

# emo•ji for all (LaTeX engines)

Robert Grimm

Version v1.0 (2023/05/22)

## Abstract

Emo implements the `\emo{emoji-name}` command for including color emoji such as `\emo{desert-island}` for  or `\emo{parrot}` for  in your documents independent of LaTeX engine. The implementation uses the Noto color emoji font if the engine supports it and includes PDF graphics otherwise. It also supports conversion to HTML with either LaTeXML or TeX4ht. Next, PDF graphics are automatically derived from Noto's SVG sources, so the visual appearance is very similar. The source repository is at <https://github.com/apparebit/emo>. Emo may come in particularly handy when dealing with academic publishers that provide only minimal support for non-Latin scripts (cough, ACM, cough).

## Contents

<b>1</b>	<b>Installation</b>	<b>2</b>
<b>2</b>	<b>Usage</b>	<b>3</b>
2.1	One Main Macro . . . . .	3
2.1.1	Naming Scheme . . . . .	3
2.1.2	Inventory of Default Configuration . . . . .	5
2.2	Two Optional Macros . . . . .	19
2.3	Conversion to HTML . . . . .	20
<b>3</b>	<b>Configuration</b>	<b>20</b>
3.1	Running the Configuration Script . . . . .	20
<b>4</b>	<b>Copyright and Licensing</b>	<b>21</b>
<b>5</b>	<b>Implementation</b>	<b>21</b>
5.1	Package Options . . . . .	21
5.2	Package Dependencies and Backend Selection . . . . .	22
5.3	The Emoji Table . . . . .	23
5.4	Internal Macros . . . . .	24
5.5	Emo's Public Macros and Render Hooks . . . . .	24
5.5.1	A Skeleton of Render Hooks . . . . .	25
5.5.2	The User Macros . . . . .	26
5.5.3	Activating the Hooks . . . . .	26
5.5.4	Background: Variability vs Code Duplication . . . . .	27

<b>6</b>	<b>LaTeXML Binding</b>	<b>28</b>
<b>7</b>	<b>Testing and Documenting Emo</b>	<b>30</b>
7.1	Emo's Support Package . . . . .	30
7.1.1	Report Generation . . . . .	31
7.1.2	Emoji Inventory . . . . .	32
7.1.3	Testing . . . . .	33
7.2	Engine Test and Report . . . . .	35
7.3	Simple Test Document . . . . .	35

## 1 Installation

The `emo` package is available through its [source repository](#) or through [CTAN](#). Installation is fairly straightforward, though it does involve a lot more files than usual.

1. Start by extracting this package's files from `emo.dtx` by running:

```
$ pdflatex emo.dtx
```

Do *not* use `tex`; it mangles the embedded `README.md`. `pdflatex` also extracts the files and then builds the documentation. Embedded files are `build.sh`, `emo.ins`, `emo.sty`, `emo.sty.ltxml`, `emo-support.sty`, `canary.tex`, `demo.tex`, and `README.md`. Extraction will overwrite existing files with the same name without asking.

2. Test the package and build its documentation by making `build.sh` executable and then running it:

```
$ chmod +x build.sh; ./build.sh
```

The shell script does three things: First, it tests `emo` by compiling `canary.tex` with pdfTeX, XeTeX, as well as LuaTeX and generating `canary.pdf` with the results from the three tests. Second, it tests `emo` by compiling `demo.tex` with LaTeXML as well as TeX4ht and generating `demo-latexml.html` as well as `demo-tex4ht.html`, which are framed together by `demo.html`. Third, it builds the package documentation in `emo.pdf` with all bells and indices.

3. Get started reconfiguring supported emoji by running:

```
$ python config/emo.py -h
```

For more detailed instructions, see §3 below.

4. Put the following files somewhere LaTeX can find them. In a pinch, your current project's directory will do. However, `emo`'s installation potentially comprises thousands of files. So, you probably want to use a dedicated directory and add that to the search path for LaTeX, e.g., by setting the `TEXINPUTS` environment variable.

- (a) `emo.sty` with the package implementation;
- (b) `emo.sty.ltxml` with the binding for [LaTeXML](#);
- (c) `emo.def` with the emoji table;
- (d) `emo-lingchi.ttf` with the two glyphs for `\lingchi`;
- (e) `emo-graphics` with the fallback PDF graphics.

TeX Live requires that each package's files have unique names. For that reason, the PDF graphics in the `emo-graphics` directory start with the `emo-` prefix as well.

When running on the LuaLaTeX engine, the `emo` package also uses the Noto color emoji (`NotoColorEmoji.ttf`) and Linux Libertine (`LinLibertine_R.otf`) fonts, with the latter used for rendering `\YHWH` only. Neither file is included with `emo`'s distribution, since both of them are distributed with major TeX distributions already. If they are not included with your LaTeX distribution, you can find them on CTAN. The `emo-lingchi.ttf` font distributed with `emo` is a two glyph subset of `Noto-Serif--Regular.otf`, i.e., the traditional Chinese version of Noto serif.

## 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 `𠙴`.

**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 build or use the processed index.

### 2.1 One Main Macro

`\emo` An `\emo{\<emoji-name>}` invocation 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 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}`.

#### 2.1.1 Naming Scheme

With a few exceptions, `emo`'s names for emoji are automatically derived from their Unicode names, with letters converted to lowercase, punctuation such as commas, colons, quotes, and parentheses stripped, and interword spaces replaced by dashes. Furthermore, instead of the rather verbose dark-skin-tone, medium-dark-skin-tone, etc modifiers, `emo` uses the more succinct darkest, darker, medium, lighter, and lightest.

For some names, `emo` goes further by hard-coding shorter names. Those names are listed in Table 1.

`Emo`'s `emo.def` contains the names and codepoints of all currently supported emoji in Unicode display order. `Emo`'s distribution also includes the `emoji-test.txt` file, which is part of [Unicode TR-51](#) and contains the names and codepoints of all *potentially supported* emoji, i.e., all emoji, also in Unicode display order. It further organizes emoji into groups and subgroups, with the current (sub)group being the one named on the closest line above the emoji that starts with `# (sub)group:`. As described in the next section, the group and subgroup names can be used during configuration for concisely naming a large number of emoji.

Table 1: Exceptional emoji names

Transformed Unicode Name	Emo Replacement Name
a-button-blood-type	a-button
ab-button-blood-type	ab-button
b-button-blood-type	b-button
o-button-blood-type	o-button
bust-in-silhouette	bust
busts-in-silhouette	busts
flag-european-union	eu
globe-showing-americas	globe-americas
globe-showing-asia-australia	globe-asia-australia
globe-showing-europe-africa	globe-africa-europe
hear-no-evil-monkey	hear-no-evil
index-pointing-at-the-viewer	index-pointing-at-viewer
index-pointing-at-the-viewer-darkest	index-pointing-at-viewer-darkest
index-pointing-at-the-viewer-darker	index-pointing-at-viewer-darker
index-pointing-at-the-viewer-medium	index-pointing-at-viewer-medium
index-pointing-at-the-viewer-lighter	index-pointing-at-viewer-lighter
index-pointing-at-the-viewer-lightest	index-pointing-at-viewer-lightest
keycap-*	keycap-star
keycap-#	keycap-hash
keycap-0	keycap-zero
keycap-1	keycap-one
keycap-2	keycap-two
keycap-3	keycap-three
keycap-4	keycap-four
keycap-5	keycap-five
keycap-6	keycap-six
keycap-7	keycap-seven
keycap-8	keycap-eight
keycap-9	keycap-nine
keycap-10	keycap-ten
magnifying-glass-tilted-left	loupe-left
magnifying-glass-tilted-right	loupe-right
palm-down-hand	palm-down
palm-down-hand-darkest	palm-down-darkest
palm-down-hand-darker	palm-down-darker
palm-down-hand-medium	palm-down-medium
palm-down-hand-lighter	palm-down-lighter
palm-down-hand-lightest	palm-down-lightest
palm-up-hand	palm-up
palm-up-hand-darkest	palm-up-darkest
palm-up-hand-darker	palm-up-darker
palm-up-hand-medium	palm-up-medium
palm-up-hand-lighter	palm-up-lighter
palm-up-hand-lightest	palm-up-lightest
rolling-on-the-floor-laughing	rofl
see-no-evil-monkey	see-no-evil
speak-no-evil-monkey	speak-no-evil

### 2.1.2 Inventory of Default Configuration

Emo's default configuration includes all emoji *but* those in the *people-and-body* Unicode group that modify gender, hair, or skin and those in the *country-flag* subgroup of the *flags* group other than the EU flag (which is included). As of Unicode 15.0, that's 1,415 out of 3,655 emoji. There are another 257 flags, but the majority of excluded emoji are variations on already included emoji depicting people or body parts. As described in the next section, enabling additional emoji is as easy as running a Python script with the emoji names as arguments.

With their names listed in Unicode display order and organized by Unicode group and subgroup, that's the following emoji.

<i>smileys-and-emotion</i>		
<hr/>		
<b>FACE-SMILING</b>		
grinning-face .....	😊	star-struck..... 😍
grinning-face-with-big-eyes .....	😊	face-blowing-a-kiss .....
grinning-face-with-smiling-eyes .....	😊	kissing-face .....
beaming-face-with-smiling-eyes .....	😊	smiling-face .....
grinning-squinting-face .....	😊	kissing-face-with-closed-eyes .....
grinning-face-with-sweat .....	😊	kissing-face-with-smiling-eyes .....
rofl .....	🤣	smiling-face-with-tear .....
face-with-tears-of-joy .....	😂	<hr/>
slightly-smiling-face .....	😊	<b>FACE-TONGUE</b>
upside-down-face .....	🙃	face-savoring-food ... 😋
melting-face .....	👻	face-with-tongue .....
winking-face .....	😉	winking-face-with-tongue .....
smiling-face-with-smiling-eyes .....	😊	zany-face .....
smiling-face-with-halo .....	😇	squinting-face-with-tongue .....
<hr/>		money-mouth-face .....
<b>FACE-AFFECTION</b>		<hr/>
smiling-face-with-hearts .....	😍	smiling-face-with-open-hands .....
smiling-face-with-heart-eyes .....	😍	face-with-hand-over-mouth .....
<hr/>		<b>FACE-SLEEPY</b>
face-with-open-eyes-and-hand-over-mouth .....	👀	relieved-face .....
face-with-peeking-eye .....	👀	pensive-face .....
shushing-face .....	쉿	sleepy-face .....
		drooling-face .....
		sleeping-face .....

