

bxcoloremoji パッケージ

八登崇之 (Takayuki YATO; aka. “ZR”)

v1.1 [2026/06/22]

概要

bxcoloremoji パッケージは L^AT_EX 文書においてカラー絵文字を出力するためのものである。他の類似の機能をもつパッケージと比較すると、本パッケージは以下の特徴をもつ。

- 全ての主要な L^AT_EX エンジンをサポートする。
- 文字自体、Unicode 符号値、短縮名での入力が可能。
- hyperref 使用時に PDF 文字列の中でも妥当に動作する。
- 和文組版環境でも絵文字を適切に取り扱える。

配布物の容量を抑えるため、1.0 版からは最終的な画像出力を twemojis パッケージに移譲することを既定の動作としているが、旧版で用いていた画像セットを用いる設定もできる。また、ユーザが用意した独自の画像セット（カスタムファミリー）を出力に利用する設定もサポートされている。

目次

1	前提環境	2
2	パッケージ読込	2
3	パラメタ設定	4
4	使い方	4
4.1	基本的な絵文字出力命令	5
4.2	キーキャップ絵文字を出力するカウンタ出力命令	5
4.3	pifont パッケージ類似の機能	5
5	絵文字の“短縮名”	6
6	PDF 文字列中での絵文字の利用	7
7	カスタムファミリー	8
7.1	手順①：設定ファイルを作成する	8
7.2	手順②：画像ファイルを改名して配置する	8
8	0.x 版のユーザのための情報	9
8.1	0.x 版からのアップデート	9
8.2	bxcoloremoji-oldstd パッケージ	9

1 前提環境

- フォーマット： \LaTeX
- エンジン： $\epsilon\text{-TeX}$ 拡張をサポートするもの
- DVI ウェア (DVI 出力時)： 用いる画像形式に対応したもの
- 依存パッケージ：
 - etoolbox
 - binhex (expl3 が有効でない場合)
 - bxghost (条件により)
 - twemojis

※ 0.x 版からのアップデートについては 8 節を参照してほしい。

2 パッケージ読込

DVI 出力の場合、事前に `graphicx` パッケージと `color` (または `xcolor`) パッケージ^{*1}と `twemojis` パッケージ^{*2}を読み込む必要がある。

※ PDF 出力の場合、およびグローバルのドライバオプションが指定されている場合は、自動的に `graphicx` と `color`^{*3}と `twemojis`^{*4}がオプション無しで読み込まれる。

※絵文字画像出力にカスタムファミリのみを利用する場合は、`twemojis` の読込は不要である。

```
\usepackage[dvipdfmx]{graphicx,color} % dvipdfmx の場合
\usepackage{twemojis}
```

また、昔の (2018-04-01 以前の) (pdf) \LaTeX および p \LaTeX の場合は、`utf8` 入力エンコーディングを有効化する必要がある。

```
\usepackage[utf8]{inputenc}
```

その後に `bxcoloremoji` パッケージを読み込む。

```
\usepackage[<オプション>]{bxcoloremoji}
```

利用可能なオプションは以下の通り。

- **設定パラメタ**： 3 節に挙げる設定パラメタ (例えば `twemoji-png` や `scale` 等) はパッケージオプションとしても指定できる。
- `jatype=<値>`： 絵文字を和文文字として扱うか。
 - ※この値に関わらず、* 付の命令 (`\coloremoji*` 等) は常に和欧文中立な “画像” として扱われる。

^{*1} 現状では (x)color パッケージの機能はある種のフォールバック出力の際にのみ用いられる。(x)color が未読込の場合は色を使わず黒色で出力する。(実際に絵文字画像が出力される場合には影響しない。)

^{*2} `twemojis` パッケージ： <https://ctan.org/pkg/twemojis>

^{*3} `color` パッケージの自動読込は \LaTeX カーネルの版が 2021-06-01 以降である場合にのみ行われる。

^{*4} パッケージオプションにおいてカスタムファミリの使用を設定している場合は `twemojis` は読み込まれない。

※和文扱いを行わない場合は * 無と * 付はどちらも“画像扱い”になるが size/size* 指定時は異なる出力になりうる。

- auto (既定): (u)pL^AT_EX であるかまたは LuaL^AT_EX で LuaTeX-j_a が用いられている場合に和文扱いを行う。
- true: 可能な限り、和文扱いを行うを試みる。
※現状では pdfL^AT_EX/欧文 L^AT_EX では効果がない。
- false: 和文扱いを行わない。
- names=(真偽値): 短縮名データベースを読み込むか。既定値は true。
※ false を指定した場合、独自短縮名 (5 節参照) のみが使用可能になる。

