode 文字の扱い

- ▶ `\enablecjktoken`  
`\kcatcode` (15 : 欧文, 16-19 : CJK) に従う
- ▶ `\disablecjktoken`  
常に欧文 (単なるバイト列) 扱い
- ▶ `\forcecjktoken`  
(`\kcatcode` に関わらず) 常に和文扱い  
`\kcatcode` が 15 の文字は「その他和文」(18) 扱い
  - 1 `\forcecjktoken\kcatcode`西=15`
  - 2 `\message{\the\kcatcode`@ % ==> 18`
  - 3 `\ifcat 西@ T\else F\fi}% ==> T`



- 省略 `\pdfsavepos` の原点位置の不具合修正
- 省略 `\pdfcreationdate` の初期化タイミング
- `\catcode` 他の引数
- 数式ディレクション

# \pdfsavepos の原点位置 (1)

ページサイズの指定 = **\pdfsavepos の原点位置**の指定

- ▶ \pdfpagewidth : ページの物理的な横幅
- ▶ \pdfpageheight : ページの物理的な縦幅

縦組時にこの仕様通りに動かなかったのを修正.

(以下は阿部さんによるテストソース)

```
1 \documentclass{tarticle}
2 \pdfpageheight=297mm % 手動設定必要
3 \usepackage[savepos]{zref}
4 \begin{document}
5 \zsavepos{label}\framebox[\zposy{label}sp]{}
6 \end{document}
```

## `\pdfsavepos` の原点位置 (2)

ページサイズの指定 = `\pdfsavepos` の原点位置の指定

- ▶ `\pdfpagewidth` : ページの物理的な横幅
- ▶ `\pdfpageheight` : ページの物理的な縦幅
- これらは生成する DVI には何の影響も与えない
- 初期値は両者とも 0. この場合, `\shipout` したボックスの寸法と `\hoffset`, `\voffset` から自動的に「現在のページサイズ」を設定

# \pdfcreationdate の初期化タイミング

pdfT<sub>E</sub>X, LuaT<sub>E</sub>X<sup>7</sup> エンジン起動時

ε-pT<sub>E</sub>X \pdfcreationdate の初回実行時

→以下のソースで挙動の違い

```
1 \loop\ifnum\count0<10000000 % 無意味に  
2 \advance\count0 1\relax % ループさせて  
3 \repeat % 時間つぶし  
4 \message{\pdfcreationdate}\bye
```

**r42381 (10/30) で pdfT<sub>E</sub>X 等と同じ挙動に修正**

---

<sup>7</sup>LuaT<sub>E</sub>X では \pdffeedback creationdate と呼び出し方が変更された.

# `\catcode` $\langle 2\text{-byte num} \rangle$ 等

	取得	設定
pTeX-2.1.4 以降 <sup>1</sup>	上位バイト	未チェック
TeX Live 2012 以降	上位バイト	上位バイト
<b>r41998 (9/6) 以降</b>	<b>禁止</b>	<b>禁止</b>

## ■ 上位バイト：

引数の上位バイトに対する操作として扱う。

```
1 \catcode"E0=1 \message{\the\catcode"E0E1}% ==> 1
```

## ■ 未チェック：引数によっては落ちる

<sup>1</sup>それ以前は遡ることができなかった。

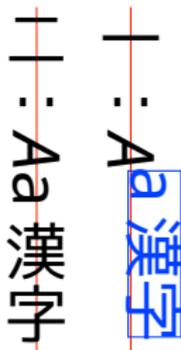
# \textbaselineshiftfactor の不具合

- 1 \noindent
- 2 一： $\$A\hbox{a漢字}\$\\$
- 3 二： $\$A\hbox{\tate a漢字}\$$

2行目の  $\hbox{a漢字}$  内の「a」は、  
数式内の「A」に比べて次の量だけ上がる：

$$\frac{\text{\textbaselineshiftfactor}}{1000} \cdot \text{\tbaselineshift} - \text{\ybaselineshift}$$

→縦組での array 環境, cases 環境等で不具合



- 1 \noindent
- 2 一： $\$A\hbox{a漢字}\$\\$
- 3 二： $\$A\hbox{\tate a漢字}\$$

一：Aa漢字  
二：Aa漢字

ボックスが数式ディレクションか否か

に関わらず，**欧文の垂直位置が揃うように**

(r42041 (9/12))

**副産物**（従来は不可能）

▶  $\ifmbox\langle num\rangle$

$\box\langle num\rangle$  が数式ディレクションか否か判定

# 詳細な仕様

周囲の組方向とボックスの組方向が一致した時のみ

「逆方向の補正」を実施

- 但し，数式ディレクションか否かは一致しなくても良い

補正量は  $ab/1000$

- $$a = \begin{cases} \scriptbaselineshiftfactor & (\scriptstyle) \\ \scriptscriptbaselineshiftfactor & (\scriptscriptstyle) \\ \textbaselineshiftfactor & (\text{otherwise}) \end{cases}$$

- 周囲が縦組か縦数式で，かつボックスが通常の縦組のとき

$$b = \text{tbaselineshift}$$

- それ以外の場合， $b = \text{ybaselineshift}$

# まとめ

## Extra primitives in $(\varepsilon)$ - $(u)$ pT $\TeX$

```
\autospacing \autoxspacing \dtou \euc \ifdbbox \ifddir \ifmbox \ifmdir \iftbox \iftdir
\ifybox \ifydir \inhibitglue \inhibitxspcode \jcharwidowpenalty \jfam \jfont \jis
\kanjiskip \kansuji \kansujichar \kcatcode \kuten \noautospacing \noautoxspacing
\postbreakpenalty \prebreakpenalty \scriptbaselineshiftfactor
\scriptscriptbaselineshiftfactor \showmode \sjis \tate \tbaselineshift
\textbaselineshiftfactor \tfont \xkanjiskip \xspcode \ybaselineshift \yoko
```

## Extra primitives in $\varepsilon$ - $(u)$ pT $\TeX$ (underlined: $\varepsilon$ -pT $\TeX$ original)

```
\epTeXinputencoding \hfi \ifpdfprimitive \lastnodechar \odelcode \odelimiter \omathaccent
\omathchar \omathchardef \omathcode \oradical \pagefistretch \pdfcreationdate \pdffiledump
\pdffilemoddate \pdffilesizes \pdflastxpos \pdflastypos \pdfmdfivesum \pdfpageheight
\pdfpagewidth \pdfprimitive \pdfsavepos \pdfshellescape \pdfstrcmp \vfi
```

## Extra primitives in $(\varepsilon)$ -upT $\TeX$

```
\disablecjktoken \enablecjktoken \forcecjktoken \kchar \kchardef \ucs
```

(r42234, 2016-10-09 08:18:34)