<hr/>		
FACE-UNWELL	astonished-face ..... 😲	
face-with-medical-mask .....	flushed-face ..... 🥱	
face-with-thermometer.....	pleading-face ..... 🥺	
face-with-head-bandage .....	face-holding-back-tears ..... 😢	
nauseated-face.....	frowning-face-with-open-mouth ..... 😦	
face-vomiting .....	anguished-face..... 😧	
sneezing-face .....	fearful-face ..... 😨	
hot-face .....	anxious-face-with-sweat ..... 😰	
cold-face.....	sad-but-relieved-face ..... 😢	
woozy-face .....	crying-face..... 😢	
face-with-crossed-out-eyes .....	loudly-crying-face ... 😭	
face-with-spiral-eyes .....	face-screaming-in-fear ..... 😱	
exploding-head.....	confounded-face .....	
<hr/>		
FACE-HAT	persevering-face .....	
cowboy-hat-face .....	disappointed-face.... 😞	
partying-face .....	downcast-face-with-sweat .....	
disguised-face.....	weary-face .....	
<hr/>		
FACE-GLASSES	tired-face .....	
smiling-face-with-sunglasses .....	yawning-face .....	
nerd-face.....	<hr/>	
face-with-monocle....	FACE-NEGATIVE	
<hr/>		
FACE-CONCERNED	face-with-steam-from-nose .....	
confused-face .....	enraged-face .....	
face-with-diagonal-mouth .....	angry-face .....	
worried-face.....	face-with-symbols-on-mouth .....	
slightly-frowning-face .....	smiling-face-with-horns .....	
frowning-face .....	angry-face-with-horns .....	
face-with-open-mouth .....	skull .....	
hushed-face.....	skull-and-crossbones .....	
<hr/>		
FACE-COSTUME	pile-of-poo..... 💩	
clown-face .....	clown-face .....	
<hr/>		
ogre .....	ghost .....	
goblin.....	alien .....	
alien-monster .....	robot .....	
<hr/>		
CAT-FACE	grinning-cat .....	
grinning-cat-with-smiling-eyes .....	cat-with-tears-of-joy..... 😊	
cat-with-wry-smile...	smiling-cat-with-heart-eyes .....	
kissing-cat.....	weary-cat .....	
weary-cat .....	crying-cat .....	
crying-cat .....	pouting-cat..... 😷	
<hr/>		
MONKEY-FACE	see-no-evil..... 🙄	
hear-no-evil .....	speak-no-evil .....	
<hr/>		
HEART	love-letter .....	
heart-with-arrow .....	heart-with-ribbon..... 🎀	
sparkling-heart .....	growing-heart .....	
beating-heart .....	revolving-hearts .....	
two-hearts .....	heart-decoration .....	
broken-heart .....	heart-exclamation.... 😍	
heart-on-fire .....	mending-heart .....	
red-heart .....	red-heart .....	
<hr/>		

orange-heart .....	♥	leftwards-pushing-hand .....	👉
yellow-heart .....	💛	rightwards-pushing-hand .....	👉
green-heart .....	💚		
blue-heart .....	💙		
light-blue-heart .....	옅️		
purple-heart .....	💜		
brown-heart .....	🤎		
black-heart .....	🖤		
grey-heart .....	🌚		
white-heart .....	🤍		
<hr/>			
EMOTION			
kiss-mark .....	💋	crossed-fingers .....	👉
hundred-points .....	💯	hand-with-index-finger-and-thumb-crossed .....	👉
anger-symbol .....	💢	love-you-gesture .....	👉
collision .....	💥	sign-of-the-horns .....	👉
dizzy .....	💫	call-me-hand .....	👉
sweat-droplets .....	💦		
dashing-away .....	💨		
hole .....	🕳		
speech-balloon .....	💬		
eye-in-speech-bubble .....	👁️		
left-speech-bubble .....	🗨️		
right-anger-bubble .....	🗯️		
thought-balloon .....	💭		
zzz .....	💤		
<hr/>			
<b>people-and-body</b>			
<hr/>			
HAND-FINGERS-OPEN			
waving-hand .....	👋	index-pointing-up .....	👉
raised-back-of-hand .....	✋	index-pointing-at-viewer .....	👉
hand-with-fingers-splayed .....	🖐		
raised-hand .....	✋		
vulcan-salute .....	🖖		
rightwards-hand .....	👉		
leftwards-hand .....	👈		
palm-down .....	👉		
palm-up .....	👉		
<hr/>			
HAND-FINGERS-CLOSED			
thumbs-up .....	👍	raised-fist .....	✊
thumbs-down .....	👎	oncoming-fist .....	👉
		left-facing-fist .....	👉
		right-facing-fist .....	👉
<hr/>			
HANDS			
clapping-hands .....	👏		
raising-hands .....	🙌		
heart-hands .....	❤️		
open-hands .....	👐		
palms-up-together .....	🤲		
handshake .....	🤝		
folded-hands .....	🙏		
<hr/>			
HAND-PROP			
writing-hand .....	✍		
nail-polish .....	💅		
selfie .....	🤳		
<hr/>			
BODY-PARTS			
flexed-biceps .....	💪		
mechanical-arm .....	🦾		
mechanical-leg .....	🦵		
leg .....	🦵		
foot .....	🦶		
ear .....	👂		
ear-with-hearing-aid .....	👂		
nose .....	👃		
brain .....	🧠		
anatomical-heart .....	心脏病		
lungs .....	肺		
tooth .....	🦷		
bone .....	🦱		
eyes .....	👀		
eye .....	👁️		
tongue .....	👅		
mouth .....	👄		
biting-lip .....	👄		
<hr/>			
PERSON			
baby .....	👶		
child .....	🧒		
boy .....	👦		
girl .....	👧		

person	...	👤	construction-	...	👤	person-in-motorized-	...
person-blond-hair	...	👱	worker	...	👷	wheelchair	...
person-beard	...	🧔	person-with-crown	...	🤴	person-in-manual-	...
older-person	...	👵	prince	...	🤴	wheelchair	...
<hr/>							
<b>PERSON-GESTURE</b>							
person-frowning	...	🙍	person-with-	...	👤	person-in-suit-	...
person-pouting	...	🙎	skullcap	...	👳	levitating	...
person-gesturing-	...	手势	person-in-tuxedo	...	🤵	people-with-bunny-	...
no	...	🚫	person-with-veil	...	👰	ears	...
person-gesturing-	...	手势	pregnant-person	...	🤰	person-in-steamy-	...
ok	...	👌	breast-feeding	...	🤱	room	...
person-tipping-	...	👉	person-feeding-	...	👶	person-climbing	...
hand	...	👉	baby	...	👶	<hr/>	
person-raising-	...	👉	<b>PERSON-FANTASY</b>				
hand	...	👉	baby-angel	...	👼	person-fencing	...
deaf-person	...	聋哑人	santa-claus	...	🎅	horse-racing	...
person-bowing	...	🙇	mrs-claus	...	🤶	skier	...
person-facepalming	...	🤦	mx-claus	...	🤶	snowboarder	...
person-shrugging	...	🤷	superhero	...	🦸	person-golfing	...
<hr/>							
<b>PERSON-ROLE</b>							
health-worker	...	👤	supervillain	...	😈	person-surfing	...
student	...	👤	mage	...	🧙	person-rowing-boat	...
teacher	...	👤	fairy	...	🧚	person-swimming	...
judge	...	👤	vampire	...	🧛	person-bouncing-	...
farmer	...	👤	merperson	...	🧜	ball	...
cook	...	👤	merman	...	🧜	person-lifting-	...
mechanic	...	👤	mermaid	...	🧜	weights	...
factory-worker	...	👤	elf	...	🧚	person-biking	...
office-worker	...	👤	genie	...	🧞	person-mountain-	...
scientist	...	👤	zombie	...	🧟	biking	...
technologist	...	👤	troll	...	🧟	person-	...
<hr/>							
<b>PERSON-ACTIVITY</b>							
singer	...	🎤	person-getting-	...	👤	cartwheeling	...
artist	...	🎨	massage	...	👤	people-wrestling	...
pilot	...	👤	person-getting-	...	👤	person-playing-water-	...
astronaut	...	👤	haircut	...	👤	polo	...
firefighter	...	🚒	person-walking	...	🚶	person-playing-	...
police-officer	...	👮	person-standing	...	🚶	handball	...
detective	...	🕵	person-kneeling	...	🚶	person-juggling	...
guard	...	💂	person-with-white-	...	🚶	<hr/>	
ninja	...	🥋	cane	...	🚶	<b>PERSON-RESTING</b>	
						person-in-lotus-	...
						position	...
						person-taking-bath	...
						person-in-bed	...

<hr/> <b>FAMILY</b> <hr/>		unicorn .....		<hr/> <b>ANIMAL-BIRD</b> <hr/>	
people-holding-hands .....		zebra .....		turkey.....	
kiss .....		deer .....		chicken .....	
couple-with-heart .....		bison .....		rooster .....	
family.....		cow-face .....		hatching-chick.....	
<hr/> <b>PERSON-SYMBOL</b> <hr/>		ox .....		baby-chick .....	
speaking-head .....		water-buffalo .....		front-facing-baby-chick .....	
bust .....		cow.....		bird.....	
busts .....		pig-face .....		penguin .....	
people-hugging.....		pig.....		dove .....	
footprints .....		boar .....		eagle .....	
<hr/> <b>animals-and-nature</b> <hr/>		pig-nose .....		duck .....	
<hr/> <b>ANIMAL-MAMMAL</b> <hr/>		ram.....		swan .....	
monkey-face.....		ewe.....		owl.....	
monkey.....		goat .....		dodo .....	
gorilla .....		camel .....		feather .....	
orangutan.....		two-hump-camel.....		flamingo .....	
dog-face .....		llama .....		peacock .....	
dog.....		giraffe .....		parrot.....	
guide-dog.....		elephant .....		wing .....	
service-dog.....		mammoth .....		black-bird .....	
poodle.....		rat.....		goose .....	
wolf .....		hamster .....		<hr/> <b>ANIMAL-AMPHIBIAN</b> <hr/>	
fox.....		rabbit-face.....		frog .....	
raccoon .....		rabbit.....		<hr/> <b>ANIMAL-REPTILE</b> <hr/>	
cat-face .....		chipmunk .....		crocodile.....	
cat.....		beaver.....		turtle.....	
black-cat.....		hedgehog .....		lizard.....	
lion .....		bat.....		snake .....	
tiger-face .....		bear .....		dragon-face.....	
tiger .....		polar-bear .....		dragon.....	
leopard .....		koala .....		sauropod .....	
horse-face .....		panda .....		t-rex .....	
moose .....		sloth .....			
donkey.....		otter .....			
horse .....		skunk .....			
		kangaroo .....			
		badger .....			
		paw-prints .....			