以下は上級者向けの設定。


- nodvidriver: ドライバ依存の動作を抑止する。具体的には、絵文字画像セット種別が no-image に固定される (絵文字は表示されない)。
- resetdvidriver: nodvidriver の否定。
- preload-names=(値): 短縮名データベースをパッケージ読込時に一括して読み込むか。
※ false は昔の“メモリ容量が少ない T_EX エンジン”向けの設定。
 - auto (既定): 以下の条件の何れかを満たす場合は (メモリが十分にあると判断して) true、それ以外は false。
 - * エンジンが X_YL^AT_EX/LuaL^AT_EX/upL^AT_EX である。
 - * expl3 が有効である。
 - * hyperref が読み込まれている。
 - true: パッケージ読込時に全てのデータを読み込む。
 - false: 必要に応じて読み込む。
- bbparam=(値): 絵文字出力の \includegraphics に bb パラメタを指定するか。
 - auto (既定): graphicx のドライバが dvipdfmx か dvisvgm である場合にのみ指定する。
※ dvipdfmx / dvisvgm の場合、bb を指定した方が動作が速い。
 - true/false: 常に指定する/しない。
※ bb 設定が禁止されているドライバもあるので注意。
- bxghost=(値): 和文ゴースト処理のために bxghost パッケージ^{*5}を利用するか。
※ jatype が偽の場合は和文ゴースト処理自体が行われないため無効 (利用しない)。
 - auto (既定): 次の条件を全て満たす場合に利用する。
 - * エンジンが (u)pL^AT_EX である、または、LuaL^AT_EX であってかつ本パッケージ読込時点で LuaTeX-j_a が読み込まれている。
 - * bxghost の 0.3.0 版以降がインストールされている。
 - true: 利用する (常に bxghost を読み込もうとする)。
※ LuaL^AT_EX 上では bxghost は LuaTeX-j_a を必ず読み込むことに注意。
 - false: 利用しない。
※この場合、和文ゴースト処理が失敗する可能性がある。

^{*5} bxghost パッケージ: <https://ctan.org/pkg/bxghost>

- `pua=〈真偽値〉`： Unicode 私用領域の文字を絵文字として扱うか。既定値は `false`。

3 パラメタ設定

パラメタ設定は以下の方法で利用できる。

- パッケージオプションに指定する。
`\usepackage[twemoji-png,scale=2]{bxcoloremoji}`
- `\coloremojisetaup` 命令の引数に指定する。その場で設定が変更されて、以降の絵文字出力命令に適用される。
`\coloremojisetaup{twemoji-png,scale=2}`
- 絵文字出力命令の先頭のオプション引数に指定する。その絵文字出力にのみ設定が適用される。
`\coloremoji[twemoji-png,scale=2]{`

利用可能なパラメタは以下の通り。

- **絵文字画像セットの種別を指定するもの。**(既定値 = `twemojis`^{*6})
 - `twemojis`： 画像出力を `twemojis` パッケージの絵文字出力命令に移譲する (`twemojis` モード)。
 - `family=〈名前〉`： カスタムファミリー指定 (7 節参照)。
 - `no-image`： 絵文字画像を使わず全てフォールバック出力にする。
 - `twemoji-pdf` / `twemoji-png`： これらは 0.x 版の「標準の画像セット」を指定するオプションであったが、1.0 版以降ではこれらは「その名前のカスタムファミリーの指定」と見なされる。つまり、`twemoji-pdf` は `family=twemoji-pdf` と同等である。
 ※ 0.x 版の「標準の画像セット」の利用については 8 節を参照。
- `size=〈長さ〉`： * 無命令での絵文字のサイズ。
- `size*=〈長さ〉`： * 付命令での絵文字のサイズ。
 ※ `size` および `size*` の既定値は (u)pL^AT_EX では 1zw、LuaL^AT_EX + LuaT_EX-j_a では 1\zw、それ以外は 1em。この既定値は `jatype` の設定には影響されない。
- `scale=〈実数〉`： 絵文字のサイズを、`size` / `size*` で指定した値からさらに指定の倍率で変更する。(既定値 = 1)

4 使い方

命令・環境名が `[*]` 付 (例えば `\coloremoji[*]`) で示されている場合、実際には「* 無」(`\coloremoji`) と「* 付」(`\coloremoji*`) の変種が存在することを示す。両者の違いは以下の通り。

- `jatype` オプションにより和文扱いが有効になっている場合は、* 無は和文扱いで、* 付は和欧文中立な“画像扱い”になる。ただし数式中は両者ともに“画像扱い”になる。
- `size` は * 無にのみ、`size*` は * 付にのみ適用される。(scale は両方に適用される。)

