

bxcoloremoji パッケージ

八登崇之 (Takayuki YATO; aka. “ZR”)

v0.22 [2024/09/16]

概要

bxcoloremoji パッケージは \LaTeX 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 \LaTeX 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。

目次

1	前提環境	1
2	パッケージ読込	2
3	パラメタ設定	3
4	使い方	4
4.1	基本的な絵文字出力命令	4
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	6
7	カスタムファミリ	7
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	twemojis モード	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 (twemojis モードの場合)

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}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`^{*2}がオプション無しで読み込まれる。

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

また、昔の (2018-04-01 以前の) (pdf)LaTeX および pLaTeX 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。
 ※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)pLaTeX であるかまたは LuaLaTeX で LuaTeX-japan が用いられている場合に和文扱いを行う。
 - `true`： 可能な限り、和文扱いを行うことを試みる。
 ※現状では pdfLaTeX/欧文 LaTeX では効果がない。
 - `false`： 和文扱いを行わない。

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

^{*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} 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。

固定される（絵文字は表示されない）。

- `resetdvidriver`： `nodvidriver` の否定。
- `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。
- `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 TeX 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 XeLaTeX / LuaLaTeX / upLaTeX 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` である場合にのみ指定する。
※ `dvipdfmx` 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^{*3}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)pLaTeX である、または、LuaLaTeX であってかつ本パッケージ読込時点で LuaTeX-jā 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）。
※ LuaLaTeX 上では `bxghost` は LuaTeX-jā 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。
- `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}`
- `\coloremojisetup` 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。
`\coloremojisetup{twemoji-png,scale=2}`

^{*3} `bxghost` パッケージ： <https://ctan.org/pkg/bxghost>

- 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 = `twemoji-pdf`)
 - `twemoji-pdf`: `twemoji` の SVG 画像から変換した PDF 画像。
 - `twemoji-png`: `twemoji` の 72 ピクセルの PNG 画像。
 - `twemojis`: 画像出力を `twemojis` パッケージ^{*4}の絵文字出力命令に移譲する (`twemojis` モード)。
※詳細は 8 節を参照。
 - `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。
- `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` オプションは * 無にのみ、`size*` オプションは * 付にのみ適用される。(scale は両方に適用される。)

[*] 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-

^{*4} twemojis パッケージ: <https://ctan.org/pkg/twemojis>

toolkit ライブラリ^{*5}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 (🅐～🅑) 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: 🅑
- `\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。

- 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。

^{*5} emoji-toolkit ライブラリ: <https://github.com/joypixels/emoji-toolkit>

※この箇条書きは `\begin{coloremojiautolist}{1}` で生成されている。

- ✏ `\coloremojicodedefill{⟨符号值列⟩}`
- ✏ `\coloremojicodeline{⟨符号值列⟩}`
- ✏ `\begin{coloremojicodelist}{⟨符号值列⟩}`
- ✏ `\begin{coloremojicodeautolist}{⟨符号值列⟩}`

- 
- 
- 
- 

```
\coloremojicode{:man: + :woman: + :girl: + :girl:}%出力: 🧑👩👧👧
\coloremojicode{!flag @g @b @w @l @s @}%出力: 🇬🇧
\coloremojicode{!F647 + !male}%出力: 🧑🏻
```

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

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)

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 upL^AT_EX であり、pxjahyper パッケージを併用している。

※ pL^AT_EX 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 では実際の絵文字の表示に twemoji の画像を使っているが、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^{*6}の中 (`png/128/` 以下) に含まれる一連の PNG 画像を `notoemoji` ファミリー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

カスタムファミリーの登録には設定ファイルが必要であり、その名前は `bxcoloremoji-〈ファミリー名〉.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。

`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。

^{*6} `noto-emoji` レポジトリ: <https://github.com/googlefonts/noto-emoji>

8 twemojis モード

0.17 版から「twemojis モード」がサポートされた。このモードでは絵文字出力の方法が「用意した画像ファイルを読み込む」ものから「twemojis パッケージ^{*7}の命令を実行する」ものに変更される。

これにより、大規模な画像ファイルセットのインストールが不要になるというメリットが得られる。もちろん、代わりに twemojis パッケージのインストールが必要になるが、twemojis は TeX Live に含まれているので、全体の手間は減ることが期待できるだろう。

twemojis モードに切り替えるには設定パラメタとして twemojis を指定する。例えば、パッケージオプションに指定すればよい。

```
\usepackage[twemojis]{bxcoloremoji}
```

補足事項：

- twemojis モードは「絵文字画像セットの選択肢の一つ」という位置づけである。従って「サポートされる絵文字の範囲が異なる」ことを除いて、他の絵文字画像セットを用いる場合と同じ機能が提供される。和文周りの調整も従来通りに機能する。
- 絵文字出力時の内部処理の最終段階の「画像を `\includegraphics` で出力する」処理を「twemojis の `\texttwemoji` を実行する」処理（これも究極的には `\includegraphics` である）に置き換えている。
- twemojis オプションをパッケージオプションとして指定した場合は、twemojis パッケージが自動で読み込まれる。（ただし `graphicx` の自動読込の条件を満たす場合に限る。）これ以外の（例えば命令のオプションに指定する）場合は twemojis を別途読み込む必要がある。

^{*7} twemojis パッケージ（再掲）：<https://ctan.org/pkg/twemojis>