ANIMAL-MARINE		PLANT-OTHER		FOOD-VEGETABLE	
spouting-whale.....		sunflower.....		kiwi-fruit .....	
whale .....		blossom .....		tomato.....	
dolphin .....		tulip .....		olive .....	
seal .....		hyacinth .....		coconut .....	
ANIMAL-BUG		FOOD-AND-DRINK		FOOD-PREPARED	
snail .....		seedling .....		avocado .....	
butterfly.....		potted-plant .....		eggplant .....	
bug.....		evergreen-tree.....		potato.....	
ant.....		deciduous-tree.....		carrot.....	
honeybee .....		palm-tree.....		ear-of-corn.....	
beetle.....		cactus.....		hot-pepper .....	
lady-beetle.....		sheaf-of-rice .....		bell-pepper.....	
cricket .....		herb.....		cucumber .....	
cockroach.....		shamrock .....		leafy-green.....	
spider .....		four-leaf-clover .....		broccoli .....	
spider-web .....		maple-leaf .....		garlic.....	
scorpion .....		fallen-leaf.....		onion .....	
mosquito .....		leaf-fluttering-in-wind.....		peanuts .....	
fly.....		empty-nest .....		beans .....	
worm.....		nest-with-eggs.....		chestnut .....	
microbe .....		mushroom .....		ginger-root.....	
				pea-pod .....	
PLANT-FLOWER		FOOD-FRUIT		FOOD-AND-DRINK	
bouquet .....		grapes.....		bread .....	
cherry-blossom.....		melon .....		croissant .....	
white-flower .....		watermelon .....		baguette-bread.....	
lotus .....		tangerine.....		flatbread.....	
rosette .....		lemon .....		pretzel .....	
rose .....		banana.....		bagel .....	
wilted-flower .....		pineapple.....		pancakes .....	
hibiscus .....		mango .....		waffle .....	
		red-apple.....		cheese-wedge .....	
		green-apple.....		meat-on-bone .....	
		pear .....		poultry-leg.....	
		peach .....		cut-of-meat.....	
		cherries .....		bacon .....	
		strawberry .....		hamburger .....	
		blueberries.....		french-fries .....	
				pizza .....	
				hot-dog .....	
				sandwich .....	

taco .....		squid .....			
burrito .....		oyster .....			
tamale .....					
stuffed-flatbread.....					
falafel .....					
egg.....		FOOD-SWEET			
cooking .....		soft-ice-cream.....			
shallow-pan-of-food .....		shaved-ice .....			
pot-of-food.....		ice-cream.....			
fondue.....		doughnut .....			
bowl-with-spoon .....		cookie.....			
green-salad.....		birthday-cake .....			
popcorn .....		shortcake.....			
butter.....		cupcake .....			
salt .....		pie.....			
canned-food.....		chocolate-bar .....			
		candy .....			
		lollipop .....			
FOOD-ASIAN		custard .....			
		honey-pot .....			
bento-box.....					
rice-cracker.....					
rice-ball.....		DRINK			
cooked-rice.....		baby-bottle.....			
curry-rice .....		glass-of-milk .....			
steaming-bowl .....		hot-beverage .....			
spaghetti.....		teapot.....			
roasted-sweet-potato.....		teacup-without-handle .....			
oden .....		sake .....			
sushi .....		bottle-with-popping-cork .....			
fried-shrimp.....		wine-glass .....			
fish-cake-with-swirl .....		cocktail-glass.....			
moon-cake.....		tropical-drink.....			
dango .....		beer-mug .....			
dumpling .....		clinking-beer-mugs .....			
fortune-cookie.....		clinking-glasses .....			
takeout-box.....		tumbler-glass .....			
		pouring-liquid.....			
FOOD-MARINE		cup-with-straw.....			
		bubble-tea .....			
crab .....		beverage-box .....			
lobster .....		mate .....			
shrimp.....		ice.....			
			DISHWARE		
			chopsticks .....		
			fork-and-knife-with-plate .....		
			fork-and-knife.....		
			spoon .....		
			kitchen-knife .....		
			jar.....		
			amphora .....		
				<b><i>travel-and-places</i></b>	
				PLACE-MAP	
				globe-africa-europe.....	
				globe-americas.....	
				globe-asia-australia.....	
				globe-with-meridians .....	
				world-map .....	
				map-of-japan .....	
				compass .....	
				PLACE-GEOGRAPHIC	
				snow-capped-mountain .....	
				mountain .....	
				volcano .....	
				mount-fuji .....	
				camping .....	
				beach-with-umbrella .....	
				desert .....	
				desert-island .....	
				national-park .....	

<hr/>							
PLACE-BUILDING	foggy .....  sport-utility- night-with-stars .....  vehicle .....   cityscape .....  pickup-truck .....   sunrise-over- building- construction .....  delivery-truck .....   mountains .....  articulated-lorry .....   sunrise .....  tractor .....   cityscape-at-dusk .....  racing-car .....   sunset .....  motorcycle .....   bridge-at-night .....  motor-scooter .....   hot-springs .....  manual-wheelchair .....   carousel-horse .....  motorized- wheelchair .....   playground-slide .....  wheelchair .....   ferris-wheel .....  auto-rickshaw .....   roller-coaster .....  bicycle .....   barber-pole .....  kick-scooter .....   circus-tent .....  skateboard .....  <hr/> <td>TRANSPORT-GROUND</td> <td>roller-skate .....  bus-stop .....  motorway .....  railway-car .....  railway-track .....  high-speed-train .....  oil-drum .....  bullet-train .....  fuel-pump .....  train .....  wheel .....  metro .....  police-car-light .....  light-rail .....  horizontal-traffic- station .....  tram .....  vertical-traffic- light .....  monorail .....  stop-sign .....  mountain-railway .....  tram-car .....  construction .....  <hr/> <td>PLACE-RELIGIOUS</td><td>bus .....  oncoming-bus .....  trolleybus .....  minibus .....  ambulance .....  anchor .....  fire-engine .....  ring-buoy .....  police-car .....  sailboat .....  oncoming-police- car .....  canoe .....  taxi .....  speedboat .....  oncoming-taxi .....  passenger-ship .....  automobile .....  ferry .....  oncoming- automobile .....  motor-boat .....  <hr/> <td>PLACE-OTHER</td><td>ship .....  <hr/></td></td></td>	TRANSPORT-GROUND	roller-skate .....  bus-stop .....  motorway .....  railway-car .....  railway-track .....  high-speed-train .....  oil-drum .....  bullet-train .....  fuel-pump .....  train .....  wheel .....  metro .....  police-car-light .....  light-rail .....  horizontal-traffic- station .....  tram .....  vertical-traffic- light .....  monorail .....  stop-sign .....  mountain-railway .....  tram-car .....  construction .....  <hr/> <td>PLACE-RELIGIOUS</td> <td>bus .....  oncoming-bus .....  trolleybus .....  minibus .....  ambulance .....  anchor .....  fire-engine .....  ring-buoy .....  police-car .....  sailboat .....  oncoming-police- car .....  canoe .....  taxi .....  speedboat .....  oncoming-taxi .....  passenger-ship .....  automobile .....  ferry .....  oncoming- automobile .....  motor-boat .....  <hr/> <td>PLACE-OTHER</td><td>ship .....  <hr/></td></td>	PLACE-RELIGIOUS	bus .....  oncoming-bus .....  trolleybus .....  minibus .....  ambulance .....  anchor .....  fire-engine .....  ring-buoy .....  police-car .....  sailboat .....  oncoming-police- car .....  canoe .....  taxi .....  speedboat .....  oncoming-taxi .....  passenger-ship .....  automobile .....  ferry .....  oncoming- automobile .....  motor-boat .....  <hr/> <td>PLACE-OTHER</td> <td>ship .....  <hr/></td>	PLACE-OTHER	ship .....  <hr/>
fountain ..... 							
tent ..... 							
<hr/>							

<b>TRANSPORT-AIR</b>	seven-oclock .....	⌚	sun-behind-large-cloud .....	☁️	
airplane .....	✈️	seven-thirty .....	⌚	sun-behind-rain-cloud .....	🌧️
small-airplane.....	✈️	eight-oclock.....	⌚	cloud-with-rain .....	🌧️
airplane-departure ...	✈️	eight-thirty.....	⌚	cloud-with-snow .....	🌨️
airplane-arrival .....	✈️	nine-oclock.....	⌚	cloud-with-lightning .....	🌩️
parachute.....	傘	nine-thirty.....	⌚	tornado .....	🌪️
seat .....	💺	ten-oclock .....	⌚	fog .....	🌫️
helicopter .....	🚁	ten-thirty .....	⌚	wind-face .....	🌬️
suspension-railway ...	🚤	eleven-oclock .....	⌚	cyclone .....	🌀
mountain-cableway.....	🚤	eleven-thirty .....	⌚	rainbow .....	🌈
aerial-tramway.....	🚤			closed-umbrella .....	🌂
satellite.....	📡			umbrella .....	☂️
rocket.....	🚀	new-moon .....	🌙	umbrella-with-rain-drops .....	☂️
flying-saucer .....	🛸	waxing-crescent-moon .....	🌙	umbrella-on-ground .....	🏖️
		first-quarter-moon ...	🌓	high-voltage .....	⚡
		waxing-gibbous-moon .....	🌔	snowflake .....	❄️
		full-moon.....	🌕	snowman .....	⛄️
		waning-gibbous-moon .....	🌖	snowman-without-snow .....	⛄️
		last-quarter-moon....	🌗	comet .....	☄️
		waning-crescent-moon .....	🌙	fire .....	🔥
		crescent-moon .....	🌙	droplet .....	💧
		new-moon-face .....	🌙	water-wave .....	🌊
<b>TIME</b>					
hourglass-done.....	⌛️				
hourglass-not-done ...	⌛️				
watch .....	⌚️				
alarm-clock.....	⏰				
stopwatch.....	⌚️				
timer-clock.....	⌚️				
mantelpiece-clock....	🕰️				
twelve-oclock .....	🕒				
twelve-thirty .....	🕒				
one-oclock .....	🕒				
one-thirty .....	🕒				
two-oclock .....	🕒				
two-thirty .....	🕒				
three-oclock.....	🕒				
three-thirty.....	🕒				
four-oclock.....	🕒				
four-thirty.....	🕒				
five-oclock.....	🕒				
five-thirty.....	🕒				
six-oclock .....	🕒				
six-thirty .....	🕒				
		sun-behind-cloud .....	☀️		
		cloud-with-lightning-and-rain .....	🌧️		
		sun-behind-small-cloud .....	☀️		

## activities

<b>EVENT</b>			
jack-o-lantern.....	🎃		
christmas-tree.....	🎄		
fireworks.....	🎆		
sparkler .....	🎇		
firecracker.....	🧨		
sparkles .....	✨		
balloon .....	🎈		
party-popper .....	🎉		
confetti-ball .....	🎊		
tanabata-tree .....	🎋		
pine-decoration .....	🎍		
japanese-dolls.....	🎎		
carp-streamer .....	🎏		

<b>objects</b>	
<hr/>	
wind-chime .....	
moon-viewing-ceremony .....	
red-envelope .....	
ribbon .....	
wrapped-gift .....	
reminder-ribbon .....	
admission-tickets .....	
ticket .....	
<hr/> <b>AWARD-MEDAL</b>	
military-medal .....	
trophy .....	
sports-medal .....	
1st-place-medal .....	
2nd-place-medal .....	
3rd-place-medal .....	
<hr/> <b>SPORT</b>	
soccer-ball .....	
baseball .....	
softball .....	
basketball .....	
volleyball .....	
american-football .....	
rugby-football .....	
tennis .....	
flying-disc .....	
bowling .....	
cricket-game .....	
field-hockey .....	
ice-hockey .....	
lacrosse .....	
ping-pong .....	
badminton .....	
boxing-glove .....	
martial-arts-uniform .....	
goal-net .....	
flag-in-hole .....	
ice-skate .....	
fishing-pole .....	
diving-mask .....	
running-shirt .....	
skis .....	
sled .....	
curling-stone .....	
<hr/> <b>CLOTHING</b>	
glasses .....	
sunglasses .....	
goggles .....	
lab-coat .....	
safety-vest .....	
necktie .....	
t-shirt .....	
jeans .....	
scarf .....	
gloves .....	
coat .....	
socks .....	
dress .....	
kimono .....	
sari .....	
one-piece-swimsuit ...	
briefs .....	
shorts .....	
bikini .....	
womans-clothes .....	
folding-hand-fan .....	
purse .....	
handbag .....	
clutch-bag .....	
shopping-bags .....	
backpack .....	
thong-sandal .....	
mans-shoe .....	
running-shoe .....	
hiking-boot .....	
flat-shoe .....	
high-heeled-shoe .....	
womans-sandal .....	
ballet-shoes .....	
womans-boot .....	
hair-pick .....	
crown .....	
womans-hat .....	
top-hat .....	
graduation-cap .....	
billed-cap .....	
<hr/> <b>ARTS-AND-CRAFTS</b>	
performing-arts .....	
framed-picture .....	
artist-palette .....	
thread .....	
sewing-needle .....	
yarn .....	
knot .....	