^{*6} 1.0 版より既定値が `twemojis` に変更された。また 0.12 版から非推奨となっていた `twitter`、`hires`、`lowres` は廃止された。

[*] 以外の [...] 表記はオプション引数で、これは実際に [] で囲った形で指定する。

※本パッケージの命令については、必須引数を囲む {} は省略できない。

4.1 基本的な絵文字出力命令

- `\coloremojisetup{<設定>}`: パラメタ設定 (3 節参照) を変更する。
※ <設定> は “<キー>=<値>,…” の形のリスト。以降も同様。
- `\coloremoji[*][<設定>]{<文字列>}`: 引数の文字列を絵文字として出力する。ただし、対象の画像がないなどの理由で絵文字として出力できない場合は、通常のテキスト出力にフォールバックする。
`\coloremoji{🍣3🍵🍡}%出力: 🍣3🍵🍡`
- `\coloremojicode[*][<設定>]{<符号値列>}`: 文字を「Unicode 符号値」または「JoyPixels の emoji-toolkit ライブラリ^{*7}で規定された短縮名」で入力してカラー絵文字を出力する。引数は、符号値で指定する場合はその 16 進表記、短縮名で指定する場合は “:<短縮名>:” の形式で入力し、複数文字を入力する場合は各文字の指定を空白区切りで並べる。
`\coloremojicode{sushi: 23 20E3 1F643 :snowman:}%出力: 🍣#🍵🍡`
※以降、この入力方式を「符号値列」と呼称する。
- `\coloremojiucs[*][<設定>]{<符号値列>}`: `\coloremojicode` の旧版における別名。
※他の `coloremojicode` の名前の命令・環境については、0.4 版以前から存在するものについては、同様に `coloremojiucs` という別名が用意されている。

4.2 キーキャップ絵文字を出力するカウンタ出力命令

0~10 の整数値を、値に対応するキーキャップ絵文字 (0~10) で出力する。

※入力値が範囲外の場合はフォールバック出力になる。

- `\coloremojikeycapof[*][<設定>]{<整数>}`: 入力の整数値に対応するキーキャップ絵文字を出力する。
`\coloremojikeycapof{8}%出力: 8`
- `\coloremojikeycap[*][<設定>]{<カウンタ名>}`: 指定のカウンタの現在の値に対応するキーキャップ絵文字を出力する。
- `\pagenumbering{coloremojikeycap}`: ページ番号の形式をキーキャップ絵文字に変更する。
※本文書で実際に `\pagenumbering{coloremojikeycap}` が指定されている。

4.3 pifont パッケージ類似の機能






































pifont パッケージの機能 (`\dingfill` 命令、`dingautolist` 環境など) の絵文字版に相当する、以下の命令が提供される。

※この小節で挙げる機能の絵文字出力は、常に * 付命令であるかのように動作する。また、* やパラメタ設定オプションを付けることはできない。

^{*7} emoji-toolkit ライブラリ: <https://github.com/joypixels/emoji-toolkit>

- 1 `\coloremojifill{<文字列>}`: 充填命令 (`\dotfill` の類) の一種で、`\coloremoji*{<文字列>}` の出力を複数並べて行を充填する。
- 2 `\coloremojiline{<文字列>}`: 絵文字による飾り罫を出力する。すなわち `\coloremojifill{<文字列>}` の出力 (ただし両端に若干の空きを入れる) のみを含む独立した行を出力する。
- 3 `\begin{coloremojilist}{<文字列>}~\end{coloremojilist}`: `\coloremoji{<文字列>}` の出力を項目ラベルとする箇条書きを出力する。
- 4 `\begin{coloremojiautolist}{<文字列>}~\end{coloremojiautolist}`: これも絵文字を項目ラベルとする箇条書きを出力する環境であるが、引数には何れかの「絵文字順序列」に含まれる絵文字の一つを指定する必要がある。その文字から始まる順序列に従ってラベルを指定する。例えば `\begin{coloremojiautolist}{♠}` では、先頭のラベルが「♠」となり以下「♥」「♦」「♣」と続く。
※現状の実装では順序列の末尾に達した場合は先頭に戻る (つまり「♣」の次は「♠」になる) が、これは将来的に変更される可能性がある。
※この箇条書きは `\begin{coloremojiautolist}{1}` で生成されている。
- 5 以上の命令・環境について、引数に符号値列を指定する版も存在する。
 - ✏ `\coloremojicodefill{<符号値列>}`
 - ✏ `\coloremojicodeline{<符号値列>}`
 - ✏ `\begin{coloremojicodelist}{<符号値列>}`
 - ✏ `\begin{coloremojicodeautolist}{<符号値列>}`

