

# p<sub>2</sub>TEX のペナルティ

山下 弘展

2019 年 10 月 12 日  
TEXConf 2019

# はじめに：ペナルティとは

## $\text{T}_{\text{E}}\text{X}$ におけるペナルティ (penalty)

行分割時やページ分割時に「その箇所がブレイクポイントとしてどの程度適切であるか」を示す評価値.

$\text{pT}_{\text{E}}\text{X}$  では、和文組版に必須の禁則処理を実現するために拡張されている.

※  $\text{pT}_{\text{E}}\text{X}$  にはもう一つ、パラグラフの最終行が和文文字一字で孤立するのを防ぐ `\jcharwidowpenalty` というペナルティもあるが、本講演では時間の都合上取り扱わない.

# 前提

- ▶ pTeX p3.8.2 (TeX Live 2019) ベース
- ▶ 特に断らない限り，初期設定を仮定

```
% 禁則ペナルティ
\prebreakpenalty' .=10000
\prebreakpenalty'。 =10000
\postbreakpenalty' (=10000
\postbreakpenalty' (=10000
% 一文字だけの行を抑制
\jcharwidowpenalty=500
```

# 禁則処理とペナルティ

## 組版規則

- ▶ 欧文：連続する文字列＝単語の途中では行分割が起こらない<sup>+1</sup>
- ▶ 和文：ほとんどの文字間が分割可能。この例外が**禁則**（行頭／行末禁則）。

---

<sup>+1</sup>ハイフネーション処理等の特別な場合を除く。

# 禁則処理とペナルティ

## 組版規則

- ▶ 欧文：連続する文字列＝単語の途中では行分割が起こらない<sup>+1</sup>
- ▶ 和文：ほとんどの文字間が分割可能。この例外が**禁則**（行頭／行末禁則）。

pT<sub>E</sub>X の実装では…

- ▶ 各文字間を分割可能に＝グルーの自動挿入  
`\kanjiskip, \xkanjiskip`
- ▶ 禁則処理＝ペナルティの自動挿入  
禁則テーブルに基づく

<sup>+1</sup>ハイフネーション処理等の特別な場合を除く。

# 参考：NTT $\text{jT}_E\text{X}$ との比較

$\text{pT}_E\text{X}$  の実装では…

- ▶ 各文字間を分割可能に = グルー の自動挿入  
`\kanjiskip, \xkanjiskip`
- ▶ 禁則処理 = ペナルティ の自動挿入  
禁則テーブルに基づく

$\text{jT}_E\text{X}$  の実装では…

- ▶ 各文字間を分割可能に = グルー の自動挿入  
`\jintercharskip, \jasciikanjiskip`
- ▶ 禁則処理 = グルーの自動挿入を抑制<sup>+2</sup>

---

<sup>+2</sup>記号類とひらがな、カタカナについてのみ指定可。

# pT<sub>E</sub>X の禁則処理の基礎

pT<sub>E</sub>X は**禁則テーブル**に従い、**ペナルティ**を文字の前後適切な位置に自動挿入する。

## 禁則テーブル (KINSOKU table)

- ▶ 禁則対象とする文字のコード
- ▶ その文字に対するペナルティ値
- ▶ ペナルティの挿入位置（行頭禁則ならその文字の前に挿入，行末禁則なら後）

を登録できる（最大 256 文字まで）。

# pT<sub>E</sub>X の禁則処理の基礎

pT<sub>E</sub>X は禁則テーブルに従い、ペナルティを文字の前後適切な位置に自動挿入する。

## 禁則テーブル (KINSOKU table)

- ▶ 禁則対象とする文字のコード
- ▶ その文字に対するペナルティ値
- ▶ ペナルティの挿入位置（行頭禁則ならその文字の前に挿入，行末禁則なら後）

を登録できる（最大 256 文字まで）。

テーブルへの登録・削除用プリミティブ：

`\prebreakpenalty`, `\postbreakpenalty`



# pTeX の禁則処理の基礎

pTeX の標準設定では

- ▶ 約物（句読点やカッコ類）
- ▶ 小書き仮名

等が禁則テーブルに登録されている。

例：

```
\prebreakpenalty'。=10000    % "。" の前に 10000
\postbreakpenalty' (=10000    % " (" の後に 10000
\prebreakpenalty'や=150      % "や" の前に 150
```

