

# T<sub>E</sub>X Live 2016 の新しい pL<sub>A</sub>T<sub>E</sub>X

---

2016 年 11 月 5 日

T<sub>E</sub>X ユーザの集い 2016

山下 弘展 (Hironobu Yamashita)

Twitter: [@aminophen](https://twitter.com/aminophen)

# **T<sub>E</sub>X Live 2015 (最終版)**

```
$ platex x
This is e-pTeX, Version 3.14159265-p3.6-141210-2.6 (utf8.euc)
  (TeX Live 2015) (preloaded format=platex)
restricted \write18 enabled.
entering extended mode
(/usr/local/texlive/2015/texmf-dist/tex/latex/tools/x.tex
pLaTeX2e <2006/11/10> (based on LaTeX2e <2016/03/31> patch level 0)
Babel <3.9q> and hyphenation patterns for 81 language(s) loaded.
```

## **T<sub>E</sub>X Live 2015** (最終版)

```
$ platex x
This is e-pTeX, Version 3.14159265-p3.6-141210-2.6 (utf8.euc)
  (TeX Live 2015) (preloaded format=platex)
  restricted \write18 enabled.
entering extended mode
(/usr/local/texlive/2015/texmf-dist/tex/latex/tools/x.tex
pLaTeX2e <2006/11/10> (based on LaTeX2e <2016/03/31> patch level 0)
Babel <3.9q> and hyphenation patterns for 81 language(s) loaded.
```

## **T<sub>E</sub>X Live 2016** (2016 年 11 月 5 日現在の最新版)

```
$ platex x
This is e-pTeX, Version 3.14159265-p3.7-160201-2.6 (utf8.euc)
  (TeX Live 2016) (preloaded format=platex)
  restricted \write18 enabled.
entering extended mode
(/usr/local/texlive/2016/texmf-dist/tex/latex/tools/x.tex
pLaTeX2e <2016/09/08> (based on LaTeX2e <2016/03/31> patch level 3)
Babel <3.9r> and hyphenation patterns for 83 language(s) loaded.
```

# 1 L<sup>A</sup>T<sub>E</sub>X のこれまで

---

1985.08 L<sup>A</sup>T<sub>E</sub>X 2.09 (Leslie Lamport)

↑これが Lamport 氏による最後の版<sup>†1</sup>

**1989.08.21** (TUG meeting, Stanford)

Lamport 氏が, 以降の L<sup>A</sup>T<sub>E</sub>X のメンテナンスと開発を L<sup>A</sup>T<sub>E</sub>X 3 Team に引き継ぐことに同意

**1994.06 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> (L<sup>A</sup>T<sub>E</sub>X 3 Team)**

- スタイルファイル → 文書クラスとパッケージへ分離
  - \documentstyle から \documentclass へ
  - \usepackage の新設
- NFSS2 (New Font Selection Scheme release 2, 1993) を採用
  - これらはもはや現在では「あたりまえ」となっている.

---

<sup>†1</sup> L<sup>A</sup>T<sub>E</sub>X 2.09 の呼称はあとになってから定着したもの

# 1 L<sup>A</sup>T<sub>E</sub>X のこれまで

---

1985.08 L<sup>A</sup>T<sub>E</sub>X 2.09 (Leslie Lamport)

1994.06 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> (L<sup>A</sup>T<sub>E</sub>X 3 Team)

現在の“あたりまえ”が出来た一方で、不都合な点も

- オリジナルの T<sub>E</sub>X の資源不足
- 式による計算, デバッグ機能の需要

**1998.02    ε-T<sub>E</sub>X version 2.0<sup>†2</sup>** (The N<sub>T</sub>S Team)

- レジスタが 256 個から 32768 個へ
- `\dimexpr` などの計算用の primitive
- デバッグ用の primitive を多数実装

**2003.12    L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> の ε-T<sub>E</sub>X での利用を推奨<sup>†3</sup>** ← 必須とはしない

---

<sup>†2</sup> ε-T<sub>E</sub>X Manual (`$TEXMF/doc/etex/base/etex_man.pdf`)

<sup>†3</sup> L<sup>A</sup>T<sub>E</sub>X News Issue 16 (`$TEXMF/source/latex/base/ltnews16.tex`)

# 1 L<sup>A</sup>T<sub>E</sub>X のこれまで

---

1985.08 L<sup>A</sup>T<sub>E</sub>X 2.09 (Leslie Lamport)

1994.06 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> (L<sup>A</sup>T<sub>E</sub>X 3 Team)

- カーネルの変更は最小限にとどめる
- 互換性を損なう修正は fixltx2e パッケージで提供

## いつまでたってもデフォルトが “良く” ならない

- $\epsilon$ -T<sub>E</sub>X 拡張を利用したければ etex パッケージを利用
  - 段組み時のフロートの順序
  - `\addpenalty` や `\setlength` の定義
  - フloat環境内の最初の単語のハイフネーション
- ... etc.