military-helmet .....		maracas .....		light-bulb .....	
rescue-workers-helmet .....		flute .....		flashlight .....	
prayer-beads .....		PHONE		red-paper-lantern .....	
lipstick .....				diya-lamp .....	
ring .....		mobile-phone .....		BOOK-PAPER	
gem-stone .....		mobile-phone-with-arrow .....			
SOUND		telephone .....		notebook-with-decorative-cover .....	
muted-speaker .....		telephone-receiver .....		closed-book .....	
speaker-low-volume .....		pager .....		open-book .....	
speaker-medium-volume .....		fax-machine .....		green-book .....	
speaker-high-volume .....		COMPUTER		blue-book .....	
loudspeaker .....		battery .....		orange-book .....	
megaphone .....		low-battery .....		books .....	
postal-horn .....		electric-plug .....		notebook .....	
bell .....		laptop .....		ledger .....	
bell-with-slash .....		desktop-computer .....		page-with-curl .....	
MUSIC		printer .....		scroll .....	
musical-score .....		keyboard .....		page-facing-up .....	
musical-note .....		computer-mouse .....		newspaper .....	
musical-notes .....		trackball .....		rolled-up-newspaper .....	
studio-microphone .....		computer-disk .....		bookmark-tabs .....	
level-slider .....		floppy-disk .....		bookmark .....	
control-knobs .....		optical-disk .....		label .....	
microphone .....		LIGHT-AND-VIDEO		MONEY	
headphone .....		radio .....		movie-camera .....	
MUSICAL-INSTRUMENT				money-bag .....	
saxophone .....				coin .....	
accordion .....				yen-banknote .....	
guitar .....				dollar-banknote .....	
musical-keyboard .....				euro-banknote .....	
trumpet .....				pound-banknote .....	
violin .....				money-with-wings .....	
banjo .....				credit-card .....	
drum .....				receipt .....	
long-drum .....				chart-increasing-with-candle .....	
				yen .....	

MAIL	
envelope .....	✉
e-mail.....	✉
incoming-envelope....	✉
envelope-with-arrow .....	✉
outbox-tray.....	📦
inbox-tray .....	📦
package .....	📦
closed-mailbox-with-raised-flag.....	📫
closed-mailbox-with-lowered-flag.....	📫
open-mailbox-with-raised-flag.....	📫
open-mailbox-with-lowered-flag.....	📫
postbox .....	📮
ballot-box-with-ballot.....	🗳️
WRITING	
pencil.....	-pencil
black-nib.....	-pen
fountain-pen.....	-pen
pen.....	-pen
paintbrush .....	-brush
crayon.....	-crayon
memo .....	-memo
OFFICE	
briefcase.....	💼
file-folder.....	📁
open-file-folder .....	📁
card-index-dividers .....	📁
calendar .....	📅
tear-off-calendar.....	📅
spiral-notepad.....	📝
spiral-calendar.....	📅
card-index .....	📝
chart-increasing .....	📈
chart-decreasing .....	
bar-chart.....	📊
clipboard.....	📋
pushpin .....	📌
round-pushpin .....	📌
paperclip.....	📎
linked-paperclips.....	📎
straight-ruler.....	📏
triangular-ruler .....	📐
scissors .....	✂️
card-file-box .....	📁
file-cabinet.....	🗄️
wastebasket.....	🗑️
LOCK	
locked.....	🔒
unlocked .....	🔓
locked-with-pen .....	🔒
locked-with-key .....	🔒
key.....	🔑
old-key .....	🔑
TOOL	
hammer.....	🔨
axe.....	斧
pick.....	⛏️
hammer-and-pick .....	⚒️
hammer-and-wrench.....	🔧
dagger.....	🗡️
crossed-swords.....	⚔️
bomb .....	💣
boomerang.....	🎯
bow-and-arrow .....	🏹
shield.....	🛡️
carpentry-saw .....	锯
wrench.....	扳手
screwdriver.....	螺丝刀
nut-and-bolt.....	螺栓
gear .....	齿轮
clamp .....	夹子
balance-scale .....	⚖️
white-cane .....	杖
link.....	🔗
chains.....	
hook.....	钩
toolbox .....	工具箱
magnet.....	磁铁
ladder.....	梯子
SCIENCE	
alembic .....	/distillation-flask
test-tube.....	试管
petri-dish .....	培养皿
dna.....	DNA
microscope .....	显微镜
telescope.....	望远镜
satellite-antenna....	卫星天线
MEDICAL	
syringe .....	注射器
drop-of-blood .....	血滴
pill.....	药丸
adhesive-bandage .....	创可贴
crutch.....	拐杖
stethoscope.....	听诊器
x-ray .....	X光
HOUSEHOLD	
door .....	门
elevator .....	电梯
mirror.....	镜子
window.....	窗户
bed.....	床
couch-and-lamp.....	沙发和台灯
chair .....	椅子
toilet.....	马桶
plunger .....	通水管
shower.....	淋浴
bathtub .....	浴缸
mouse-trap .....	老鼠夹
razor .....	剃须刀
lotion-bottle .....	乳液瓶
safety-pin .....	别针
broom .....	扫帚

basket.....		WARNING	RELIGION
roll-of-paper .....			
bucket.....		warning .....	
soap .....		children-crossing.....	
bubbles .....		no-entry .....	
toothbrush .....		prohibited .....	
sponge.....		no-bicycles.....	
fire-extinguisher.....		no-smoking .....	
shopping-cart .....		no-littering.....	
<hr/>		non-potable-water.....	
<hr/>		no-peDESTrians.....	
<hr/>		no-mobile-phones .....	
<hr/>		no-one-under-	
<hr/>		eighteen .....	
<hr/>		radioactive.....	
<hr/>		biohazard.....	
<hr/>		ARROW	ZODIAC
nazar-amulet.....		up-arrow .....	
hamsa .....		up-right-arrow.....	
moai .....		right-arrow.....	
placard .....		down-right-arrow .....	
identification- card .....		down-arrow .....	
<hr/>		down-left-arrow .....	
<hr/>		left-arrow .....	
<hr/>		up-left-arrow .....	
<hr/>		up-down-arrow .....	
<hr/>		left-right-arrow .....	
<hr/>		right-arrow-curving- left .....	
<hr/>		left-arrow-curving- right .....	
<hr/>		right-arrow-curving- up .....	
<hr/>		right-arrow-curving- down .....	
<hr/>		clockwise-vertical- arrows.....	
<hr/>		counterclockwise- arrows-button .....	
<hr/>		back-arrow .....	
<hr/>		end-arrow.....	
<hr/>		on-arrow .....	
<hr/>		soon-arrow .....	
<hr/>		top-arrow.....	
<hr/>			next-track-button.....
<b>symbols</b>			
<hr/>			
TRANSPORT-SIGN			
atm-sign .....			
litter-in-bin-sign .....			
potable-water .....			
wheelchair-symbol.....			
mens-room.....			
womens-room.....			
restroom .....			
baby-symbol.....			
water-closet.....			
passport-control .....			
customs .....			
baggage-claim .....			
left-luggage.....			

play-or-pause-button	▶	white-question-mark	?	keycap-zero.....
reverse-button	◀	white-exclamation-mark	!	keycap-one.....
fast-reverse-button	⏪	red-exclamation-mark	❗	keycap-two.....
last-track-button	⏮	mark	!	keycap-three.....
upwards-button	🔼	wavy-dash.....	〰	keycap-four.....
fast-up-button	⏫	CURRENCY		keycap-five.....
downwards-button	🔽			keycap-six.....
fast-down-button	⏬			keycap-seven.....
pause-button	⏸	currency-exchange.....	\$¥	keycap-eight.....
stop-button	⏹	heavy-dollar-sign.....	\$	keycap-nine.....
record-button	⏺	OTHER-SYMBOL		keycap-ten.....
eject-button	⏏			ALPHANUM
cinema	🎦			input-latin-uppercase.....
dim-button	☀️	medical-symbol.....	⚕️	input-latin-lowercase.....
bright-button	☀️	recycling-symbol .....	♻️	input-numbers.....
antenna-bars	📶	fleur-de-lis.....	⚜️	input-symbols.....
wireless	📶	trident-emblem.....	🔱	input-latin-letters.....
vibration-mode	📱	name-badge .....	📛	a-button .....
mobile-phone-off	📴	japanese-symbol-for-beginner .....	🔰	ab-button .....
GENDER		hollow-red-circle.....	⭕	b-button .....
female-sign	♀	check-mark-button.....	✓	cl-button .....
male-sign	♂	check-box-with-check .....	✓	cool-button .....
transgender-symbol	⚧	check-mark .....	✓	free-button .....
MATH		cross-mark .....	✗	information .....
multiply	✖	cross-mark-button .....	✗	id-button .....
plus	➕	curly-loop .....	♾️	circled-m .....
minus	➖	double-curly-loop .....	♾️	new-button .....
divide	➗	part-alternation-mark .....	〽️	ng-button .....
heavy-equals-sign	⩵	eight-spoked-asterisk .....	✳️	o-button .....
infinity	♾️	eight-pointed-star .....	✳️	ok-button .....
PUNCTUATION		sparkle .....	✳️	p-button .....
double-exclamation-mark	‼️	copyright .....	©	sos-button .....
exclamation-question-mark	⁉️	registered .....	®	up-button .....
red-question-mark	❓	trade-mark .....	™	vs-button .....
KEYCAP				japanese-here-button .....
				japanese-service-charge-button .....
				japanese-monthly-amount-button .....

japanese-not-free-of-charge-button .....	有	yellow-circle .....	●	red-triangle-pointed-down .....	▼
japanese-reserved-button .....	指	green-circle .....	●	blue-circle .....	●
japanese-bargain-button .....	得	purple-circle .....	●	diamond-with-a-dot .....	◆
japanese-discount-button .....	割	brown-circle .....	●	radio-button .....	●
japanese-free-of-charge-button .....	無	black-circle .....	●	white-square-button .....	■
japanese-prohibited-button .....	禁	white-circle .....	●	black-square-button .....	■
japanese-acceptable-button .....	可	red-square .....	●	red-square .....	■
japanese-application-button .....	申	orange-square .....	●	yellow-square .....	●
japanese-passing-grade-button .....	合	green-square .....	●	green-square .....	●
japanese-vacancy-button .....	空	blue-square .....	●	blue-square .....	●
japanese-congratulations-button .....	祝	purple-square .....	●	purple-square .....	●
japanese-secret-button .....	秘	brown-square .....	●	chequered-flag .....	🏁
japanese-open-for-business-button .....	营	black-large-square .....	■	triangular-flag .....	🚩
japanese-no-vacancy-button .....	满	white-large-square .....	■	crossed-flags .....	🎌
<hr/>		black-medium-square .....	■	black-flag .....	🏴
GEOMETRIC		white-medium-square .....	■	white-flag .....	🏳️
<hr/>		black-medium-small-square .....	■	rainbow-flag .....	🏳️‍🌈
red-circle .....		white-medium-small-square .....	■	transgender-flag .....	🏳️⚧️
<hr/>		black-small-square .....	■	pirate-flag .....	🏴‍☠️
<hr/>		white-small-square .....	■	COUNTRY-FLAG	
<hr/>		large-orange-diamond .....	◆	eu .....	🇪🇺
<hr/>		large-blue-diamond .....	◆	<hr/>	
red-circle .....		small-orange-diamond .....	◆	extra	
<hr/>		small-blue-diamond .....	◆	EXTRA	
<hr/>		red-triangle-pointed-up .....	▲	lingchi .....	凌遲
<hr/>		orange-circle .....	●	YHWH .....	יהה

## 2.2 Two Optional Macros

\lingchi The \lingchi and \YHWH macros take no arguments and produce 凌遲 and יהה. They are only available if emo is used with the extra option. The former renders the Chinese term for “death by a thousand cuts.” While originally an execution method, the term applies to surprisingly many software systems as well. The latter produces the Tetragrammaton, the Hebrew name for God. Observant Jews never utter what’s written, not even in their thoughts, substituting Adonai (“My Lord”), Elohim (“God”), or HaShem (“The Name”) instead. In my mind, that nicely mirrors the very incomprehensibility of יהה. Both macros preserve a subsequent space as space, no backslash needed.