しかし、ユーザが自由に登録・削除・値を変更することもできるようになっている<sup>+3</sup>。

---

<sup>+3</sup>実際に削除できるようになったのは、コミュニティ版 p3.8.1 以降。

# pTeX の禁則処理の制約事項

## 1. 禁則の指定は文字コードに依る

禁則テーブルには和文文字と欧文文字の区別なく登録できるが、登録されるのは「文字」そのものではなく「**文字コード**」である。

→ pTeX では和文文字の内部コードは 2 バイト，欧文文字は 1 バイトであるから衝突しない。

upTeX では和文文字 (CJK) の内部コードが欧文文字 (8-bit Latin) に被ることがある：

- ▶ 0xA1: i (CJK) vs. a (8-bit Latin)
- ▶ 0xAB: « (CJK) vs. n (8-bit Latin)
- ▶ 0xB7: · (CJK) vs. u (8-bit Latin)

“一方を立てれば一方が立たず”

＝前者を禁則扱いすれば，後者も禁則扱いに

# pTeX の禁則処理の制約事項

## 2. 行頭禁則と行末禁則は同時指定不可

同一の文字コードに対して、`\prebreakpenalty`と`\postbreakpenalty`の両方を同時に与えるような指定はできない。もし両方指定された場合、後から指定されたものに置き換えられる。

例：

```
\prebreakpenalty'~ = 10000
\showthe\prebreakpenalty'~ % => 10000
\showthe\postbreakpenalty'~ % => 0
\postbreakpenalty'~ = 10000
\showthe\prebreakpenalty'~ % => 0
\showthe\postbreakpenalty'~ % => 10000
```

# 本題

ここまでがアスキーによる「仕様」.  
しかし、いくつか注意したい点がある.

# 1. 欧文文字に対する禁則処理の課題

禁則テーブルには和文文字と欧文文字の区別なく登録できるが、実際には細かい注意が必要.



# 1. 欧文文字に対する禁則処理の課題

禁則テーブルには和文文字と欧文文字の区別なく登録できるが、実際には細かい注意が必要.



pTEX がボックスを構築する様子を明らかにするため、`\showlists` や `\showbox` といったプリミティブを活用.

ボックスの中身を完全に表示するためのおまじない :

```
\tracingonline = 1
\showboxdepth = 10000
\showboxbreadth = 10000
```

# 1.1. 禁則ペナルティの挿入タイミング

## 和文文字の場合

禁則ペナルティは、その文字ノードをリストに追加する際に一緒に挿入される。

例：upTeX で和文フォント upjisr-h を選択した状況

```
▶ \font\x=upjisr-h \x  
   \setbox0=\vbox{です。 \showlists}
```

---

```
\x で  
\x す  
\penalty 10000(for kinsoku)  
\x 。  
\glue(refer from jfm) 5.0
```

---

→ 〈。〉の前に \prebreakpenalty 由来のペナルティ 10000 が入った

# 1.1. 禁則ペナルティの挿入タイミング

## 和文文字の場合

禁則ペナルティは、その文字ノードをリストに追加する際に一緒に挿入される。

例：upTeX で和文フォント upjisr-h を選択した状況

```
▶ \font\x=upjisr-h \x
   \setbox0=\vbox{まず (\showlists)}



---



\x ま
\x ず
\glue(refer from jfm) 5.0 minus 5.0
\x (
\penalty 10000(for kinsoku)
```

→ 〈 ( ) の後に \postbreakpenalty 由来のペナルティ 10000 が入った



# 1.1. 禁則ペナルティの挿入タイミング

## 和文文字の場合

禁則ペナルティは、その文字ノードをリストに追加する際に一緒に挿入される。

→ この仕様から導かれる挙動（定理）：

- ▶ 和文文字の `\postbreakpenalty` は「現在のリストの最後」になりうるため、`\lastpenalty` で値を取得できるし、`\unpenalty` で取り除くこともできる。
- ▶ 和文文字の `\prebreakpenalty` は原理的に必ず後ろに文字ノードを伴うため、挿入後にその値を取得したり取り除いたりすることはできない。

# 1.1. 禁則ペナルティの挿入タイミング

例：和文文字の `\postbreakpenalty` の取得

```
▶ \font\x=upjisr-h \x
   \setbox0=\vbox{まず (\showthe\lastpenalty}


---


   > 10000.
```

例：和文文字の `\postbreakpenalty` の削除

```
▶ \font\x=upjisr-h \x
   \setbox0=\vbox{まず (\unpenalty\showlists}


---


   \x ま
   \x ず
   \glue(refer from jfm) 5.0 minus 5.0
   \x (
```

# 1.1. 禁則ペナルティの挿入タイミング

## 欧文文字の場合

禁則ペナルティは、**禁則対象の欧文文字が和文文字と隣り合う場合にのみ**挿入される。つまり、欧文のみの組版では挿入されない（Knuth の T<sub>E</sub>X と互換）。