# 1 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ のこれまで

---

1985.08  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  2.09 (Leslie Lamport)

1994.06  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\varepsilon}$  ( $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  3 Team)

- カーネルの変更は最小限にとどめる
- 互換性を損なう修正は `fixltx2e` パッケージで提供



**2015.01  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\varepsilon}$  2015/01/01 大規模改修<sup>†4</sup>**

- $\varepsilon$ - $\text{T}_{\text{E}}\text{X}$  拡張が利用可能ならば活用
- 修正が必要な場合はカーネルを直接修正する
- 過去のカーネルをエミュレートする `latexrelease` パッケージの新設
- `fixltx2e` パッケージの廃止

→ 2016 年 11 月現在,  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\varepsilon}$  2016/03/31 Patch level 3

---

<sup>†4</sup>  $\text{T}_{\text{E}}\text{X}$  Live 2015 以降に反映；[怒涛の Change History \(forum:1558\)](#)

## 2 pL<sup>A</sup>T<sub>E</sub>X のこれまで

---

1980s      pL<sup>A</sup>T<sub>E</sub>X 2.09 (ASCII Corporation)

**1995.11      pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> (ASCII Corporation)**

2006.11      アスキーによる現時点で最後の版

pT<sub>E</sub>X は日本ローカルのまま, UTF-8 入力不可,  $\epsilon$ -T<sub>E</sub>X 拡張なし

- teT<sub>E</sub>X ベースの ptetex や後継の ptexlive による配布
- 入力文字コードとして UTF-8 に対応
- $\epsilon$ -T<sub>E</sub>X + pT<sub>E</sub>X =  $\epsilon$ -pT<sub>E</sub>X の登場
- 内部コードを Unicode 化した upT<sub>E</sub>X,  $\epsilon$ -upT<sub>E</sub>X の登場



## 2 pL<sup>A</sup>T<sub>E</sub>X のこれまで

---

1980s pL<sup>A</sup>T<sub>E</sub>X 2.09 (ASCII Corporation)

**1995.11 pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> (ASCII Corporation)**

2006.11 アスキーによる現時点で最後の版

pT<sub>E</sub>X は日本ローカルのまま, UTF-8 入力不可, ε-T<sub>E</sub>X 拡張なし

- teT<sub>E</sub>X ベースの ptetex や後継の ptexlive による配布
- 入力文字コードとして UTF-8 に対応
- ε-T<sub>E</sub>X + pT<sub>E</sub>X = ε-pT<sub>E</sub>X の登場
- 内部コードを Unicode 化した upT<sub>E</sub>X, ε-upT<sub>E</sub>X の登場

**2010.05 pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> が pT<sub>E</sub>X とともに T<sub>E</sub>X Live へ収録**

2011.01 ε-pT<sub>E</sub>X が T<sub>E</sub>X Live に収録される

pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> の既定エンジンが ε-pT<sub>E</sub>X に

2011.08 upL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> が upT<sub>E</sub>X/ε-upT<sub>E</sub>X とともに T<sub>E</sub>X Live へ収録

→ pT<sub>E</sub>X が T<sub>E</sub>X Live と一体で開発可能 (pL<sup>A</sup>T<sub>E</sub>X だけ取り残される)

## 2 pL<sup>A</sup>T<sub>E</sub>X のこれまで

---

1980s      pL<sup>A</sup>T<sub>E</sub>X 2.09 (ASCII Corporation)  
**1995.11    pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> (ASCII Corporation)**  
2006.11    アスキーによる現時点で最後の版

**2016.01.06** (devel mailing list)

日本語 T<sub>E</sub>X 開発コミュニティが pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> を fork し, メンテナンス開始 = コミュニティ版 pL<sup>A</sup>T<sub>E</sub>X

日本語 T<sub>E</sub>X 開発コミュニティ (Japanese T<sub>E</sub>X Development Community)  
… 旧 pT<sub>E</sub>X-ml の後継としての開発者向けメーリングリスト  
日本語まわりの T<sub>E</sub>X 環境の整備 (pT<sub>E</sub>X 系にかぎらず開発全般)

- <https://texjp.org>
- <https://github.com/texjporg>

## 2 pL<sup>A</sup>T<sub>E</sub>X のこれまで

---

1980s      pL<sup>A</sup>T<sub>E</sub>X 2.09 (ASCII Corporation)  
**1995.11    pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> (ASCII Corporation)**  
2006.11    アスキーによる現時点で最後の版

**2016.01.06** (devel mailing list)

日本語 T<sub>E</sub>X 開発コミュニティが pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> を fork し, メンテナンス開始 = コミュニティ版 pL<sup>A</sup>T<sub>E</sub>X

2016.06    T<sub>E</sub>X Forum にて**コミュニティ版 pL<sup>A</sup>T<sub>E</sub>X 発表**<sup>†5</sup>

- T<sub>E</sub>X Live 2016 以降 (含 W32T<sub>E</sub>X)<sup>†6</sup> に反映
- upL<sup>A</sup>T<sub>E</sub>X もコミュニティ版 pL<sup>A</sup>T<sub>E</sub>X ベースに
- ライセンスは BSD 3-Clause (アスキー版 pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> と同じ)

---

<sup>†5</sup> pL<sup>A</sup>T<sub>E</sub>X と upL<sup>A</sup>T<sub>E</sub>X のコミュニティ版 (2016/05/07) (forum:1934)

<sup>†6</sup> より正確には W32T<sub>E</sub>X では 5 月 2 日, T<sub>E</sub>X Live 2016 pretest では 5 月 10 日以降.