現状では、以下に挙げる「絵文字順序列」が定められている。

-  →  →  →  →  →  →  →  →  →  → 
-  →  →  → 
-  →  →  →  →  →  →  →  →  →  → 
-  →  →  →  →  →  →  →  →  →  → 

5 絵文字の“短縮名”

「符号値列」中で用いる絵文字の短縮名については JoyPixels (旧称 EmojiOne) の emoji-toolkit ライブラリで規定する名前が利用できる。その他に表 1 にある**独自短縮名**が利用できる。

※これらは emoji sequence の入力の便宜のためのものである。

※独自短縮名については名前を囲む `:~:` は省略可能である。

※独自短縮名以外の短縮名については、現状では「整数の 16 進表記と解釈できないものは `:~:` が省略可能」という仕様であるが、これは将来変更される可能性がある。

使用例:

```
\coloremojicode{man: + :woman: + :girl: + :girl:}%出力:👨👩👧👧
\coloremojicode{!flag @g @b @w @l @s @}%出力:🚩🇬🇧🇨🇦🇺🇸
\coloremojicode{1F647 + !male}%出力:👨🎓
```

表 1 独自短縮名の一覧

+	U+200D		(ZWJ)
/1	U+1F3FB		(light skin tone 修飾子)
/2	U+1F3FC		(medium-light skin tone 修飾子)
/3	U+1F3FD		(medium skin tone 修飾子)
/4	U+1F3FE		(medium-dark skin tone 修飾子)
/5	U+1F3FF		(dark skin tone 修飾子)
!/red	U+1F9B0		(“+ !/red” で赤髪の hair style)
!/curly	U+1F9B1		(“+ !/curly” で巻毛の hair style)
!/bald	U+1F9B2		(“+ !/bald” で禿頭の hair style)
!/white	U+1F9B3		(“+ !/white” で白髪の hair style)
!black	U+2B1B		(“+ !black” で黒色の color indicator)
!white	U+2B1C		(“+ !white” で白色の color indicator)
!red	U+1F7E5		(“+ !red” で赤色の color indicator)
!blue	U+1F7E6		(“+ !blue” で青色の color indicator)
!orange	U+1F7E7		(“+ !orange” で橙色の color indicator)
!yellow	U+1F7E8		(“+ !yellow” で黄色の color indicator)
!green	U+1F7E9		(“+ !green” で緑色の color indicator)
!purple	U+1F7EA		(“+ !purple” で紫色の color indicator)
!brown	U+1F7EB		(“+ !brown” で茶色の color indicator)
!female	U+2640		(“+ !female” で女性の gender indicator)
!male	U+2642		(“+ !male” で男性の gender indicator)
!flag	U+1F3F4		(旗を表す tag sequence の base 文字)
!<	U+2B05		(“+ !<” で左の direction indicator)
!>	U+27A1		(“+ !>” で右の direction indicator)
!A~!Z	U+1F1E6~1F1FF		(flag sequence の構成要素)
@	U+E007F		(tag sequence の終端)
@0~@9	U+E0030~E0039		(tag sequence の構成要素)
@a~@z	U+E0061~E007A		(tag sequence の構成要素)

6 PDF 文字列中での絵文字の利用

hyperref パッケージ使用時の文書情報文字列（以降では「PDF 文字列」と呼ぶ）の入力の中でも絵文字出力用の命令を使用できる。例えば、\section の引数の中で \coloremoji を含めた場合、版面の上では絵文字（の画像）として出力され、PDF のしおりの中では文字として表示される。

ただし「PDF 文字列中の Unicode 文字が正しく処理される」状態が担保されていることが前提となる。具体的には、次の何れかが満たされる必要がある。

※ T_EX Live 2022 以降を前提とすると、要するに「(u)pL^AT_EX では pxjahyper の併用が必要、それ以外は何も要らない」ということである。

- hyperref の “PDF エンコーディング” が Unicode である。

最近（7.00g 版以降）の hyperref では既定でそうになっている。それより古い版では hyperref の読込時に unicode オプションを付ける必要がある。(u)pL^AT_EX では pxjahyper パッケージを併用する必要がある。


```
\usepackage[unicode]{hyperref}
```

- (hyperref の “PDF エンコーディング” が Unicode ではないが) エンジンが upLaTeX であり、pxjahyper パッケージを併用している。