例：upT<sub>E</sub>X でさらに欧文フォント cmr10 を選択した状況

```
▶ \font\x=upjisr-h \x \font\A=cmr10 \A
   \setbox0=\vbox{だるま.\showlists}
```

---

```
\x だ
\x る
\x ま
\penalty 10000(for kinsoku)
\A .
```

---

→ 和文〈ま〉-欧文〈.〉間に、〈.〉由来の `\prebreakpenalty` が入った

# 1.1. 禁則ペナルティの挿入タイミング

## 欧文文字の場合

禁則ペナルティは、**禁則対象の欧文文字が和文文字と隣り合う場合にのみ**挿入される。つまり、欧文のみの組版では挿入されない（Knuth の  $\text{T}_{\text{E}}\text{X}$  と互換）。

例： $\text{upT}_{\text{E}}\text{X}$  でさらに欧文フォント `cmr10` を選択した状況

```
▶ \font\x=upjisir-h \x \font\A=cmr10 \A
   \setbox0=\vbox{A dog.\showlists}
```

---

```
\A A
\glue 3.33333 plus 1.66498 minus 1.11221
\A d
\A o
\A g
\A .
```

---

→ 欧文 `<g>`-欧文 `<.>` 間には `\prebreakpenalty` が入らない

# 1.1. 禁則ペナルティの挿入タイミング

## 欧文文字の場合

禁則ペナルティは、**禁則対象の欧文文字が和文文字と隣り合う場合にのみ**挿入される。つまり、欧文のみの組版では挿入されない（Knuth の T<sub>E</sub>X と互換）。

例：upT<sub>E</sub>X でさらに欧文フォント cmr10 を選択した状況

```
▶ \font\x=upjlsr-h \x \font\A=cmr10 \A  
\setbox0=\vbox{A dog.}\showbox0
```

---

```
..\A A  
..\glue 3.33333 plus 1.66498 minus 1.11221  
..\A d  
..\A o  
..\A g  
..\A .
```

---

→ ボックスを組み終わっても `\prebreakpenalty` は入らない

# 1.1. 禁則ペナルティの挿入タイミング

## 欧文文字の場合

禁則ペナルティは、**禁則対象の欧文文字が和文文字と隣り合う場合にのみ**挿入される。つまり、欧文のみの組版では挿入されない（Knuth の T<sub>E</sub>X と互換）。

例：upT<sub>E</sub>X でさらに欧文フォント cmr10 を選択した状況

```
▶ \font\x=upjisr-h \x \font\A=cmr10 \A  
   \setbox0=\vbox{(だるま \showlists}
```

---

```
\A (  
\penalty 10000(for kinsoku)  
\x だ  
\x る  
\x ま
```

---

→ 欧文 ⟨(⟩-和文 ⟨だ⟩ 間に、⟨(⟩ 由来の `\postbreakpenalty` が入った

# 1.1. 禁則ペナルティの挿入タイミング

## 欧文文字の場合

禁則ペナルティは、**禁則対象の欧文文字が和文文字と隣り合う場合にのみ**挿入される。つまり、欧文のみの組版では挿入されない（Knuth の  $\text{T}_\text{E}\text{X}$  と互換）。

例： $\text{upT}_\text{E}\text{X}$  でさらに欧文フォント `cmr10` を選択した状況

▶ `\font\x=upjistr-h \x \font\A=cmr10 \A  
\setbox0=\vbox{(dog\showlists)}`

---

```
\A (
\A d
\A o
\A g
```

---

→ 欧文 `<(>`-欧文 `<d>` 間には `\postbreakpenalty` が入らない

# 1.1. 禁則ペナルティの挿入タイミング

## 欧文文字の場合

禁則ペナルティは、**禁則対象の欧文文字が和文文字と隣り合う場合にのみ**挿入される。つまり、欧文のみの組版では挿入されない（Knuth の  $\text{T}_{\text{E}}\text{X}$  と互換）。

→ 隣接する文字が重要なので…

- ▶ 欧文文字の `\postbreakpenalty` は**後続の和文文字ノード追加時**に挿入される（欧文文字自身と同時ではない）。  
→ 「**リストの最後**」になりえず、`\lastpenalty` で取得できないし、`\unpenalty` でも消せない。
- ▶ 欧文文字の `\prebreakpenalty` は和文と同様。



