Emo: emoji for all (ahem, modern LaTeX engines)

Robert Grimm

Version v1.0 (2023/02/10)

Abstract

Emo implements the `\emo{`⟨*emoji-name*⟩`}` command for including color emoji such as 🏝 or 🦜 in your documents independent of input encoding or LaTeX engine. The implementation uses the Noto color emoji font if the engine supports it and includes PDF graphics otherwise. The latter are derived from Noto's SVG sources, so the visual appearance is very similar. The source repository is at https://github.com/apparebit/emo.

Contents

1 Installation

The Emo package comprises the `emo.dtx` file with all things LaTeX, Noto font files for Hebrew and Simplified Chinese, and the `emo-graphics` directory with PDF graphics. To manually install this package, first extract all embedded files by running `pdflatex emo.dtx` and then follow the generic install instructions for LaTeX to put the files into their places. In a pinch, your project directory will do just fine.

2 Usage

As usual, you declare your document's dependency on Emo with `\usepackage{emo}`. In addition to the unadorned form, Emo also supports two options:

extra Also define the `\lingchi` and `\YHWH` macros. They are described below.

index Create an emoji index tagged `emo` with the `.edx` extension for the raw index and the `.end` extension for the processed index.

Note that the `index` option declares the index with the `index` package and generates the `.edx` file. But it does not build the final index.

`\emo` `\emo{`⟨*emoji-name*⟩`}` expands to the named emoji. For LuaLaTeX, it uses the Noto color emoji font. For all other engines, it uses PDF graphics. That way, `\emo{desert-island}` results in 🏝 and `\emo{parrot}` results in 🦜.

Since LaTeX tends to produce a lot of command line noise about underfull boxes and loaded fonts, it's a bit too easy to miss meaningful warnings. For that reason, `\emo` expands to an attention-seeking error message upon undefined emoji names. For example, `\emo{boo}` produces Bad \emo{boo}.

`\lingchi` `\YHWH` The `\lingchi` and `\YHWH` macros take no arguments and produce 凌遲 and יהוה, respectively. They are only available if Emo is used with the `extra` option. The former is the Chinese term for "death by a thousand cuts." The latter is the Tetragrammaton, the Hebrew name for God. Observant Jews must not speak the name out loud. In my mind that nicely mirrors that we cannot know God. I am partial to both phrases and needed them when I was writing this package. So they are an extra for myself.

3 Emo Implementation

The two lines of the package declaration come well before documentation starts:

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{emo}[...]
```

```
1 ⟨*package⟩
```

3.1 Package Options

Declare Emo's `extra` and `index` options.

```
2 \newif\ifemo@mkextra\emo@mkextrafalse
3 \DeclareOption{extra}{\emo@mkextratrue}
4 \newif\ifemo@mkindex\emo@mkindexfalse
5 \DeclareOption{index}{\emo@mkindextrue}
6 \ProcessOptions\relax
```

3.2 Dependencies

The first dependency effectively declares this file's encoding to be UTF-8. `xelatex` and `lualatex` already expect files to be encoded that way and hence ignore the declaration. `pdflatex` supports other (legacy) encodings and hence needs to be told.

```
7 \RequirePackage[utf8]{inputenc}
```

This package also requires `xcolor` for formatting error messages and, depending on the engine, either `fontspec` or `graphicx` as emoji emitting backend. `iftex` helps keep those engines apart. Finally, `index` takes care of the emoji index — iff the `index` option has been provided.

Always including a package that is only used when there are errors is not ideal. I'd prefer to only load `xcolor` if needed, but that doesn't seem possible in LaTeX. It might be polite to add an option for disabling this feature.