## 2.3 Conversion to HTML

Emo supports conversion to HTML with either [LaTeXML](#) or [TeX4ht](#). LaTeXML support is implemented by a separate “binding” against LaTeXML’s Perl API. I chronicled my exploration of suitable options leading to that less than ideal choice in a [GitHub issue](#). TeX4ht support is implemented by the `emo` package itself. It requires processing with LuaLaTeX e.g., by passing `-l` or `--lua` to the `make4ht` tool.

# 3 Configuration

Emo’s implementation is actually split over two files: `emo.sty` is extracted from `emo.dtx` and defines the substance of the package, its options, its helper macros, and the user-visible `\emo`, `\lingchi`, and `\YHWH` macros. Currently supported emoji are defined by the emoji table in the second file, `emo.def`. For every supported emoji, the file contains an invocation of the `\DefineEmoji<emoji-name><emoji-codepoints>` macro with the emoji’s name and codepoints. With exception of the emoji listed in Table 1, all names are the Unicode names written with a dash instead of space and using only lowercase letters. The codepoints always are the fully qualified Unicode codepoints for the emoji.

Configuration automates the regeneration of the emoji table for arbitrary numbers of emoji. `config/emo.py` is the script and `config/emoji-test.txt` is the list of all emoji from the Unicode standard.

## 3.1 Running the Configuration Script

To update emo’s configuration, invoke the `config/emo.py` script:

```
$ python3 config/emo.py <selector> <selector> ...
```

Each selector may be:

- The literal `ALL` (case-sensitive) for *all* emoji.
- Name of a group in `emoji-test.txt` lowercased and with spaces replaced by dashes and ampersand `&` replaced by an `and`; e.g., `travel-and-places`.
- Name of a group, a double colon `::`, and name of a subgroup, again lowercased and with spaces replaced by dashes and `&` by an `and`; e.g., `travel-and-places::placegeographic`.
- The name of an emoji; e.g., `desert-island`.

For conjunctive group names, such as “Smileys & Emotion” (`emoji-test.txt`) or “smileys-and-emotion” (`emo.py`), the configuration script also accepts either of the two nouns as a shortcut, e.g., “smileys” or “emotion.”

For data safety, `emo.py` does not overwrite PDF graphics and hence can only *add* emoji to the configuration. To remove emoji, simply remove their PDF graphics from `emo-graphics` and then run `emo.py` without selector arguments, which updates the emoji table accordingly.

`emo.py` effectively treats `emoji-test.txt` as registry of all emoji and the filenames of PDF graphics in `emo-graphics` as emo’s current inventory. For all emoji named by selector arguments but not in the inventory, `emo.py` converts the SVG source graphic from the Noto color emoji sources to a PDF file and deletes the `/Page /Group` object from

the the PDF again, since that object trips up `pdflatex`. And yeah, `emo.py` automatically downloads the Noto color emoji sources if necessary.

## 4 Copyright and Licensing

Since `emo`'s distribution includes not only LaTeX code but also a substantial Python script, Unicode data about emoji, as well as graphics and fonts derived from Google's Noto project, a number of different licenses apply. All of them are [OSI approved](#) and non-copyleft:

- This package's LaTeX and also Perl code extracted from `emo.dtx` is © Copyright 2023 by Robert Grimm and has been released under the [LPPL v1.3c](#) or later.
- The `config/emo.py` script also is © Copyright 2023 by Robert Grimm but has been released under the [Apache 2.0 license](#).
- The `[config/emoji-test.txt]` configuration file is a data file from [Unicode TR-51](#) and hence subject to the [Unicode License](#).
- The `emo-lingchi.ttf` font is a two-glyph subset of the traditional Chinese version of Google's [Noto serif](#) and hence subject to the [SIL Open Font License v1.1](#).
- The PDF graphics in the `emo-graphics` directory are derived from the sources for [Noto's color emoji](#) and hence subject to the Apache 2.0 license.

## 5 Implementation

Let's get started on `emo`'s implementation:

```
1 (*package)
```

Except, that implementation started near the top of this `emo.dtx` file, well before the documentation's preamble. For completeness, here is the package declaration again:

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{emo}
[2023/05/22 v1.0 emo.ji for all (LaTeX engines)]
```

Unfortunately, `emo`'s version number appears in triplicate in this file, which makes cutting a release a little more tricky than it should be. Alas, the above, faux package declaration is not one of those repeat offenders. Really.

### 5.1 Package Options

```
\ifEmojiExtra Define a conditional flag for each package option.
```

```
\ifemo@index
\ifemo@debug 2 \newif\ifEmojiExtra
3 \newif\ifemo@index
4 \newif\ifemo@debug
```

Wire each conditional to the package option and then process them all.

```
5 \DeclareOption{extra}{\EmojiExtratrue}
```

```

6 \DeclareOption{index}{\emo@indextrue}
7 \DeclareOption{debug}{\emo@debugtrue}
8 \ProcessOptions\relax

```

## 5.2 Package Dependencies and Backend Selection

Require `inputenc` to declare this file's character encoding as UTF-8. XeTeX and LuaTeX already use that encoding by default and hence this line is redundant. But pdfTeX originates from darker, pre-Unicode times and thus needs to be told. Though thankfully, it too does support the encoding.

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

`\ifemo@use@unicode` Define a conditional flag for each major backend feature. They are *not* orthogonal: `\ifemo@use@font` and `\ifemo@use@pdf` are mutually exclusive and determine `\ifemo@use@pdf` whether `\emo` generates Unicode text or PDF graphics. For `\ifemo@use@font` to be enabled, `\ifemo@use@unicode` must already be enabled as well.

```

10 \newif\ifemo@use@unicode
11 \newif\ifemo@use@font
12 \newif\ifemo@use@pdf

```

Inspect the runtime environment and accordingly set the just defined backend flags. `\HCode` is defined by TeX4ht. Since correctly converting to HTML with TeX4ht requires LuaTeX, print a helpful error message for other engines.

```

13 \RequirePackage{iftex}
14 \ifdef{\HCode}
15 \ifluatex\else
16 \PackageError{\emo}{%
17   Use of TeX4ht for converting LaTeX to HTML requires LuaTeX.\MessageBreak
18   You may see ‘‘Missing character’’ messages or\MessageBreak
19   ‘‘Unicode character ... not set up’’ errors.\MessageBreak
20   To fix, please pass the -l or --lua option to TeX4ht’s make4ht tool}{%
21   Run TeX4ht with LuaTeX by passing -l or --lua option to make4ht}
22 \fi
23 \emo@use@unicodetrue
24 \else
25 \ifluatex
26 \emo@use@unicodetrue
27 \emo@use@fonttrue
28 \else
29 \emo@use@pdftrue
30 \fi
31 \fi

```

We need the backend flags to decide whether to load `fontspec` or not. We do that before requiring any other packages because, that way, the packages related to encoding and fonts are loaded first (together with `iftex`).

```
32 \ifemo@use@font
```

```
33 \RequirePackage{fontspec}  
34 \fi
```

Require `xcolor` for formatting highly visible error messages within the generated document. Always including another package that is only used when there are errors is not ideal. But I couldn't get on-demand package loading to work. So we have to eagerly require the package.

```
35 \RequirePackage{xcolor}
```

The two remaining requirements depends on a package option and on the backend, respectively.

```
36 \ifemo@index  
37 \RequirePackage{index}  
38 \fi  
39 \ifemo@use@pdf  
40 \RequirePackage{graphicx}  
41 \fi
```

### 5.3 The Emoji Table

The emoji table contains an entry for every emoji supported in the current configuration. To facilitate reuse of the same table definition for different purposes, including validating emoji names, determining emoji codepoints, and typesetting the currently supported inventory, `emo` abstracts over the table itself with the following five macros.

`\EmojiBeginGroup` The file format for `emo.def` relies on `\DefineEmoji` to associate emoji names with their `\EmojiBeginSubgroup` Unicode codepoints, `\EmojiBeginGroup` and `\EmojiEndGroup` to organize emoji into `\DefineEmoji` groups, and on `\EmojiBeginSubgroup` and `\EmojiEndSubgroup` to organize emoji into `\EmojiEndSubgroup` subgroups. All definitions with `\DefineEmoji` are listed in Unicode display order and `\EmojiEndGroup` grouped in their Unicode group and subgroup. Subgroups are properly nested inside groups. The group and subgroup for `lingchi` and `YHWH` are both called `extra`.

`\DefineEmoji{\<name>}{\<codepoints>}` takes an emoji's name and its Unicode codepoints as arguments—except for `keycap-hash`, which uses has the `\char`ed` codepoints as value because TeX gets very much confused by the leading hash character in that emoji's codepoints.

`\EmojiBeginGroup{\<group>}` and `\EmojiEndGroup{\<group>}` take a group name as their only argument. In contrast, `\EmojiBeginSubgroup{\<group>}{\<subgroup>}` and `\EmojiEndSubgroup{\<group>}{\<subgroup>}` take both the group and subgroup names as their two arguments.

If the `extra` option is enabled, `\lingchi` and `\YHWH` are treated just like emoji, and their names and Unicode codepoints (sans any group, font selection, or text direction) are included with the emoji table.

The conversion from the abstract file format for `emo.def` to the concrete emoji table is simple enough, especially since four macros remain no-ops:

```
42 \def\EmojiBeginGroup#1{}  
43 \def\EmojiBeginSubgroup#1#2{}
```

```

44 \def\DefineEmoji#1#2{%
45     \expandafter\def\csname emo@emoji@#1\endcsname{#2}%
46 \def\EmojiEndSubgroup#1#2{}%
47 \def\EmojiEndGroup#1{}%

```

With that, we are ready to read `emo.def`:

```
48 \input{emo.def}
```

The emoji table is automatically generated with the `config/emo.py` Python script. When invoked without arguments, that script simply recreates the emoji table based on the emoji present in the `emo-graphics` directory. If invoked with arguments naming Unicode emoji groups, groups and subgroups, or individual emoji, the script converts SVG graphics from the Noto emoji sources to PDF graphics compatible with LaTeX and then updates the inventory in `emo.def`. The Python script even downloads the Noto emoji sources if they haven't been already downloaded to the local machine.

## 5.4 Internal Macros

`\emo@error@fg` Define two colors and the `\emo@error{<emoji-name>}` macro. The latter uses the two `\emo@error@bg` colors for formatting an attention-grabbing error message. If you use an invalid emoji `\emo@error` name and overlook the warning in the console, you *will* notice the error message in the document thusly formatted.