# 1.1. 禁則ペナルティの挿入タイミング

例：欧文文字の `\postbreakpenalty` の削除（失敗）

- ▶ `\font\x=upjisr-h \x \font\A=cmr10 \A  
\setbox0=\vbox{(だるま \showlists}`

---

```
\A (  
\penalty 10000(for kinsoku)  
\x だ  
\x る  
\x ま
```

---

- ▶ `\font\x=upjisr-h \x \font\A=cmr10 \A  
\setbox0=\vbox{(\unpenalty だるま \showlists}`

---

```
\A (  
\penalty 10000(for kinsoku)  
\x だ  
\x る  
\x ま
```

---

# 1.1. 禁則ペナルティの挿入タイミング

例：欧文文字の `\postbreakpenalty` の削除（失敗）

▶ `\font\x=upjisir-h \x \font\A=cmr10 \A  
\setbox0=\vbox{(だるま \showlists}`

---

```
\A (  
\penalty 10000(for kinsoku)  
\x だ  
\x る  
\x ま
```

---

▶ `\font\x=upjisir-h \x \font\A=cmr10 \A  
\setbox0=\vbox{(\unpenalty だるま \showlists}`

---

```
\A (  
\penalty 10000(for kinsoku)  
\x だ  
\x る  
\x ま
```



---

→ 問題とまではいえないが、使うときは注意が必要。

## 1.2. グループ境界での禁則ペナルティ

### 欧文文字の場合

禁則ペナルティは、**禁則対象の欧文文字が和文文字と隣り合う場合にのみ**挿入される。

…はずだが、現在の pTeX では**グループ境界 { や } で隔てられた場合に**欧文文字の禁則ペナルティが 入ったり入らなかったり する。

---

## 1.2. グループ境界での禁則ペナルティ

### 欧文文字の場合

禁則ペナルティは、**禁則対象の欧文文字が和文文字と隣り合う場合にのみ**挿入される。

…はずだが、現在の pTEX では**グループ境界 { や }**で隔てられた場合に欧文文字の禁則ペナルティが🍱入ったり入らなかったり🍱する。

- ▶ ベースライン補正が 0 のとき：
  - ▶ `\postbreakpenalty` は大丈夫らしい
  - ▶ `\prebreakpenalty` はグループ境界で失敗🍱
- ▶ ベースライン補正が 0 でないとき：
  - ▶ `\postbreakpenalty` も `\prebreakpenalty` も、グループ境界で失敗🍱

## 1.2. グループ境界での禁則ペナルティ

`\ybaselineshift=0pt` の場合 :

▶ `\setbox0=\vbox{(漢 ({漢}({}漢 \showlists}`

---

```
\displace 0.0
\A (
\penalty 10000(for kinsoku)
\x 漢
\A (
\penalty 10000(for kinsoku)
\x 漢
\A (
\penalty 10000(for kinsoku)
\x 漢
```

---

欧文文字の `\postbreakpenalty` は問題なさそう

## 1.2. グループ境界での禁則ペナルティ

`\ybaselineshift=0pt` の場合 :

▶ `\setbox0=\vbox{漢.{漢}.漢{.}\showlists}`

---

```
\displace 0.0
\x 漢
\penalty 10000(for kinsoku)
\A .
\x 漢
\A .
\x 漢
\A .
```

---

欧文文字の `\prebreakpenalty` はグループ境界が挟まると挿入されない？

## 1.2. グループ境界での禁則ペナルティ

`\ybaselineshift=3pt` の場合 :

▶ `\setbox0=\vbox{(漢 ({漢}({}漢 \showlists}`

---

```
\displace 3.0
\A (
\penalty 10000(for kinsoku)
\displace 0.0
\x 漢
\displace 3.0
\A (
\displace 0.0
\x 漢
\displace 3.0
\A (
\displace 0.0
\x 漢
```

---

今度は `\postbreakpenalty` も **グループ境界で NG?**

## 1.2. グループ境界での禁則ペナルティ

`\ybaselineshift=3pt` の場合 :

▶ `\setbox0=\vbox{漢.{漢}.漢{.}\showlists}`

---

```
\displace 0.0
\x 漢
\penalty 10000(for kinsoku)
\displace 3.0
\A .
\displace 0.0
\x 漢
\displace 3.0
\A .
\displace 0.0
\x 漢
\displace 3.0
\A .
\displace 0.0
```

---

やはり `\prebreakpenalty` はグループ境界で **NG** ?