### 3 コミュニティ版 pL<sup>A</sup>T<sub>E</sub>X の目的

---

最優先課題：

2013 年の pT<sub>E</sub>X の仕様変更による副作用への対処  
(脚注の合印前後に入る不自然な `\xkanjiskip` アキ問題)

### 3 コミュニティ版 pL<sup>A</sup>T<sub>E</sub>X の目的

---

最優先課題：

2013 年の pT<sub>E</sub>X の仕様変更による副作用への対処  
(脚注の合印前後に入る不自然な `\xkanjiskip` アキ問題)

もっと広げると

**pT<sub>E</sub>X の仕様変更・L<sup>A</sup>T<sub>E</sub>X の仕様変更で  
不都合が生じないようにメンテナンス**

### 3 コミュニティ版 pL<sup>A</sup>T<sub>E</sub>X の目的

---

最優先課題：

2013 年の pT<sub>E</sub>X の仕様変更による副作用への対処  
(脚注の合印前後に入る不自然な `\xkanjiskip` アキ問題)

もっと広げると

## pT<sub>E</sub>X の仕様変更・L<sup>A</sup>T<sub>E</sub>X の仕様変更で 不都合が生じないようにメンテナンス

- エンジンである pT<sub>E</sub>X や  $\epsilon$ -pT<sub>E</sub>X は, T<sub>E</sub>X Live のリポジトリで修正・変更されている.
- L<sup>A</sup>T<sub>E</sub>X も 2015 年以降, どんどん変わっている.
- この流れから pL<sup>A</sup>T<sub>E</sub>X だけ取り残されるのは嬉しくない.

## 4 組版結果にかかわる pL<sup>A</sup>T<sub>E</sub>X の修正・仕様変更

---

- 脚注番号（合印）前後のアキ：pT<sub>E</sub>X への追随
- 表組前後のアキ：pT<sub>E</sub>X への追随
- 下線前後のアキ：仕様変更
- 脚注番号（合印）前後のベタ組：仕様変更
- 合印直後の改行を許容：仕様変更
- 8-bit font encoding の欧文文字前後の四分アキ：仕様変更

## 4.1 脚注番号（合印）前後のアキ：pTEX への追隨

---

[2012/10/01]

Unicode に収録されている膨大な文字のなかでも、ゆきだるまは興味深い歴史を持っている。というのも、Unicode 5.1 以前では U+2603 SNOWMAN の一点のみ<sup>1</sup>の収録であったのに対し、Unicode 5.2 において新たに…

ここに注目！



---

<sup>1</sup>字形の差異はすべて包摂されていたことになる。



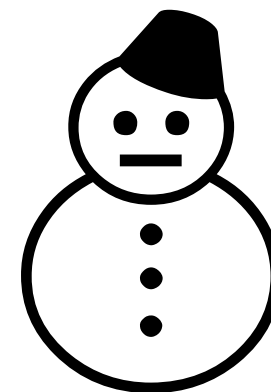
## 4.1 脚注番号（合印）前後のアキ：pTEX への追隨

---

[2013/10/01]

Unicode に収録されている膨大な文字のなかでも、ゆきだるまは興味深い歴史を持っている。というのも、Unicode 5.1 以前では U+2603 SNOWMAN の一点のみ<sup>1</sup>の収録であったのに対し、Unicode 5.2 において新たに…

不自然なアキ



---

<sup>1</sup>字形の差異はすべて包摂されていたことになる。

2013 年の pTEX の仕様変更<sup>†7</sup>に起因する \xkanjiskip アキ

---

<sup>†7</sup> 数式の前後の四分アキ (forum:913)

## 4.1 脚注番号（合印）前後のアキ：pTEX への追隨

---

[2016/09/08]

Unicode に収録されている膨大な文字のなかでも、ゆきだるまは興味深い歴史を持っている。というのも、Unicode 5.1 以前では U+2603 SNOWMAN の一点のみ<sup>1</sup>の収録であったのに対し、Unicode 5.2 において新たに…

元に戻った！



---

<sup>1</sup>字形の差異はすべて包摂されていたことになる。

pLATEX カーネルで \@makefnmark を再定義することにより，従来の挙動を取り戻した

## 4.2 表組前後のアキ：pT<sub>E</sub>X への追隨

[2012/10/01]

ここに、いくつかのフォントのゆきだるまの

	マフラー	地面
ヒラギノ	なし	なし
小塚	あり	なし
メイリオ	あり	あり

特徴を

とめてみると、一目瞭然である。

表にま

ここに注目！



## 4.2 表組前後のアキ：pT<sub>E</sub>X への追随

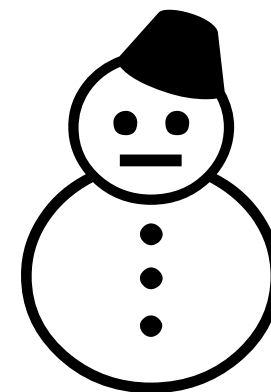
[2013/10/01]

ここに、いくつかのフォントのゆきだるまの

特徴を		マフラー	地面	表に
	ヒラギノ	なし	なし	
	小塚	あり	なし	
	メイリオ	あり	あり	

まとめてみると、一目瞭然である。

アキが入る



これもやはり、2013 年の pT<sub>E</sub>X の仕様変更起因する

## 4.2 表組前後のアキ：pT<sub>E</sub>X への追隨


[2016/09/08]

ここに、いくつかのフォントのゆきだるまの

		マフラー	地面	
特徴を	ヒラギノ	なし	なし	表にま
	小塚	あり	なし	
	メイリオ	あり	あり	

とめてみると、一目瞭然である。

元に戻った！



→ pL<sub>A</sub>T<sub>E</sub>X で tabular 環境を再定義し，従来の挙動へ復帰

## 4.2 表組前後のアキ：pT<sub>E</sub>X への追隨

[2013/10/01]

ここに、いくつかのフォントのゆきだるまの

		マアアアア	地面	2012 年
特徴を	ヒラギノ	なし	なし	表表は
	小塚	あり	なし	
	メイリオ	あり	あり	2013 年

まとめみると、一目瞭然である。

pT<sub>E</sub>X が変われば（pL<sub>A</sub>T<sub>E</sub>X は変わらなくても）出力は変わってしまう

## 4.2 表組前後のアキ：pT<sub>E</sub>X への追隨

[2013/10/01]