```

49 \definecolor{\emo@error@fg}{rgb}{1,1,1}
50 \definecolor{\emo@error@bg}{rgb}{.6824,.0863,.0863}
51 \def\emo@error#1{%
52     \colorbox{\emo@error@bg}{%
53         \textcolor{\emo@error@fg}{%
54             \textsf{Bad} \texttt{\char`\\emo\{#1\}}}}}

```

`\emo@ifdef` The `\emo@ifdef{<name>}{<code>}` macro validates the emoji name given as first argument. If the name is invalid, it expands to an error message. If the name is valid, it executes the code given as the second argument. The implementation critically relies on the emoji table being accurate.

```

55 \def\emo@ifdef#1#2{%
56     \ifcsname \emo@emoji@#1\endcsname#2\else%
57         \PackageWarning{\emo}{Unknown emoji name in '\string\emo{#1}'}%
58         \emo@error{#1}%
59     \fi}

```

## 5.5 Emo's Public Macros and Render Hooks

To support up to three macros across three LaTeX engines and two HTML conversion tools, `emo`'s implementation originally defined the same macros twice or thrice, with some code duplication between them as well. As I gained experience with LaTeX, I settled on *hooks* as a powerful means for avoiding extra macros as well as code duplication.

### 5.5.1 A Skeleton of Render Hooks

- emo/render/before To determine just those hooks, let's review `emo`'s functional requirements and pay close  
emo/render/emoji attention to how those requirements converge and diverge between `emo`'s three macros:  
emo/render/chinese  
emo/render/hebrew  
emo/render/content  
emo/render/after
1. All three macros visually render one or more glyphs. Let's introduce the `emo/render/content` hook for doing so.
  2. Each macro generates glyphs in a different "language" and writing system and thus may need to set the font, text direction, and so on. Let's introduce the `emo/render/emoji`, `emo/render/chinese`, and `emo/render/hebrew` hooks for that.
  3. Just as in aspect-oriented programming, we sometimes need to track macro invocations (e.g., for the `index` package option) or modify macro results (e.g., for the `debug` package option). Let's introduce the `emo/render/before` and `emo/render/after` hooks for doing so.

Before version 1.0, `\lingchi` and `\YHWH` relied on a trailing `\xspace` to obviate the need for trailing backslashes before spaces. Continued support would require introducing another hook, which would add more complexity than it avoids. For that reason, I removed that feature with version 1.0.

We are ready to define `emo`'s render hooks:

```
60 \NewHook{emo/render/before}
61 \NewHook{emo/render/emoji}
62 \ifEmojiExtra
63 \NewHook{emo/render/chinese}
64 \NewHook{emo/render/hebrew}
65 \fi
66 \NewHook{emo/render/content}
67 \NewReversedHook{emo/render/after}
```

`\emo@makecommand` With the hooks declared, we are ready to define the internal `\emo@makecommand` macro,  
`\emo@command` which provides the shared skeleton for each of the three user macros. Since those three  
`\emo@key` macros differ in control sequence and font selection hook, `\emo@makecommand` takes two  
`\emo@value` arguments, the macro control sequence and the font selection hook. The user macro defined by `\emo@makecommand` in turn takes one argument, the name of the entity to render. Its body invokes the right hooks in the right order. Since hooks do not pass arguments or results, it also creates bindings for the current command, the key argument, and the resulting, rendered value. They are named `\emo@command`, `\emo@key`, and `\emo@value`, respectively.

Let's put this all together: `\emo@makecommand` is a macro-generating macro and hence defines a new user macro. That macro's implementation first binds `\emo@command` and `\emo@key`. It then tests whether the key is valid. If so, it invokes the `emo/render/before` hook, renders the content by invoking the font selection and content hooks inside a group, binds the result to `\emo@value`, invokes the `emo/render/after` hook, and yields the value. The group ensures that any sticky changes to the font and so on are nicely contained within the macro invocation.

```
68 \def\emo@makecommand#1#2{
69     \newcommand*{#1}[1]{%
```

```

70      \def\emo@command{\#1}%
71      \def\emo@key{\##1}%
72      \emo@ifdef{\#1}{%
73          \UseHook{emo/render/before}%
74          \def\emo@value{%
75              \begingroup%
76              \#2%
77              \UseHook{emo/render/content}%
78              \endgroup}%
79          \UseHook{emo/render/after}%
80          \emo@value}}}

```

Nice and simple? I'd say so! But it took some trial and error to arrive at this implementation. In particular, I started out with `\emo@makecommand` only assembling the body of the user macro instead of fully defining it. That required using `\edef` for the macro definition and prefixing almost every token in `\emo@makecommand` above with `\noexpand`. Once I realized that I can pass the control sequence as an argument, I was able to write this far simpler version.

### 5.5.2 The User Macros

`\emo` Render the named emoji either as Unicode or as a PDF graphic using the emoji-specific render hook.

```
81 \emo@makecommand\emo{\UseHook{emo/render/emoji}}
```

`\lingchi` Render 凌遲 and 𩫱𩫱, respectively. Since neither `\lingchi` nor `\YHWH` takes arguments, `\YHWH` we require a helper macro each that provides the argument expected by the skeleton implementation.

```

82 \ifEmojiExtra
83 \emo@makecommand\emo@lingchi{\UseHook{emo/render/chinese}}
84 \emo@makecommand\emo@YHWH{\UseHook{emo/render/hebrew}}
85 \newcommand*\lingchi{\emo@lingchi{lingchi}}
86 \newcommand*\YHWH{\emo@YHWH{YHWH}}
87 \fi

```

### 5.5.3 Activating the Hooks

So far, the three user macros aren't particularly useful: They do not generate any visible content. For that, we need to activate the hooks as necessary for the current engine and package options. Conveniently, we only need to add to a hook if we need it to do something. In the following, we activate hooks in invocation order, from `emo/render/before` to font selection hooks to `emo/render/content` to `emo/render/after`.

If the `index` option is enabled, create the necessary index and emit entries through the `emo/render/before` hook. The hook uses the `\emo@emit@index` helper macro because stepping through `\index[emo]` would require three more `\expandafter` invocations than this one:

```

88 \ifemo@index
89 \newindex{emo}{edx}{end}{Emoji Index}

```

```

90 \def\emo@emit@index#1{\index[emo]{#1}}
91 \AddToHook{emo/render/before}{%
92   \expandafter\emo@emit@index\expandafter{\emo@key}}
93 \fi

```

Next, if the backend uses fonts, set up font selection in the corresponding hooks. We only activate the `emo/render/chinese` and `emo/render/hebrew` hooks if the `extra` package option is enabled. Note that since Hebrew is written right-to-left, the font selection hook also set the text direction. This is safe to do because font selection and content are always enclosed in a group.

```

94 \ifemo@use@font
95 \newfontface\emo@font@emoji[Renderer=Harfbuzz]{NotoColorEmoji.ttf}
96 \AddToHook{emo/render/emoji}{\emo@font@emoji}
97 \ifEmojieExtra
98 \newfontface\emo@font@chinese{emo-lingchi.ttf}
99 \newfontface\emo@font@hebrew{LinLibertine_R.otf}
100 \AddToHook{emo/render/chinese}{\emo@font@chinese}
101 \AddToHook{emo/render/hebrew}{\emo@font@hebrew{textdir TRT}}
102 \fi
103 \fi

```

Next, we activate the `emo/render/content` hook to either emit Unicode codepoints or PDF graphics.

```

104 \ifemo@use@unicode
105 \AddToHook{emo/render/content}{\csname emoji@\emo@key\endcsname}
106 \fi
107 \ifemo@use@pdf
108 \AddToHook{emo/render/content}{%
109   \raisebox{-0.2ex}{%
110     \includegraphics[height=1em]{emo-graphics/emo-\emo@key}}}
111 \fi

```

Finally, if the `debug` package option is enabled, we wrap the rendered content in a frame box.

```

112 \ifemo@debug
113 \AddToHook{emo/render/after}{%
114   \let\emo@realvalue\emo@value%
115   \def\emo@value{\fbox{\emo@realvalue}}}
116 \fi

```

Et voilà. That's it!

```
117 </package>
```

#### 5.5.4 Background: Variability vs Code Duplication

Using hooks, `emo`'s implementation can handle considerable variability in the runtime environment with less than 60 lines of well-documented LaTeX code. That's pretty neat.

But `emo` most certainly didn't start out that way and its implementation changed considerably between version 0.1 and 1.0.

**Versions 0.1 and 0.2** took the most direct approach to managing the variability between LaTeX engines: Conditionally define `\emo`, `\lingchi`, and `\YHWH` twice, once using PDF graphics and once using fonts and Unicode codepoints. It also conditionally defined `\emo@index` twice, once when the package option is off and once when it is on. That approach works, but it also results in code duplication. Furthermore, it doesn't scale when the variability increases.

**Version 0.3** integrated TeX4ht into the implementation, which effectively added a third backend. Despite supporting more functionality than the previous two versions, its implementation seemed simpler. The primary reason was that I added `\lingchi` and `\YHWH` to the emoji table. That, in turn, made it possible to use the same code for all three user macros for two backends. Only the font-based backend required three implementations, which were rather similar as well.

**Version 0.4** introduced the debug package option, hence increasing variability again. Sadly, the implementation added two extra definitions for each of the three user macros, hence increasing clutter and code duplication again. At the same time, the experience of adding unit tests helped improve my TeX-fu considerably. So after releasing that version, I started looking for a more elegant approach to `emo`'s implementation and discovered [LaTeX hooks](#).

**Version 1.0** switched to using hooks. It also abstracted over the emoji table contents and significantly grew the default inventory of emoji.

## 6 LaTeXML Binding

To support conversion from LaTeX to HTML, `emo` includes a so-called binding for [LaTeXML](#). It effectively is a (much simplified) re-implementation of `emo`'s core functionality, only written in Perl against LaTeXML's API. The binding ignores the `index` option and does not perform error checking on emoji names. If either is important to you, please compile the document with LaTeX first. Furthermore, the binding emits necessary Unicode codepoints only, without font annotations. If you want to specify fonts, please use a CSS fontstack.

Asking package authors to reimplement their packages for LaTeXML seems unreasonable to me. It leads to code duplication and places the maintenance burden on package authors. Yet, right after announcing `emo`, the question of LaTeXML support came up. LaTeXML includes the `latexml.sty` package, which defines `\iflatexml`. I would have used that command to make the three-line change to `emo.sty` necessary to support LaTeXML, except `latexml.sty` contains lots of other stuff that isn't needed. Always loading lots of macros only to detect LaTeXML slows down compilation and wastes memory. Since reimplementing `\iflatexml` would require a binding anyways, I just wrote a minimal binding. As I said, LaTeXML's approach is broken.

With that out of the way, let's get started:

```
1 (*latexml-binding)
```

The binding starts with an explicit preamble because `docstrip` does not allow for a

redefinition of the starting characters of a line comment. It is followed by the Perl dependencies.

```

2## emo's LaTeXML binding.
3## (C) 2023 by Robert Grimm.
4## Released under LPPL v1.3c or later.
5use strict;
6use warnings;
7use LaTeXML::Package;
```

\ifEmojiExtra Next, we use raw TeX to declare the LaTeX package and the \ifEmojiExtra conditional. The binding does not require any other conditionals since it only runs under LaTeXML and does not support options other than extra.