## 1.3. リガチャに対する禁則ペナルティ

リガチャ（合字）といえは：

- ▶  $\langle f \rangle + \langle i \rangle$  で  $\langle fi \rangle$  ではなく  $\langle fi \rangle$
- ▶  $\langle f \rangle + \langle f \rangle + \langle i \rangle$  で  $\langle ffi \rangle$  ではなく  $\langle ffi \rangle$

など、ある並びの**2文字以上が合体して1文字に**

- ▶ `\font\x=upjisir-h \x \font\A=cmr10 \A  
\setbox0=\vbox{f i fi ffi\showlists}`

---

```
\A f  
\glue 3.33333 plus 1.66666 minus 1.11111  
\A i  
\glue 3.33333 plus 1.66666 minus 1.11111  
\A ^^L (ligature fi)  
\glue 3.33333 plus 1.66666 minus 1.11111  
\A ^^N (ligature ffi)
```

---

→ 元とは違う**単一の文字コード**に置き換わる

## 1.3. リガチャに対する禁則ペナルティ

禁則ペナルティを設定したくなるリガチャ：

- ▶  $\langle ' \rangle + \langle ' \rangle$  で  $\langle '' \rangle$  ではなく  $\langle \text{”} \rangle$
- ▶  $\langle \text{‘} \rangle + \langle \text{‘} \rangle$  で  $\langle \text{“} \rangle$  ではなく  $\langle \text{“} \rangle$

→ さて，禁則ペナルティはどう設定すればよい？

## 1.3. リガチャに対する禁則ペナルティ

禁則ペナルティを設定したくなるリガチャ：

- ▶ `<'>+<'>` で `<'>` ではなく `<'>`
- ▶ `<'>+<'>` で `<'>` ではなく `<'>`

→ さて，禁則ペナルティはどう設定すればよい？

現在の pT<sub>E</sub>X では…

- ▶ `\postbreakpenalty` はリガチャ自体の文字コードから決まる.
- ▶ `\prebreakpenalty` はリガチャ自体ではなく，それを構成する最初の要素文字コードから決まる.

## 1.3. リガチャに対する禁則ペナルティ

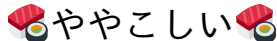
禁則ペナルティを設定したくなるリガチャ：

- ▶ `<'>+<'>` で `<'>` ではなく `<'>`
- ▶ `<'>+<'>` で `<'>` ではなく `<'>`

→ さて，禁則ペナルティはどう設定すればよい？

現在の pT<sub>E</sub>X では…

- ▶ `\postbreakpenalty` はリガチャ自体の文字コードから決まる.
- ▶ `\prebreakpenalty` はリガチャ自体ではなく，それを構成する最初の要素文字コードから決まる.




## 1.3. リガチャに対する禁則ペナルティ

例：upLaTeX で T1 エンコーディングを仮定

- ▶  $\langle \rangle$  +  $\langle \rangle$  は合字  $\langle \rangle$  を作る.
  - ▶ 合字の文字コードは  $\text{\textasciitilde}$  の位置.
  - ▶ Unicode では U+00BB にあたる.

ここで次の入力を考えると…

```
\documentclass{article}
\usepackage[utf8]{inputenc} % LaTeX 既定
\usepackage[T1]{fontenc}
\prebreakpenalty'\textasciitilde=2560
\prebreakpenalty'\>=128
\begin{document}
\setbox0=\hbox{あ>> い \char'\textasciitilde う} \showbox0
```

→ 3 回  $\langle \rangle$  が出力されるが、ペナルティ値は不統一 

# 1.3. リガチャに対する禁則ペナルティ

- ▶ `\setbox0=\hbox{あ>> い \char'\^^T う} \showbox0`

---

```
. \displace 0.0
. \x あ
. \penalty 128(for kinsoku)
. \glue(\xkanjiskip) 2.5 plus 2.5 minus 1.25
. \A ^^T (ligature >>)
. \glue 2.5 plus 1.99997 minus 1.00006
. \x い
. \penalty 2560(for kinsoku)
. \A ^^T
. \x う
. \penalty 2560(for kinsoku)
. \A ^^T
```

---

- ▶ `<>` + `<>` の合字 (あの後) : 要素 `<>` に由来する 128
- ▶ 文字コード直接指定 (いの後) : 合字 `<»` に由来する 2560
- ▶ UTF-8 入力 (うの後) : 合字 `<»` に由来する 2560

→ 現れるすべての `<»` を行頭禁則にするには, `<>` と `<^^T` の両方に `\prebreakpenalty` を設定しないといけない 🍷

# もっとややこしい話

欧文文字と和文文字の連続箇所は処理が複雑.