ここに、いくつかのフォントのゆきだるまの

		マアアアア	地面	2012 年
特徴を	ヒラギノ	なし	なし	表表は
	小塚	あり	なし	
	メイリオ	あり	あり	2013 年

まとめみると、一目瞭然である。

pT<sub>E</sub>X が変われば (pL<sub>A</sub>T<sub>E</sub>X は変わらなくても) 出力は変わってしまう  
→ pT<sub>E</sub>X を変える場合は, pL<sub>A</sub>T<sub>E</sub>X も必要に応じて変えなければならない

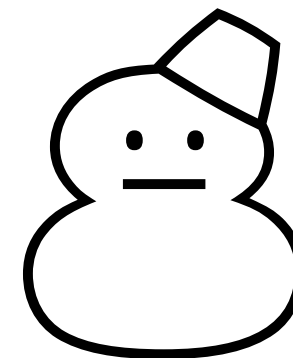
## 4.3 下線前後のアキ：pL<sup>A</sup>T<sub>E</sub>X の仕様変更

---

[2013/10/01]

このように ゆきだるま の特徴（降雪の有無、表情、マフラーの有無、地面の有無など）に注目すると、いわば「ゆきだるまの 親子関係」のようなものが見えてくるという先行研究がある。Unicode 1.0 で U+2603 SNOWMAN の例示字形にも採用…

下線の前後に  
アキがある



たびたび言及あり

奥村晴彦・黒木裕介 著「改訂第 6 版 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 美文書入門」(2013) p.291,  
吉永徹美 著「独習 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>」(2008) p.110 など



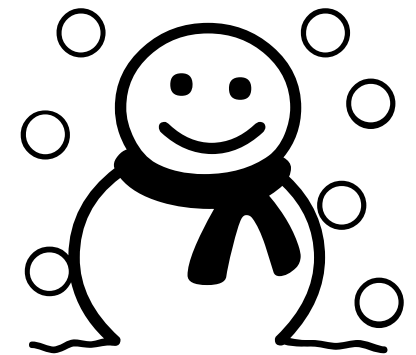
## 4.3 下線前後のアキ：pL<sup>A</sup>T<sub>E</sub>X の仕様変更

---

[2016/09/08]

このようにゆきだるまの特徴（降雪の有無、表情、マフラーの有無、地面の有無など）に注目すると、いわば「ゆきだるまの親子関係」のようなものが見えてくるという先行研究がある。Unicode 1.0 で U+2603 SNOWMAN の例示字形にも採用…

**アキを削除！**



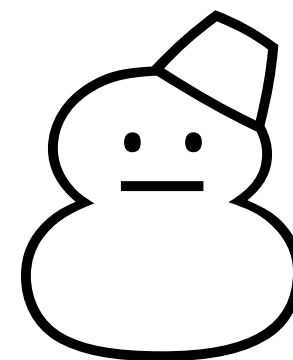
→ pL<sup>A</sup>T<sub>E</sub>X カーネルで `\underline` を再定義し、アキを削除

## 4.4 脚注番号（合印） 前後のベタ組：pL<sup>A</sup>T<sub>E</sub>X の仕様変更

[2012/10/01]

ヒラギノのゆきだるまはマフラーを巻いていないことは周知の事実である<sup>1</sup>。一方で、メイリオや小塚のゆきだるまはマフラーを巻いていて、雪が降っている。<sup>2</sup>本論文では、…

区切り約物の  
後ろにアキ



---

<sup>1</sup> しかも、なんとなく悲しそうな表情をしている。

<sup>2</sup> 笑顔がかわいい。

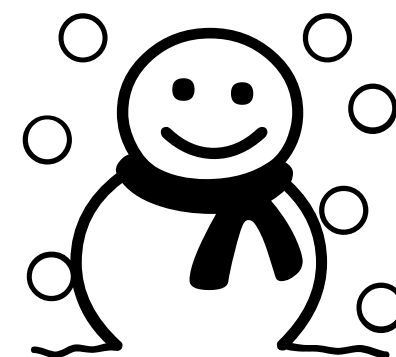
jsclasses ではかなり昔から修正済み

## 4.4 脚注番号（合印） 前後のベタ組：pL<sup>A</sup>T<sub>E</sub>X の仕様変更

[2016/09/08]

ヒラギノのゆきだるまはマフラーを巻いていないことは周知の事実である<sup>1</sup>。一方で、メイリオや小塚のゆきだるまはマフラーを巻いていて、雪が降っている。<sup>2</sup>本論文では、…

**ベタ組に！**



---

<sup>1</sup> しかも、なんとなく悲しそうな表情をしている。

<sup>2</sup> 笑顔がかわいい。

→ pL<sup>A</sup>T<sub>E</sub>X カーネルに jsclasses と同様のコードを導入  
（どちらの流儀が “一般的” かは問わない）

## 4.5 合印直後の改行を許容：pL<sup>A</sup>T<sub>E</sub>X の仕様変更

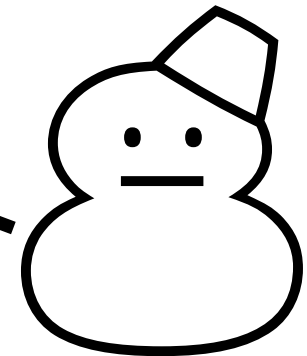
[2012/10/01]

