Emo: emoji for all (LaTeX engines)

Robert Grimm

Version v0.1 (2023/03/07)

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 takes up to two options:

extra Also define the `\lingchi` and `\YHWH` macros, which produce 凌遲 and יהוה, respectively, and are documented below.

index Create an emoji index tagged `emo` with the `.edx` extension for the raw index and the `.end` extension for the processed index. This option relies on the `index` package, generates the raw `.edx` file, but does not use or build the processed 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 first command is the Chinese term for "death by a thousand cuts." While originally a Chinese execution method, the term applies to surprisingly many software systems as well. The second command is the Tetragrammaton, the Hebrew name for God. Observant Jews do not speak the name, not even in thoughts. In my mind, that nicely mirrors the very incomprehensibility of יהוה and so I started using the Tetragrammaton myself. Both macros preserve a subsequent space as space, no backslash needed.

3 Implementation

3.1 The Prelude

Two following two lines come early on in the `.dtx` file containing all sources for this package. After all, the version number should be visible at the head of the file. The package documentation starts quite a bit later, but has access to the version, **v0.1**, and date, **2023/03/07**.

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

The rest of the prelude comes next, declaring Emo's options and requiring its dependencies.

1 ⟨*prelude⟩

3.1.1 Package Options

Declare Emo's extra and index options.

```
\newif\ifemo@extra\emo@extrafalse
\DeclareOption{extra}{\emo@extratrue}
\newif\ifemo@indexing\emo@indexingfalse
\DeclareOption{index}{\emo@indexingtrue}
\ProcessOptions\relax
```

3.1.2 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.

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

This package also requires `xcolor` for formatting error messages and, depending on the engine, either `fontspec` or `graphicx` as the emoji-emitting backend. `iftex` helps keep those engines apart. 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, at least not with `\RequirePackage`. It might be polite to add an option for disabling this feature.

```
\RequirePackage{xcolor}
\RequirePackage{iftex}
\ifluatex
\RequirePackage{fontspec}
\else
\RequirePackage{graphicx}
\fi
```

Emo's options also have dependencies, with extra requiring the `xspace` package and index requiring the `index` package:

```
\ifemo@extra
\RequirePackage{xspace}
\fi
\ifemo@indexing
\RequirePackage{index}
\fi
```

That's it for the prelude.

```
⟨/prelude⟩
```

3.2 The Emoji Table

For each emoji with a PDF graphic in the `emo-graphics` directory, a macro named `\emo@emoji@`⟨*emoji-name*⟩ expands to its Unicode sequence. With over 3,000 distinct emoji in Unicode 15, Emo relies on a Python script for populating the graphics directory and writing the table to the `emo.table.dtx` file. In the

generated package file `emo.sty`, the contents of the table file come right after the prelude.

Since there are over 3,000 distinct emoji, the emoji table may be quite large. It definitely isn't very interesting or clever. Consequently, it is elided from the documentation.

The rest of the package follows after the table.

```
⟨*package⟩
```

3.3 Internal Macros

`emo@error@fg` `emo@error@bg` `emo@error` Define two colors and a function that uses the two colors for formatting an attention-grabbing error message. If you use an invalid emoji name and overlook the warning in the console, you *will* notice the error messsage in the document, Bad \emo{error}, 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. Its implementation relies on the `emo@emoji` table.

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

3.4 Public Macros

`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 (a) there are no arguments to validate and hence no equivalent to `\emo@ifdef`; (b) Hebrew is written right-to-left and hence `\YHWH` requires a `\textdir TRT`; (c) subsequent space should be preserved and hence the macros end with `\xspace`. While it would be nice to use Unicode inside the explicit groups, doing so breaks the package documentation. So `\char` it is.

```
\ifemo@extra
\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%
    \xspace}
\newcommand\YHWH{%
    \emo@index{YHWH}%
   \begingroup\textdir TRT\emo@hebrew \char"5D9\char"5D4\char"5D5\char"5D4\endgroup%
    \xspace}
\else
\newcommand\lingchi{%
    \emo@index{lingchi}%
    \raisebox{-0.2ex}{\includegraphics[height=1em]{./emo-graphics/lingchi}}%
    \xspace}
\newcommand\YHWH{%
    \emo@index{YHWH}%
    \raisebox{-0.2ex}{\includegraphics[height=1em]{./emo-graphics/YHWH}}%
    \xspace}
\fi
\fi
```

Et voilà!

83 ⟨/package⟩

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.