- ▶ 禁則ペナルティ
  - ▶ 挿入される値は「`\post...` はリガチャ自体, `\pre...` は最初の構成要素」で決まる 🍡
- ▶ 和文文字の JFM 由来グルー・カーン
- ▶ 欧文文字の TFM 由来カーン
- ▶ ベースライン補正用ノード
- ▶ 和欧文間の四分アキ (`\xkanjiskip`)
  - ▶ 挿入可否は「和文文字に隣接する構成要素の `\xspcode`」で決定 🍡

→ しかも, **どの順番で挿入するか**も割と問題.

グルー境界 (1.2) とリガチャ (1.3) の挙動については, GitHub にて `kinsoku_around_word` ブランチまたは `kinsoku_ligature` ブランチで検討中だが, 未完成

## 2. 和文文字に対する禁則処理

pT<sub>E</sub>X の和欧文混植は難しい 🍣🍱

しかし，和文文字の禁則処理についてもいくつかの課題が残っている。



## 2.1. 和文文字の禁則ペナルティ取得

### 1.1 で述べた

- ▶ 和文文字の `\postbreakpenalty` は「現在のリストの最後」になりうるため、`\lastpenalty` で値を取得できるし、`\unpenalty` で取り除くこともできる。

の「少々デリケートな挙動」の話。

## 2.1. 和文文字の禁則ペナルティ取得

min10系 (p $\TeX$ ), jis系 (jsclasses), upjis系 (up $\TeX$ ) の JFM の場合

▶ `\font\x=upjisr-h \x`  
`\setbox0=\vbox{あ (\showthe\lastpenalty)}`

---

> 10000.

---

▶ `\font\x=upjisr-h \x`  
`\setbox0=\vbox{あ (\unpenalty\showthe\lastpenalty)}`

---

> 0. % 元々 10000 あったものが消えた

---

▶ `\font\x=upjisr-h \x`  
`\setbox0=\vbox{あ (\ifnum\lastpenalty=10000 \T \fi)}`

---

! Undefined control sequence. \T

---

## 2.1. 和文文字の禁則ペナルティ取得

jlreq クラスの JFM の場合

- ▶ `\font\X=ujlreq \X`  
`\setbox0=\vbox{あ (\showthe\lastpenalty)}`

---

> 0. % あれっ？

---

- ▶ `\font\X=ujlreq \X`  
`\setbox0=\vbox{あ (\unpenalty\showthe\lastpenalty)}`

---

> 0. % 元々 0 だったので `\unpenalty` とは関係ない

---

- ▶ `\font\X=ujlreq \X`  
`\setbox0=\vbox{あ (\ifnum\lastpenalty=10000 \T \fi)}`

---

! Undefined control sequence. \T % 0 なら出ないはず

---

## 2.1. 和文文字の禁則ペナルティ取得

min10系 (p $\TeX$ ), jis系 (jsclasses), upjis系 (up $\TeX$ ) のJFMの場合

- ▶ `\font\x=upjisr-h \x`  
`\setbox0=\vbox{あ (\showlists)}`

---

```
\x あ
\glue(refer from jfm) 5.0 minus 5.0
\x (
\penalty 10000(for kinsoku)
```

---

- ▶ `\font\x=upjisr-h \x`  
`\setbox0=\vbox{あ (\unpenalty\showlists)}`

---

```
\x あ
\glue(refer from jfm) 5.0 minus 5.0
\x (
```

---

→ リストの最後が `\penalty` ノードなので取得・削除可

## 2.1. 和文文字の禁則ペナルティ取得

jlreq クラスの JFM の場合：`< (>` の後に **JFM** グループを設定

- ▶ `\font\X=ujlreq \X`  
`\setbox0=\vbox{あ (\showlists)}`

---

```
\X あ
\glue(refer from jfm) 5.0 minus 5.0
\X (
\penalty 10000(for kinsoku)
\glue(refer from jfm) 0.0
```

---

- ▶ `\font\X=ujlreq \X`  
`\setbox0=\vbox{あ (\unpenalty\showlists)}`

---

```
\X あ
\glue(refer from jfm) 5.0 minus 5.0
\X (
\penalty 10000(for kinsoku)
\glue(refer from jfm) 0.0
```

---

→ `\penalty` ノードより後に JFM グループがあるので取得・削除に失敗

## 2.1. 和文文字の禁則ペナルティ取得

### pTeX の JFM グルー挿入規則

- ▶ 同じ箇所では「禁則ペナルティ → JFM グルー」の順に挿入  
JFM グルーのほうが後
- ▶ 展開不能トークンが来たら、文字タイプ 0 とみなして JFM グルー挿入 `\relax`, `\showthe`, `\unpenalty`, 代入も該当

## 2.1. 和文文字の禁則ペナルティ取得

### pTeX の JFM グルー挿入規則

- ▶ 同じ箇所では「禁則ペナルティ → JFM グルー」の順に挿入  
JFM グルーのほうが後
- ▶ 展開不能トークンが来たら、文字タイプ0とみなして JFM グルー挿入 `\relax`, `\showthe`, `\unpenalty`, 代入も該当

jlreq クラスの JFM の場合：`< (>` の後に **JFM グルー** を設定

- ▶ `\setbox0=\vbox{あ (\ifnum\lastpenalty>0 \T \fi}`

---

```
! Undefined control sequence. \T
```

---

→ `\ifnum` の展開 (=判定) 時点でグルー挿入はまだ!