メイリオのゆきだるまと小塚のゆきだるま<sup>1</sup>を比較すると、メイリオのゆきだるまには地面が描かれているという特徴に気づく。地面が描かれているゆきだるまは、我々の調査ではメイリオ以外には見受けられず、…

---

<sup>1</sup>いずれも雪が降っていて、マフラーを巻いている点で共通している。

「1」と「を」の間は分割禁止  
でないはず…



行分割してよいはずの場所で改行が起きない  
→ スカスカの行が出来たり，Overfull 警告が出たり…

## 4.5 合印直後の改行を許容：pL<sup>A</sup>T<sub>E</sub>X の仕様変更

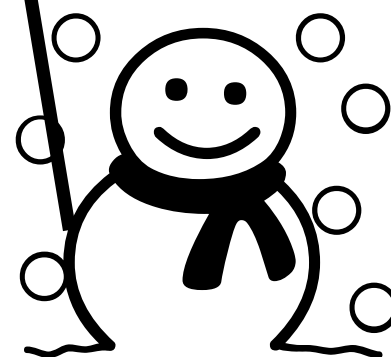
[2016/09/08]

メイリオのゆきだるまと小塚のゆきだるま<sup>1</sup>を比較すると、メイリオのゆきだるまには地面が描かれているという特徴に気づく。地面が描かれているゆきだるまは、我々の調査ではメイリオ以外には見受けられず、…

---

<sup>1</sup>いずれも雪が降っていて、マフラーを巻いている点で共通している。

**分割可能に！**



→ pL<sup>A</sup>T<sub>E</sub>X カーネルを修正し，自然な行組版を可能に

## 4.6 8-bit font encoding の欧文文字前後の四分アキ

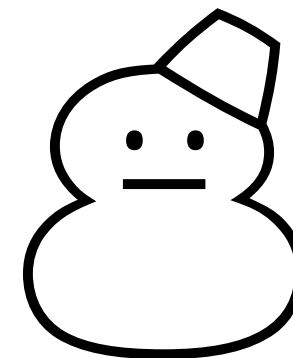
[2012/10/01]

昨日、ドイツのÖhringen にある Caféに入りました<sup>1</sup>。早速メニューを持ってきてもらおうとすると、なんと定食がでてきました。そう、ドイツ語で Menüとはメニューでは…

---

<sup>1</sup>ちなみに「カフェ」は、英語やフランス語ではドイツ語と同様に caféと書きますが、イタリア語では caffèと書きます。

**アクセント文字  
の前後に和欧文  
間空白がない！**



当時は T1 エンコーディングなどが一般的ではなく、未考慮だったらしい (jsclasses や upL<sup>A</sup>T<sub>E</sub>X では対処済み)

## 4.6 8-bit font encoding の欧文文字前後の四分アキ

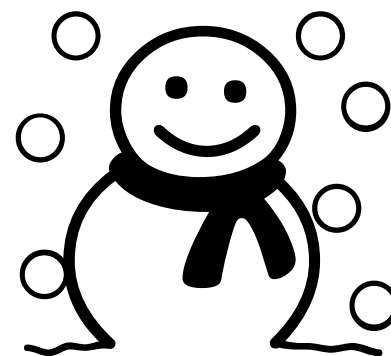
[2016/09/08]

昨日、ドイツの Öhringen にある Café に入りました<sup>1</sup>。早速メニューを持ってきてもらおうとすると、なんと定食が出てきました。そう、ドイツ語で Menü とはメニューでは…

**四分アキあり**

---

<sup>1</sup>ちなみに「カフェ」は、英語やフランス語ではドイツ語と同様に café と書きますが、イタリア語では caffè と書きます。



→ pL<sup>A</sup>T<sub>E</sub>X でも文字コード 128–255 に \xspcode=3 を設定

## 5 L<sup>A</sup>T<sub>E</sub>X への追随

---

今のところ大事なものはひとつだけ：

- 入れ子の強調書体 `\eminnershape` の採用 ← 要するに `{\em{\em この中}}`



## 5 L<sup>A</sup>T<sub>E</sub>X への追随

今のところ大事なのはひとつだけ：

- 入れ子の強調書体 `\eminnershape` の採用 ← 要するに `{\em{\em この中}}`

- デフォルトは従来どおり：

```
\def\eminnershape{\mcfamily \upshape}
```

- カスタマイズ例：

```
\renewcommand{\eminnershape}{\gtfamily \scshape}
```

	地の文	強調	入れ子の強調
L <sup>A</sup> T <sub>E</sub> X のデフォルト	Roman	<i>italic</i>	<i>ROMAN</i>
pL <sup>A</sup> T <sub>E</sub> X のデフォルト	明朝体	ゴシック	明朝体
	Roman	<i>italic</i>	Roman
カスタマイズ	明朝体	ゴシック	明朝体
	Roman	<i>italic</i>	<i>ROMAN</i>

## 6 警告やエラーが出る現象への対処

---

- 縦組時の Overfull 警告の抑制
- 縦組で longtable パッケージが無限ループする現象への対処
  - 以上 2 件は \@makecol の修正
- 縦組で \AtBeginDocument{\AtBeginDvi}}のエラーを解消
  - \@begindvibox を常に横組ボックスへ