※ pLaTeX ではそもそも PDF 文字列中に JIS 外の文字を含ませることができないため、この場合は対応ができない。

7 カスタムファミリ

bxcoloremoji では実際の絵文字画像の出力に twemojis の機能を使っているが、その代わりに、ユーザが用意した一連の画像ファイル群を「カスタムファミリ」として登録して表示に使うことができる。

例として、noto-emoji レポジトリ^{*8}の中 (png/128/ 以下) に含まれる一連の PNG 画像を notoemoji ファミリとして登録する手順を示す。

7.1 手順①：設定ファイルを作成する

カスタムファミリの登録には設定ファイルが必要であり、その名前は bxcoloremoji-〈ファミリ名〉.cfg である。今の例では bxcoloremoji-notoemoji.cfg を作成して TeX から見える位置に配置することになる。設定ファイルの書式は以下の通りである。

```
% prefix: 画像ファイルのパス名接頭辞
prefix = notoemoji/notoemoji-
% extension: 画像ファイルの拡張子
extension = png
% bbox = 画像の bounding box の値 (省略可)
bbox = 0 0 128 128
```

bbox は dipdfmx での画像の読込を高速化するための指定であり、省略することもできる。全ての画像ファイルの bounding box が一致しているのではない場合は省略するしかない。

※ bbox が使われるかは実際には bbparam パッケージオプションの指定により決められる。

prefix は画像ファイルの (Kpathsea 上の) パス名を決定するのに使われる。上の設定の場合、例えば、「U+2603 🌨️:snowman2:」の画像ファイルのパス名は notoemoji/notoemoji-2603.png となる。

7.2 手順②：画像ファイルを改名して配置する

パス名の命名規則は以下の通りである。

- 絵文字を構成する Unicode 文字の符号値の 16 進表記 (0 埋め無し、大文字) を順に “-” でつないだものを「符号値列」とする。
※ただし EVS (U+FE0F) は除外される。例えば、**2** <0032 FE0F 20E3> に対する「符号値列」は 32-20E3 となる。
- “<prefix の値><符号値列>.<extension の値>” がパス名である。

^{*8} noto-emoji レポジトリ: <https://github.com/googlefonts/noto-emoji>

noto-emoji の各々の画像ファイルをこの規則に従って配置する。例えば、「U+2603 🍀」の画像ファイル（元の名前は emoji_u2603.png）について、notoemoji/notoemoji-2603.png のパス名で読める位置に配置する。

※例えば、Kpathsea 変数 `$TEXINPUTS` にディレクトリ `~/texmf/tex/latex//` が含まれる場合、画像ファイルを `~/texmf/tex/latex/custom_images/notoemoji/notoemoji-2603.png` に置くことができる。

8 0.x 版のユーザのための情報

8.1 0.x 版からのアップデート

1.0 版より画像出力の部分を twemojis パッケージに（既定で）移譲するようになったため、本パッケージの配布物には画像ファイル（emoji_images/ ディレクトリ下にあった）は含まれなくなった。本パッケージを 0.x 版から 1.0 版以降にアップデートする場合、既存の emoji_images/ ディレクトリの画像ファイル群については、削除してもそのまま残してもかまわない。

もし画像ファイル群を残した場合は、画像セット指定オプションの twemoji-pdf（0.x 版での既定値）と twemoji-png^{*9}は 0.x 版のときと同じように機能する。

※ただしこの場合でも画像セットの既定値は twemojis であることに注意。

※ twemoji-pdf と現在の既定値の twemojis が扱う画像セットはどちらも「Twemoji の SVG 画像を PDF 形式に変換したもの」に由来するため、両者の出力結果はほぼ同一である。

8.2 bxcloremoji-oldstd パッケージ

本パッケージの 0.x 版の配布物に含まれていた画像ファイル群は、現在では bxcloremoji-oldstd パッケージ^{*10}で配布されている。インストール方法はそちらの説明資料を参照してほしい。

bxcloremoji-oldstd パッケージは「公式が用意しているカスタムファミリー」という位置づけである。インストールすると、twemoji-pdf と twemoji-png の 2 つのカスタムファミリーが使用可能になる。^{*11}

※ bxcloremoji-oldstd をインストールする場合には、0.x 版の画像ファイル群は削除する必要がある。

^{*9} twemoji-pdf/twemoji-png のオプションは現在ではカスタムファミリー指定の一種と見なされる。従って family=twemoji-pdf 等と書いても同じ動作になる。

^{*10} bxcloremoji-oldstd パッケージ： <https://github.com/zr-tex8r/bxcloremoji-oldstd>

^{*11} 従って、例えば twemoji-pdf のカスタムファミリーを使うには family=twemoji-pdf と指定すればよいが、先述の通り、これは twemoji-pdf と書いてもよい。