- ▶ `\setbox0=\vbox{あ (\relax\ifnum\lastpenalty>0 \T \fi}`



---

```
% エラーなし
```


---

→ 展開不能トークン `\relax` の時点でグルーが挿入された

## 2.1. 和文文字の禁則ペナルティ取得

 pL<sup>A</sup>T<sub>E</sub>X カーネル (plcore.ltx) より 

```
\def\footnotetext{%
  \ifhmode\pltx@foot@penalty\lastpenalty\unpenalty\fi % ここで取得→削除
  \@ifnextchar [\@xfootnotenext
    {\protected@xdef\@thefnmark{\thempfn}%
     \@footnotetext}}
\long\def\@footnotetext#1{%
  \ifydir\def\@tempa{\yoko}\else\def\@tempa{\tate}\fi
  \insert\footins{\@tempa%
    ... 中略 ...
    \color@begingroup
      \@makefntext{%
        \rule\z@\footnotesep\ignorespaces#1\@finalstrut\strutbox}%
      \color@endgroup}\ifhmode\null\fi
  \ifnum\pltx@foot@penalty=\z@\else
    \penalty\pltx@foot@penalty % ここで復帰
    \pltx@foot@penalty\z@
  \fi}
```

→ jlreq クラスでは思うように動いていない 

取得・削除という操作自身が展開不能なため



## 2.2. 禁則ペナルティの合算

### pTeX の禁則ペナルティ挿入

- ▶ `\prebreakpenalty` は、既にその位置にペナルティがあればそれに**合算**する形で発行される（別には発行されない）。
- ▶ `\postbreakpenalty` は、直後に（`\prebreakpenalty` 以外の）ペナルティが来ても合算されず、常に**別々**に発行される。

## 2.2. 禁則ペナルティの合算

### pT<sub>E</sub>X の禁則ペナルティ挿入

- ▶ `\prebreakpenalty` は、既にその位置にペナルティがあればそれに合算する形で発行される（別には発行されない）。
- ▶ `\postbreakpenalty` は、直後に (`\prebreakpenalty` 以外の) ペナルティが来ても合算されず、常に別々に発行される。