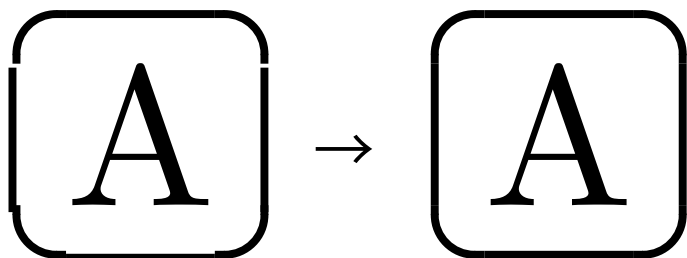
## 6 警告やエラーが出る現象への対処

---

- 縦組時の Overfull 警告の抑制
- 縦組で longtable パッケージが無限ループする現象への対処
  - 以上 2 件は \@makecol の修正
- 縦組で \AtBeginDocument{\AtBeginDvi}}のエラーを解消
  - \@begindvibox を常に横組ボックスへ

ascmac パッケージも修正

- pict2e パッケージと併用するとエラーが出る現象への対処
  - 従来は! Missing number, treated as zero. などのエラー
- pdfL<sup>A</sup>T<sub>E</sub>X などでも利用可能なように拡張
  - 従来は \tbaselineshift が未定義というエラー
- \keytop や screen/itembox の角のズレ修正



(印刷すると綺麗なはず)

## 7 カーネルの互換性…過去の挙動のエミュレート

新しい pL<sup>A</sup>T<sub>E</sub>X には組版上の改良が多数加わりました

→ でも, 前に書いたソースは “同じ見た目” になってほしい (場合もある)

## 7 カーネルの互換性…過去の挙動のエミュレート

---

新しい pL<sup>A</sup>T<sub>E</sub>X には組版上の改良が多数加わりました

→ でも, 前に書いたソースは “同じ見た目” になってほしい (場合もある)

そんなときは

### platexrelease パッケージ

- 本家 L<sup>A</sup>T<sub>E</sub>X に添付の latexrelease パッケージの pL<sup>A</sup>T<sub>E</sub>X 版
- 日付を指定して読み込むと, その日付の pL<sup>A</sup>T<sub>E</sub>X カーネルを再現

```
\RequirePackage[2016/03/31]{platexrelease}
```

```
\documentclass{jsarticle}
```

→ 2016/03/31 時点の L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> と pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> の組み合わせ

多くの場合には, これで過去の挙動をエミュレートすれば “同じ見た目” になるはず.

## 7 カーネルの互換性…過去の挙動のエミュレート

新しい pL<sup>A</sup>T<sub>E</sub>X には組版上の改良が多数加わりました

→ でも, 前に書いたソースは “同じ見た目” になってほしい (場合もある)

そんなときは

### platexrelease パッケージ

- 本家 L<sup>A</sup>T<sub>E</sub>X に添付の latexrelease パッケージの pL<sup>A</sup>T<sub>E</sub>X 版
- 日付を指定して読み込むと, その日付の pL<sup>A</sup>T<sub>E</sub>X カーネルを再現

```
\RequirePackage[2016/03/31]{platexrelease}
```

```
\documentclass{jsarticle}
```

→ 2016/03/31 時点の L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> と pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> の組み合わせ

多くの場合には, これで過去の挙動をエミュレートすれば “同じ見た目” になるはず. **ただし:**

**原理的に完全ではない (pT<sub>E</sub>X の挙動が変わった場合は, 再現不能)**

## 8 pL<sup>A</sup>T<sub>E</sub>X の今後

---

最大の課題

**どのような変更を許容するか**

## 8 pL<sup>A</sup>T<sub>E</sub>X の今後

---

### 最大の課題

### どのような変更を許容するか

pT<sub>E</sub>X の挙動は案外コロコロ変わっている（特に四分アキと和欧文ベースライン補正で顕著）

- 数式も欧文とみなしてベースラインシフト (2009)
- 数式周囲の `\xkanjiskip` 挿入漏れを解消 (2013)
- 数式内の明示的な `\hbox` ではベースラインシフトを戻す (2016)
- 数式周囲の `\xkanjiskip` 挿入漏れをさらに解消 (2017)
- 縦ディレクションと縦数式ディレクションの区別 (2017)



# 8 pL<sup>A</sup>T<sub>E</sub>X の今後

---

## 最大の課題

### どのような変更を許容するか

pT<sub>E</sub>X の挙動は案外コロコロ変わっている（特に四分アキと和欧文ベースライン補正で顕著）

- 数式も欧文とみなしてベースラインシフト (2009)
- 数式周囲の `\xkanjiskip` 挿入漏れを解消 (2013)
- 数式内の明示的な `\hbox` ではベースラインシフトを戻す (2016)
- 数式周囲の `\xkanjiskip` 挿入漏れをさらに解消 (2017)
- 縦ディレクションと縦数式ディレクションの区別 (2017)

いずれもアスキーによる最後の pL<sup>A</sup>T<sub>E</sub>X 2006/11/10 より**後**の変更

→ これによって **pL<sup>A</sup>T<sub>E</sub>X の挙動が変わっている（いた）！**

この事態に対する指摘が 6 年間で極端に少なかったのはなぜ？

## 8.1 pL<sup>A</sup>T<sub>E</sub>X の変更をどこまで許容するか

---

pT<sub>E</sub>X/ε-pT<sub>E</sub>X の変化に由来する pL<sup>A</sup>T<sub>E</sub>X の変化に「気づいていない」？  
組版結果の変化に対して寛容？

## 8.1 pL<sup>A</sup>T<sub>E</sub>X の変更をどこまで許容するか

---

pT<sub>E</sub>X/ε-pT<sub>E</sub>X の変化に由来する pL<sup>A</sup>T<sub>E</sub>X の変化に「気づいていない」？  
組版結果の変化に対して寛容？

一つの主張：

「pL<sup>A</sup>T<sub>E</sub>X カーネルは**安定**であるべき」

言うのは簡単，では具体的には？

## 8.1 pL<sup>A</sup>T<sub>E</sub>X の変更をどこまで許容するか

---

pT<sub>E</sub>X/ε-pT<sub>E</sub>X の変化に由来する pL<sup>A</sup>T<sub>E</sub>X の変化に「気づいていない」？  
組版結果の変化に対して寛容？

一つの主張：

「pL<sup>A</sup>T<sub>E</sub>X カーネルは**安定**であるべき」

言うのは簡単，では具体的には？

- 出力が変わらないこと？
- コードに触らないこと？

コードに触ればバグもつきもの.