```

8RawTeX(<<'EOTeX');
9\ProvidesPackage{emo}
10    [2023/05/22 v1.0 emo.ji for all (LaTeX engines)]
11\newif\ifEmojiExtra
12EOTeX
```

Option processing is almost trivial:

```

13DeclareOption('extra', '\EmojiExtratrue');
14DeclareOption('index', '');
15DeclareOption('debug', '');
16ProcessOptions();
```

\EmojiBeginGroup Define the five macros for parsing the emoji table. They are the same as those defined \EmojiBeginSubgroup by the package itself.

```

\DefineEmoji
\EmojiEndSubgroup17DefMacro('\EmojiBeginGroup{}', '');
\EmojiEndGroup18DefMacro('\EmojiBeginSubgroup{}{}', '');
19DefMacro('\DefineEmoji{}{}',
20    '\expandafter\def\csname emoji@#1\endcsname{#2}');
21DefMacro('\EmojiEndSubgroup{}{}', '');
22DefMacro('\EmojiEndGroup{}', '');
```

Thusly prepared, read in the emoji table.

```
23InputDefinitions('emo', type => 'def', noltxml => 1);
```

\emo Define the simplest possible version of \emo. It has no hooks, no error checking, no font selection, and no support for index or debug. It does, however, expand the emoji table entry.

```
24DefMacro('\emo{}', '\csname emoji@#1\endcsname');
```

\lingchi If the \EmojiExtra conditional is enabled, provide similarly minimal re-definitions of the \YHWH \lingchi and \YHWH macros.

```
25if (IfCondition(T_CS('\ifEmojiExtra'))) {
```

```
26     DefMacro('\'lingchi', "\x{51cc}\x{9072}");
27     DefMacro('\'YHWH', "\x{05D9}\x{05D4}\x{05D5}\x{05D4}");
28 }
```

That's it for the binding, too.

```
29 </latexml-binding>
```

## 7 Testing and Documenting Emo

The following package and documents help test `emo` across all major LaTeX engines and generate informative reports.

Alas, we don't want to run these tests while generating the documentation. Let's prevent that:

```
30 <*scaffold>
31 \iffalse
32 </scaffold>
```

### 7.1 Emo's Support Package

```
1 <*support>
```

The `emo-support` package defines high-level macros that make testing and documenting the `emo` package easier. At the same time, no attempt has been made to make `emo-support` reusable. The package exists only to support `emo`.

```
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{emo-support}[2023/05/22 v1.0 Test & document emo]
```

There are no options to process.

```
4 \ProcessOptions\relax
```

We require `iftex` for determining the LaTeX engine, `multicol` for compact inventory listings, `xcolor` for appearances, and `emo` itself for its `\ifemo@use@unicode`, `\ifemo@use@font`, and `\ifemo@use@pdf` conditionals. This package also reuses `emo`'s font variables during test execution.

```
5 \RequirePackage{iftex}
6 \RequirePackage{multicol}
7 \RequirePackage{xcolor}
8 \RequirePackage[extra]{emo}
```

Since font selection across engines is a mouthful, we select the document fonts here inside the support package. That may include requiring `fontspec`, too.

```
9 \iftutex
10 \RequirePackage{fontspec}
11 \RequirePackage{libertinus}
12 \setmonofont{inconsolata}
13 \else
```

```

14 \RequirePackage{libertinus}
15 \RequirePackage{inconsolata}
16 \fi

```

### 7.1.1 Report Generation

\enginename I couldn't find an existing macro that provides the desired functionality, so we gotta round up the usual suspects to define the engine name.

```

17 \ifxetex
18 \def\enginename{XeTeX}
19 \else
20 \ifluatex
21 \def\enginename{LuaTeX}
22 \else
23 \ifpdftex
24 \def\enginename{pdfTeX}
25 \else
26 \def\enginename{\emph{unknown engine}}
27 \fi
28 \fi
29 \fi

```

emo@canary@frameinner Define frame and background colors for boundary boxes of sample text.

emo@canary@frameouter  
 emo@canary@background<sup>30</sup> \definecolor{emo@canary@frameinner}{HTML}{636366}  
<sup>31</sup> \definecolor{emo@canary@frameouter}{HTML}{48484A}  
<sup>32</sup> \definecolor{emo@canary@background}{HTML}{E5E5EA}

Adjust settings for \fcolorbox so that it serves as bounding box.

```

33 \setlength{\fboxrule}{0.5pt}
34 \setlength{\fboxsep}{0pt}

```

\emo@nobox The sample text may or may not show bounding boxens for words and emoji. It always  
 \emo@wordbox shows the bounding box for the entire line.

\emo@linebox  
 35 \newcommand\emo@nobox[1]{#1}  
 36 \newcommand\emo@wordbox[1]{%
 37 \fcolorbox{emo@canary@frameinner}{white}{#1}}
 38 \newcommand\emo@linebox[1]{%
 39 \fcolorbox{emo@canary@frameouter}{emo@canary@background}{#1}}

\@sampletext Show a single line of text that makes use of emo's three user macros. To help identify  
 \sampletext incorrect font metrics, spurious whitespace, and other issues, show the line's bounding  
 box and, for the starred version, the bounding boxens for words and emoji, too.

```

40 \def\@sampletext#1{%
41   \emo@linebox{%
42     #1{It's} #1{\lingchi}:%
43     #1{Please}, #1{\YHWH}, #1{have} #1{mercy}}

```

```

44      #1{\emo{pleading-face}!}%  

45      \vspace{1ex}  

46 \newcommand*\samplertext{  

47   \@ifstar{\@samplertext{\emo@wordbox}}{\@samplertext{\emo@nobox}}}

```

### 7.1.2 Emoji Inventory

The inventory makes use of custom section headings `\group` and `\subgroup`. Let's define their counters.

```

48 \newcounter{group}  

49 \newcounter{subgroup}[group]

```

`\emo@subgrouprule` Subgroup headings are particularly fancy. They show the subgroup name in small-caps `\emo@subgroupstyle` between horizontal rules. It's cleanest to use separate paragraphs for each rule as well as the subgroup name. To prevent very ugly column breaks between them, we wrap everything in a `minipage` environment.

```

50 \def\emo@subgrouprule{\rule{\linewidth}{1pt}\par}  

51 \def\emo@subgroupstyle#1{  

52   \begin{minipage}{\linewidth}  

53     \emo@subgrouprule\par  

54     \rmfamily\textsc{#1}\vspace{-1.0ex}\par  

55     \emo@subgrouprule  

56   \end{minipage}}

```

`\group` We are ready to define the new sectioning commands. By using `\@startsection`, we `\subgroup` get LaTeX's support for managing the space above and below the heading for free. That makes a huge difference, if the heading appears at the top of a page or columns.

```

57 \newcommand{\group}{%  

58   \@startsection{group}{1}{0pt}{  

59     {-3ex plus -1ex minus -0.2ex}{  

60     {3ex plus 1ex minus 0.2ex}{  

61       {\centering\rmfamily\Large\bfseries\itshape}}}  

62 \newcommand{\subgroup}{%  

63   \@startsection{subgroup}{2}{0pt}{  

64     {-3.5ex plus -1.0ex minus -0.2ex}{  

65     {2ex plus 0.5ex minus 0.2ex}{  

66       {\emo@subgroupstyle}}}

```

`\emo@dotfilla` The inventory formats emoji names flush left and the corresponding emoji flush right `\emo@lineup` with dots in between. Whereas `\emo@dotfilla` formats the dots in the middle, `\emo@lineup@emoji` `\emo@lineup` takes care of putting it all together into one paragraph. It also is careful to `\emo@lineup@extra` enable hyphenation for typewriter text. `\emo@lineup@emoji` and `\emo@lineup@extra` `\emo@group@extra` account for the differences between actual emoji and the two extra macros. Finally, `\emo@group@extra` names the extra group.

```

67 \def\emo@dotfilla{  

68   \unskip\nobreak%  

69   \leaders\hbox{\hspace{0.1ex}\hspace{0.1ex}}%

```

```

70      \hskip 1em plus 1fill\relax}
71 \def\emo@lineup#1#2{%
72     \noindent\raggedright%
73     \texttt{\hyphenchar\font=\-\#1}%
74     \emo@dotfilla#2\par}
75 \def\emo@lineup@emoji#1#2{\emo@lineup{#1}{\emo{#1}}}
76 \def\emo@lineup@extra#1#2{\emo@lineup{#1}{\csname#1\endcsname}}
77 \def\emo@group@extra{extra}

```

\listinventory With that, we are ready to define the \listinventory macro. It does all formatting by reading the emoji table from `emo.def` with its own version of the file format macros. It even changes \DefineEmoji while reading `emo.def` to account for the final *extra* group. The resulting format lists emojis with their names under \group and \subgroup headings in three columns per page, all contained in a group to contain format changes like the redefined \parskip.

```

78 \newcommand{\listinventory}{%
79     \begingroup
80     \setlength{\parskip}{0.3ex plus 0.2ex minus 0.1ex}
81     \let\DefineEmoji=\emo@lineup@emoji
82     \def\EmojiBeginGroup##1{%
83         \def\emo@group{##1}%
84         \ifx\emo@group\emo@group@extra%
85             \let\DefineEmoji=\emo@lineup@extra%
86         \fi%
87         \group*{##1}%
88         \def\EmojiBeginSubgroup##1##2{\subgroup*{##2}}
89         \begin{multicols}{3}
90         \input{emo.def}
91         \end{multicols}
92     \endgroup}

```

### 7.1.3 Testing

\emo@canary@actual Validating `emo`'s user macros turned out to be a bit trickier than I had expected. The \emo@canary@expected obvious approach, fully expanding the macros and then comparing the results, doesn't work. While TeX does support eager expansion via, for example, \expandafter and \edef, it does so only for some macros.

Instead, we need to take a sneakier approach: Generate a box with the macro invocation and another box with the expected result and then compare the dimensions of the two boxes (currently only widths). While that is an incomplete comparison and hence cannot detect all bugs, it *can* detect any bug where the macro's output is shorter or longer than expected. That conveniently includes whitespace, which is one of the likeliest regressions.

We get started on that testing strategy by defining two box registers:

```

93 \newsavebox{\emo@canary@actual}
94 \newsavebox{\emo@canary@expected}

```

\checkwidth The \checkwidth{\langle name\rangle}{\langle code\rangle}{\langle font\rangle}{\langle codepoints\rangle}{\langle file\rangle} macro can test emo's three user-facing macros across all engines. It captures the variability for doing so across all supported engines with its five arguments:

1. {\langle name\rangle} of the user macro being tested—e.g., `emo`;
2. {\langle code\rangle} to invoke the macro—e.g., `\emo{robot}`;
3. name of the command selecting {\langle font\rangle}—e.g., `emoji`;
4. expected Unicode {\langle codepoints\rangle}—e.g., `\char"1F916`;
5. {\langle file\rangle} name of the PDF graphic—e.g., `robot`.

\checkwidth's implementation starts by filling a box each with the actual macro content and the expected macro content. Since the actual content implicitly differs depending on backend, the latter must explicitly vary the expected content depending on backend.

```

95 \newcommand\checkwidth[5]{%
96     \sbox\emo@canary@actual{\#2}%
97     \ifemo@use@font%
98         \sbox\emo@canary@expected{%
99             \begingroup\csname\emo@font@\endcsname #4\endgroup}%
100    \else%
101    \ifemo@use@unicode%
102        \sbox\emo@canary@expected{\begingroup #4\endgroup}%
103    \else%
104        \sbox\emo@canary@expected{%
105            \raisebox{-0.2ex}{%
106                \includegraphics[height=1em]{emo-graphics/emo-\#5}}}%
107    \fi%
108    \fi%

```

It then generates the test report inside a box. The report states the macro name followed by a green check mark or red cross mark, depending on whether the two boxes have the same width. If they do not, the red cross mark is followed by each box's content and width.