```
8 \RequirePackage{xcolor}
9 \RequirePackage{iftex}
10 \ifluatex
11 \RequirePackage{fontspec}
```

```
\else
\RequirePackage{graphicx}
\fi

\ifemo@mkindex
\RequirePackage{index}
\fi
```

3.3 The Emoji Table

Next come a large number of macros named `\emo@emoji@`⟨*emoji-name*⟩. They define the table mapping emoji names to their Unicode character sequences. Since the table is (potentially) large and contains raw Unicode emoji (which trip up `pdflatex`), we elide the table from documentation.

3.4 Internal Macros

`emo@error@fg` `emo@error@bg` `emo@error` Define two colors and a function to format an attention-grabbing error message with those two colors. If you overlook a warning in the console, you *will* notice the error messsage in the document, thusly formatted.

```
\definecolor{emo@error@fg}{rgb}{1,1,1}
\definecolor{emo@error@bg}{rgb}{.6824,.0863,.0863}
\def\emo@error#1{%
    \colorbox{emo@error@bg}{%
        \textcolor{emo@error@fg}{%
            \textsf{Bad} \texttt{\textbackslash emo\{#1\}}%
        }%
    }%
}
```

`emo@ifdef` Validate the emoji name given as first argument. The macro expands to the second argument if the name is valid and an error message otherwise.

```
\def\emo@ifdef#1#2{%
    \ifcsname emo@emoji@#1\endcsname#2\else%
        \PackageWarning{emo}{Unknown emoji name in `\string\emo{#1}'}%
        \emo@error{#1}%
    \fi%
}
```

`emo@index` If indexing is enabled, record the use of an emoji. Otherwise, do nothing.

```
\ifemo@mkindex
\newindex{emo}{edx}{end}{Emoji Index}
\def\emo@index#1{\index[emo]{#1}}
\else
\def\emo@index#1{}
\fi
```

3.5 User Interface

`emo` Emit the named color emoji. Both the font-based `lualatex` and the graphics-based fallback version validate the emoji name and then invoke the `\emo@index` macro. The `lualatex` version next switches to the Noto color emoji font followed

by the named emoji sequence in Unicode from the `eme@emoji@` table, all within a group. The fallback version instead includes the named PDF graphic, suitably scaled.

```
\ifluatex
\newfontface\emo@font[Renderer=Harfbuzz]{NotoColorEmoji.ttf}
\newcommand\emo[1]{%
    \emo@ifdef{#1}{%
        \emo@index{#1}%
        {\emo@font\csname emo@emoji@#1\endcsname}%
    }%
}
\else
\newcommand\emo[1]{%
    \emo@ifdef{#1}{%
        \emo@index{#1}%
        \raisebox{-0.2ex}{\includegraphics[height=1em]{./emo-graphics/#1}}%
    }%
}
\fi
```

`lingchi` `YHWH` The definitions for the optional `\lingchi` and `\YHWH` macros follow from that of `\emo`, except that there is no argument to validate and that Hebrew requires switching text direction to right-to left with `\textdir TRT`. While it would be nice to use Unicode inside the explicit groups, doing so breaks the package documentation. So `\char` it is.

```
\ifemo@mkextra
\ifluatex
\newfontface\emo@chinese{NotoSansSC-Regular.otf}
\newfontface\emo@hebrew{NotoSerifHebrew-Regular.ttf}
\newcommand\lingchi{%
    \emo@index{lingchi}%
    \begingroup\emo@chinese \char"51CC\char"9072\endgroup%
}
\newcommand\YHWH{%
    \emo@index{YHWH}%
    \begingroup\textdir TRT\emo@hebrew \char"5D9\char"5D4\char"5D5\char"5D4\endgroup%
}
\else
\newcommand\lingchi{%
    \emo@index{lingchi}%
    \raisebox{-0.2ex}{\includegraphics[height=1em]{./emo-graphics/lingchi}}%
}
\newcommand\YHWH{%
    \emo@index{YHWH}%
    \raisebox{-0.2ex}{\includegraphics[height=1em]{./emo-graphics/YHWH}}%
}
\fi
\fi
```

Et voilà!

```
⟨/package⟩
```