でも触らないで不都合な挙動を残すのも親切とは思えない.

## 8.1 pL<sup>A</sup>T<sub>E</sub>X の変更をどこまで許容するか

---

pT<sub>E</sub>X/ε-pT<sub>E</sub>X の変化に由来する pL<sup>A</sup>T<sub>E</sub>X の変化に「気づいていない」？  
組版結果の変化に対して寛容？

一つの主張：

「pL<sup>A</sup>T<sub>E</sub>X カーネルは**安定**であるべき」

言うのは簡単，では具体的には？

- 出力が変わらないこと？
- コードに触らないこと？

コードに触ればバグもつきもの.

でも触らないで不都合な挙動を残すのも親切とは思えない.

私の立場：

- バグが入ることも覚悟で**コードを変更** ← L<sup>A</sup>T<sub>E</sub>X もそうやっているし.
- バグや不都合の報告が来たら**速攻で直す** or 戻す！

## 8.2 「変える」ことによる安定性

---

大事なことなのでもう一度，私の立場：

- バグが入ることも覚悟で**コードを変更**
- バグや不都合の報告が来たら**速攻で直す** or 戻す！

まずは認識のすりあわせ（立場の明言は大事です）

- 「pL<sup>A</sup>T<sub>E</sub>X は極力いじらない」ではなく  
「その時点の pT<sub>E</sub>X ( $\varepsilon$ -pT<sub>E</sub>X) と L<sup>A</sup>T<sub>E</sub>X の上に構築された状態」を考える
  - pT<sub>E</sub>X（または  $\varepsilon$ -pT<sub>E</sub>X）も L<sup>A</sup>T<sub>E</sub>X も変わっているのに，その上で動作する **pL<sup>A</sup>T<sub>E</sub>X の出力が変わらないのを期待するのは酷**だよね.
  - pT<sub>E</sub>X（または  $\varepsilon$ -pT<sub>E</sub>X）を改良するのであれば，pL<sup>A</sup>T<sub>E</sub>X も改良していくのが自然.（ダブルスタンダードの防止）
- では，pL<sup>A</sup>T<sub>E</sub>X 標準クラス（jarticle, tarticle など）は？

## 8.2 「変える」ことによる安定性

---

大事なことなのでもう一度、私の立場：

- バグが入ることも覚悟で**コードを変更**
- バグや不都合の報告が来たら**速攻で直す** or 戻す！

もちろん、バグが入らないように努力する（→ 後述）  
でも完璧ではない。大事なものは：

バグを入れないことよりも、**入りうるバグの影響の最小化**

## 8.2 「変える」ことによる安定性

---

大事なことなのでもう一度，私の立場：

- バグが入ることも覚悟で**コードを変更**
- バグや不都合の報告が来たら**速攻で直す** or 戻す！

もちろん，バグが入らないように努力する（→ 後述）  
でも完璧ではない．大事なものは：

バグを入れないことよりも，**入りうるバグの影響の最小化**

たとえば：

- T<sub>E</sub>X Live の年度終盤には大きな変更を入れない
  - frozen になってからでは最大 2 ヶ月にわたり不都合が続く．
- 入ってしまったバグへの対症療法をあらかじめ準備
  - これは platexrelease パッケージの仕事．



## 8.3 バグを入れない努力：実験的 pL<sup>A</sup>T<sub>E</sub>X

---

みなさんも「開発版 pL<sup>A</sup>T<sub>E</sub>X」をすぐにテストできます！

## 8.3 バグを入れない努力：実験的 pL<sup>A</sup>T<sub>E</sub>X

---

みなさんも「開発版 pL<sup>A</sup>T<sub>E</sub>X」をすぐにテストできます！

**exppl2e パッケージ**<sup>†8</sup> (EXPerimental PLatex2E)

---

<sup>†8</sup> expl3 パッケージや fixltx2e パッケージをもじった命名

## 8.3 バグを入れない努力：実験的 pL<sup>A</sup>T<sub>E</sub>X

---

みなさんも「開発版 pL<sup>A</sup>T<sub>E</sub>X」をすぐにテストできます！

**exppl2e パッケージ** (EXPerimental PLatex2E)

“おまじない” は `\RequirePackage{exppl2e}`

- 任意のファイルの冒頭, あるいは
- T<sub>E</sub>X から見つかる任意の場所に用意した `platex.cfg` ファイル

に書くと, テスト版 pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> (と実質的に同じもの) が動作する.

```
\RequirePackage{exppl2e}
\documentclass{jsarticle}
...
```

これで, 「フォーマット作成って何？」な人でも安心.

(もちろん, GitHub – <https://github.com/texjporg/platex> を見ていただけるともっと嬉しい.)

## 8.4 具体的なテスト内容（本日，2016 年 11 月 5 日現在）

---

- “アクセント文字パッチ”
- “`\strutbox` パッチ”
- $\varepsilon$ -pT<sub>E</sub>X の “FAM256 パッチ” の活用（GitHub のみで配布中）

組版に大きくかわるのは最初の 2 つ.