- ▶ `\setbox0=\vbox{あ} \showlists}`

---

```
\x あ
\penalty 10000(for kinsoku)    % => 10000
\x )
```

---

- ▶ `\setbox0=\vbox{ ( ) } \showlists}`

---

```
\x (
\penalty 20000(for kinsoku)    % => 10000 + 10000 = 20000
\x )
```

---

## 2.2. 禁則ペナルティの合算

例：`\prebreakpenalty`の前に明示的な`\penalty`を入れてみる

▶ `\setbox0=\vbox{あ \penalty5000 } \showlists}`

---

```
\x あ
\penalty 15000                % => 5000 + 10000 = 15000
\x )
```

---

## 2.2. 禁則ペナルティの合算

例：`\prebreakpenalty` の前に明示的な `\penalty` を入れてみる

▶ `\setbox0=\vbox{あ \penalty5000 } \showlists`

---

```
\x あ
\penalty 15000           % => 5000 + 10000 = 15000
\x )
```

---

▶ `\setbox0=\vbox{あ \break } \showlists`

---

```
\x あ
\penalty 0              % => -10000 + 10000 = 0
\x )
```

---

→ `\break` すなわち `\penalty -10000` を入れても強制改行できない 🍒

→ 合算されるのを見越して `\penalty -20000` を入れのも面倒 🍒…

# 今日のトピックまとめ

Why GitHub? Enterprise Explore Marketplace Pricing

Search Sign in Sign up

texjorg / tex-jp-build

Watch 15 Star 11 Fork 4

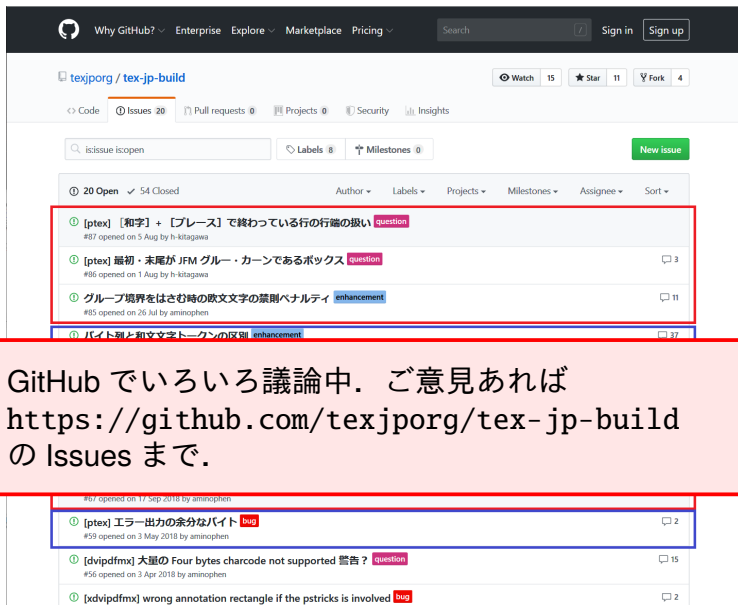
Code Issues 20 Pull requests 0 Projects 0 Security Insights

is:issue is:open Labels 0 Milestones 0 New issue

20 Open 54 Closed Author Labels Projects Milestones Assignee Sort

- [ptex] [和字] + [ブレース] で終わっている行の行端の扱い **question**  
#87 opened on 5 Aug by h-kitagawa
- [ptex] 最初・末尾が JFM グルー・カンであるボックス **question** 3  
#86 opened on 1 Aug by h-kitagawa
- グループ境界をはさむ時の欧文文字の禁則ペナルティ **enhancement** 11  
#85 opened on 26 Jul by aminophen
- バイト列と和文字トークンの区別 **enhancement** 37  
#81 opened on 8 Jun by h-kitagawa
- [ptexenc] Inconsistent error message **question** 9  
#80 opened on 29 May by aminophen
- [ppltotf, ptftopl] 9 区から 15 区, および 85 区以上の和文字 6  
#78 opened on 5 Apr by h-kitagawa
- [XeTeX] SourceHanSerif で 数字の ToUnicode が入らない 11  
#75 opened on 26 Feb by aminophen
- \prebreakpenalty の合算と JFM グルー, \lastpenalty 4  
#67 opened on 17 Sep 2018 by aminophen
- [ptex] エラー出力の余分なバイト **bug** 2  
#59 opened on 3 May 2018 by aminophen
- [dviptdmtx] 大量の Four bytes charcode not supported 警告? **question** 15  
#56 opened on 3 Apr 2018 by aminophen
- [xdvipdmtx] wrong annotation rectangle if the pstricks is involved **bug** 2

# 今日のトピックまとめ



The screenshot shows the GitHub interface for the repository `texjorg/tex-jp-build`. The top navigation bar includes links for "Why GitHub?", "Enterprise", "Explore", "Marketplace", and "Pricing", along with a search bar and "Sign in" / "Sign up" buttons. The repository header shows "20 Issues", "0 Pull requests", "0 Projects", "0 Security", and "Insights". A search bar contains the text "is:issue is:open". A "New issue" button is visible on the right. The issues list is sorted by "Open" status, showing 20 open and 54 closed issues. The visible issues are:

- #87: [ptex] [和字] + [ブレース] で終わっている行の行端の扱い (question) - opened on 5 Aug by h-kitagawa
- #86: [ptex] 最初・末尾が JFM グルー・カーンであるボックス (question) - opened on 1 Aug by h-kitagawa (3 comments)
- #85: グループ境界をはさむ時の欧文文字の禁則ペナルティ (enhancement) - opened on 26 Jul by aminophen (11 comments)
- #84: バイト列と欧文文字トークンの区別 (enhancement) - opened on 17 Sep 2018 by aminophen (37 comments)
- #59: [ptex] エラー出力の余分なバイト (bug) - opened on 3 May 2018 by aminophen (2 comments)
- #56: [dvi] 大量の Four bytes charcode not supported 警告? (question) - opened on 3 Apr 2018 by aminophen (15 comments)
- #55: [xdvipdmtx] wrong annotation rectangle if the pstricks is involved (bug) - opened on 3 Apr 2018 by aminophen (2 comments)

GitHub でいろいろ議論中。ご意見あれば  
<https://github.com/texjorg/tex-jp-build>  
の Issues まで。