```

109    \mbox{%
110    \texttt{\char`\#\#1}\space%
111    \ifdim\wd\emo@canary@actual=\wd\emo@canary@expected%
112        \emo{check-mark-button}%
113    \else%
114        \emo{cross-mark}%
115        \fbox{\usebox{\emo@canary@actual}}\space%
116        \the\wd\emo@canary@actual\space%
117        \fbox{\usebox{\emo@canary@expected}}\space%
118        \the\wd\emo@canary@expected%
119    \fi}%

```

That's it for emo's support package.

```
120 </support>
```

## 7.2 Engine Test and Report

```
1 <*canary>
```

Each test report identifies the LaTeX engine and then shows the results of the width tests for emoji's three user-visible macros. If a test passes, the output only contains the macro name and a  check mark. If a test fails, the output contains the macro name, the divergent box widths, and a  cross mark. In the latter case, the width test results spill into the next line (at the least).

```
2 \documentclass[border=10pt, varwidth=6in]{standalone}
3 \usepackage{emo-support}
4 \setlength\parindent{0pt}
5 \setlength{\parskip}{1ex}
6 \begin{document}
7 \Huge
8 \emo{keycap-hash} \enginename: {\Large
9 \checkwidth{\emo{\robot}}{\emoji{\char"1F916}{robot}},
10 \checkwidth{\lingchi}{\lingchi}{chinese}{\char"51CC\char"9072}{lingchi},
11 \checkwidth{\YHWH}{\YHWH}{hebrew}{%
12   \csname textdir\endcsname TRT\char"5D9\char"5D4\char"5D5\char"5D4%
13 }{\YHWH}}
14 \vspace{1ex}\par
```

Next is the sample text, first with and then without boundary boxes for words and emoji.

```
15 \sampletext*\par\sampletext
16 \end{document}
```

That's it for the tests and report.

```
17 </canary>
```

## 7.3 Simple Test Document

```
1 <*oneliner>
```

Not much to see here besides one line of contents.

```
2 \documentclass[border=10pt, varwidth=6in]{standalone}
3 \usepackage{emo-support}
4 \begin{document}
5 \Huge\sampletext
6 \end{document}
```

It's a wrap 😊

```
7 </oneliner>
```

## Change History

0.1	General: Make initial release . . . . .	1	engines with <code>canary.tex</code> . . . . .	1
0.2	General: Add LaTeXML binding for conversion to HTML . . . . .	28	Introduce a simple unit testing framework . . . . .	30
	Prefix font and graphic files with “emo-” . . . . .	1	<code>\ifemo@debug</code> : Add debug option for drawing boundary boxes . . . . .	21
	Support <code>pdftex</code> for extracting <code>emo.dtx</code> . . . . .	1		
0.3	General: Include <code>\lingchi</code> and <code>\YHWH</code> in emoji table . . . . .	23	1.0	
	Support <code>TeX4ht</code> for conversion to HTML . . . . .	1	General: Drop <code>\textdir</code> from emoji table entry for <code>\YHWH</code> . . . . .	23
	<code>\lingchi</code> : Build on <code>\emo</code> by default . . . . .	26	Introduce high-level emoji table format that preserves Unicode grouping and order . . . . .	23
	<code>\YHWH</code> : Build on <code>\emo</code> by default . . . . .	26	Use LaTeX hooks for handling engines and options . . . . .	1
0.4	General: Automate testing across		<code>\checkwidth</code> : Since <code>xspace</code> is not used anymore, drop period from measured content . . . . .	33
			<code>\lingchi</code> : Drop use of trailing <code>xspace</code> . . . . .	26
			<code>\YHWH</code> : Drop use of trailing <code>xspace</code> . . . . .	26

## 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.

Symbols	D
<code>\-</code> . . . . .	73
<code>\@ifstar</code> . . . . .	47
<code>\@sampletext</code> . . . . .	40
<code>\@startsection</code> . . . . .	58, 63
<code>\`</code> . . . . .	54, 110
<code>\{</code> . . . . .	54
<code>\}</code> . . . . .	54
 <b>A</b>	
<code>\AddToHook</code> . . . . .	91, 96, 100, 101, 105, 108, 113
 <b>B</b>	
<code>\begin</code> . . . . .	4, 6, 52, 89
<code>\begingroup</code> . . . . .	75, 79, 99, 102
<code>\bfseries</code> . . . . .	61
 <b>C</b>	
<code>\centering</code> . . . . .	61
<code>\char</code> . . . . .	9, 10, 12, 54, 110
<code>\checkwidth</code> . . . . .	9, 10, 11, 95
<code>\colorbox</code> . . . . .	52
<code>\csname</code> . . . . .	12, 20, 24, 45, 76, 99, 105
 <b>E</b>	
<code>\else</code> . . . . .	13, 15, 19, 22, 24, 25, 28, 56, 100, 103, 113
<code>\emo</code> . . . . .	8, 9, 24, 44, 57, 75, 81, 112, 114
<code>\emo/render/after</code> . . . . .	60
<code>\emo/render/before</code> . . . . .	60
<code>\emo/render/chinese</code> . . . . .	60
<code>\emo/render/content</code> . . . . .	60
<code>\emo/render/emoji</code> . . . . .	60
<code>\emo/render/hebrew</code> . . . . .	60
<code>\emo@canary@actual</code> .	93, 96, 111, 115, 116
<code>\emo@canary@background</code> . . . . .	30
<code>\emo@canary@expected</code> . . . . .	93, 98, 102, 104, 111, 117, 118
<code>\emo@canary@frameinner</code> . . . . .	30

\emo@canary@frameouter . . . . .	30
\emo@command . . . . .	68
\emo@debugtrue . . . . .	7
\emo@dotfilla . . . . .	67
\emo@emit@index . . . . .	90, 92
\emo@error . . . . .	49, 58
\emo@error@bg . . . . .	49
\emo@error@fg . . . . .	49
\emo@font@chinese . . . . .	98, 100
\emo@font@emoji . . . . .	95, 96
\emo@font@hebrew . . . . .	99, 101
\emo@group . . . . .	83, 84
\emo@group@extra . . . . .	67, 84
\emo@ifdef . . . . .	55, 72
\emo@indextrue . . . . .	6
\emo@key . . . . .	68, 92, 105, 110
\emo@linebox . . . . .	35, 41
\emo@lineup . . . . .	67
\emo@lineup@emoji . . . . .	67, 81
\emo@lineup@extra . . . . .	67, 85
\emo@lingchi . . . . .	83, 85
\emo@makecommand . . . . .	68, 81, 83, 84
\emo@nobox . . . . .	35, 47
\emo@realvalue . . . . .	114, 115
\emo@subgrouprule . . . . .	50
\emo@subgroupstyle . . . . .	50, 66
\emo@use@fonttrue . . . . .	27
\emo@use@pdftrue . . . . .	29
\emo@use@unicodetrue . . . . .	23, 26
\emo@value . . . . .	68, 114, 115
\emo@wordbox . . . . .	35, 47
\emo@YHWH . . . . .	84, 86
\EmojiBeginGroup . . . . .	17, 42, 82
\EmojiBeginSubgroup . . . . .	17, 43, 88
\EmojiBeginSubroup . . . . .	42
\EmojiEndGroup . . . . .	17, 42
\EmojiEndSubgroup . . . . .	17, 42
\EmojiExtratrue . . . . .	5, 13
\emph . . . . .	26
\end . . . . .	6, 16, 56, 91
\endcsname . . . . .	12, 20, 24, 45, 56, 76, 99, 105
\endgroup . . . . .	78, 92, 99, 102
\enginename . . . . .	8, 17
\expandafter . . . . .	20, 45, 92
<b>F</b>	
\fbox . . . . .	115, 115, 117
\fboxrule . . . . .	33
\fboxsep . . . . .	34
\fcolorbox . . . . .	37, 39
\fi . . . . .	16, 22, 27, 28, 29, 30, 31, 34, 38, 41, 59, 65, 86, 87, 93, 102, 103, 106, 107, 108, 111, 116, 119
\font . . . . .	73
<b>G</b>	
\group . . . . .	57, 87
<b>H</b>	
\hbox . . . . .	69
\HCode . . . . .	14
\hskip . . . . .	69, 70
\Huge . . . . .	5, 7
\hyphenchar . . . . .	73
<b>I</b>	
\ifcsname . . . . .	56
\ifdefined . . . . .	14
\ifdim . . . . .	111
\ifemo@debug . . . . .	2, 112
\ifemo@index . . . . .	2, 36, 88
\ifemo@use@font . . . . .	10, 32, 94, 97
\ifemo@use@pdf . . . . .	10, 39, 107
\ifemo@use@unicode . . . . .	10, 101, 104
\ifEmojiExtra . . . . .	2, 8, 25, 62, 82, 97
\iffalse . . . . .	31
\ifluatex . . . . .	15, 20, 25
\ifpdftex . . . . .	23
\iftutex . . . . .	9
\ifx . . . . .	84
\ifxetex . . . . .	17
\includegraphics . . . . .	106, 110
\index . . . . .	90
\input . . . . .	48, 90
\itshape . . . . .	61
<b>L</b>	
\Large . . . . .	8, 61
\leaders . . . . .	69
\let . . . . .	81, 85, 114
\linewidth . . . . .	50, 52
\lingchi . . . . .	10, 25, 42, 82
\listinventory . . . . .	78
<b>M</b>	
\mbox . . . . .	109
\MessageBreak . . . . .	17, 18, 19
<b>N</b>	
\NeedsTeXFormat . . . . .	2
\newcommand . . . . .	35,
	36, 38, 46, 57, 62, 69, 78, 85, 86, 95
\newcounter . . . . .	48, 49
\newfontface . . . . .	95, 98, 99
\NewHook . . . . .	60, 61, 63, 64, 66
\newif . . . . .	2, 3, 4, 10, 11, 11, 12
\newindex . . . . .	89
\NewReversedHook . . . . .	67
\newsavebox . . . . .	93, 94
\nobreak . . . . .	68
\noindent . . . . .	72

<b>P</b>	
\PackageError	16
\PackageWarning	57
\par	14, 15, 50, 53, 54, 74
\parindent	4
\parskip	5, 80
\ProcessOptions	4, 8
\ProvidesPackage	3, 9
<b>R</b>	
\raggedright	72
\raisebox	105, 109
\relax	4, 8, 70
\RequirePackage	5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 33, 35, 37, 40
\rmfamily	54, 61
\rule	50
<b>S</b>	
\sampletext	5, 15, 40
\sbox	96, 98, 102, 104
\setlength	4, 5, 33, 34, 80
\setmonofont	12
\space	110, 115, 116, 117
\string	57
\subgroup	57, 88
<b>T</b>	
\textcolor	53
\textdir	101
\textsc	54
\textsf	54
\texttt	54, 73, 110
\the	116, 118
<b>U</b>	
\unskip	68
\usebox	115, 117
\UseHook	73, 77, 79, 81, 83, 84
\usepackage	3, 3
<b>V</b>	
\vspace	14, 45, 54
<b>W</b>	
\wd	111, 116, 118
<b>X</b>	
\x	26, 27
<b>Y</b>	
\YHWH	11, 25, 43, 82