ちなみに，これらは現行の LuaT<sub>E</sub>X-ja でも同様らしい.

※ FAM256 パッチを活用すれば，数式ファミリを従来の 16 個から 256 個まで増やせる.

→ `\DeclareMathAlphabet` がたくさんあっても大丈夫

## 8.4 具体的なテスト内容（本日, 2016 年 11 月 5 日現在）

---

- “アクセント文字パッチ” の例：`\AA` の出力

```
\documentclass{tarticle}
```

```
\begin{document}
```

$1\,\mathrm{nm}$  の  $1/10$  を表す長さの単位は、  
オングストローム (`\AA`) と書きます。

```
\end{document}
```

## 8.4 具体的なテスト内容（本日，2016 年 11 月 5 日現在）

- “アクセント文字パッチ” の例：\AA の出力

現状

1mm の 1/10 を表す長さの単位は、  
オングストローム (Å) と書きます。

OT1 encoding の  
アクセント合成文  
字が変になる



修正案

1mm の 1/10 を表す長さの単位は、  
オングストローム (Å) と書きます。

一度はカーネルに  
修正を加えたが、  
副作用のため現在  
テスト段階



## 8.4 具体的なテスト内容（本日, 2016 年 11 月 5 日現在）

---

- “\strutbox パッチ” の例：amsmath の alignment (&) と数式番号

```
\documentclass{tarticle}
```

```
\usepackage{amsmath}
```

```
\begin{document}
```

align 環境 \texttt{\&} が 1 つ

```
\begin{align}
```

```
  a_1 &= b_1+c_1\\
```

```
  a_2 &= b_2+c_2-d_2
```

```
\end{align}
```

align 環境 \texttt{\&} なし

```
\begin{align}      a_1=b_1+c_1    \end{align}
```

比較用の equation 環境

```
\begin{equation}  a_1=b_1+c_1    \end{equation}
```

```
\end{document}
```

## 8.4 具体的なテスト内容（本日，2016 年 11 月 5 日現在）

- “\strutbox パッチ” の例：amsmath の alignment (&) と数式番号

現状

align 環境が 1 つ

$$a_1 = b_1 + c_1$$
$$a_2 = b_2 + c_2 - d_2$$

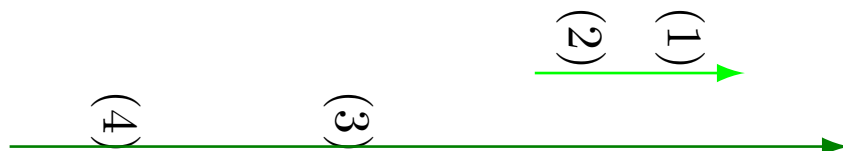
align 環境なし

$$a_1 = b_1 + c_1$$

比較用の equation 環境

$$a_1 = b_1 + c_1$$

数式番号が揃わない



修正案

align 環境が 1 つ

$$a_1 = b_1 + c_1$$
$$a_2 = b_2 + c_2 - d_2$$

align 環境なし

$$a_1 = b_1 + c_1$$

比較用の equation 環境

$$a_1 = b_1 + c_1$$

副作用がないか調査中





## 9 まとめ

---

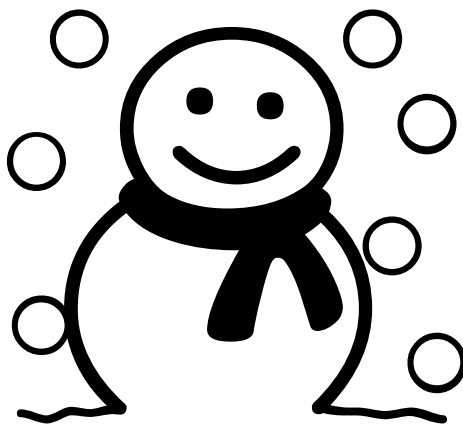
- pL<sup>A</sup>T<sub>E</sub>X は 2016 年以降, コミュニティ版になった
- 組版の改善, エラーや警告の解消など
- 互換性を platexrelease パッケージで極力サポート
- テスト版 pL<sup>A</sup>T<sub>E</sub>X = exppl2e パッケージへのご協力を
- これからは, pT<sub>E</sub>X/ $\epsilon$ -pT<sub>E</sub>X と L<sup>A</sup>T<sub>E</sub>X に追随した pL<sup>A</sup>T<sub>E</sub>X の開発を
  - 2017 年頭には「L<sup>A</sup>T<sub>E</sub>X は  $\epsilon$ -T<sub>E</sub>X 拡張を必須とする」そうです

## 9 まとめ

---

- pL<sup>A</sup>T<sub>E</sub>X は 2016 年以降, コミュニティ版になった
- 組版の改善, エラーや警告の解消など
- 互換性を platexrelease パッケージで極力サポート
- テスト版 pL<sup>A</sup>T<sub>E</sub>X = exppl2e パッケージへのご協力を
- これからは, pT<sub>E</sub>X/ $\varepsilon$ -pT<sub>E</sub>X と L<sup>A</sup>T<sub>E</sub>X に追随した pL<sup>A</sup>T<sub>E</sub>X の開発を
  - 2017 年頭には「L<sup>A</sup>T<sub>E</sub>X は  $\varepsilon$ -T<sub>E</sub>X 拡張を必須とする」そうです

御清聴ありがとうございました



Happy pL<sup>A</sup>T<sub>E</sub>Xing!

ソースコードあります

<https://github.com/aminophen/hytexconf16>

ブログも見てね

<http://acetaminophen.hatenablog.com>