

emoji plugin

Support for conveniently working with Unicode emoji in Zsh.

Features

This plugin provides support for working with Unicode emoji characters in `zsh` using human-readable identifiers. It provides global variables which map emoji names to the actual characters, country names to their flags, and some named groupings of emoji. It also provides associated functions for displaying them.

Variables

Variable	Description
<code>\$emoji</code>	Maps emoji names to characters (except flags)
<code>\$emoji_flags</code>	Maps country names to flag characters (using region indicators)
<code>\$emoji_groups</code>	Named groups of emoji. Keys are group names; values are whitespace-separated lists of character names

You may define new emoji groups at run time by modifying `$emoji_groups`. The special group name `all` is reserved for use by the plugin. You should not modify `$emoji` or `$emoji_flags`.

Functions

Function	Description
<code>random_emoji</code>	Prints a random emoji character
<code>display_emoji</code>	Displays emoji, along with their names

Usage and Examples

To output a specific emoji, use:

```
$> echo $emoji[<name>]
```

E.g.:

```
$> echo $emoji[mouse_face]
```

To output a random emoji, use:

```
$> random_emoji
```

To output a random emoji from a particular group, use:

```
$> random_emoji <group>
```

E.g.:

```
$> random_emoji fruits
$> random_emoji animals
$> random_emoji vehicles
$> random_emoji faces
```

The defined group names can be found with `echo ${(k)emoji_groups}` .

To list all available emoji with their names, use:

```
$> display_emoji
$> display_emoji faces
$> display_emoji people
```

To use emoji in a prompt:

```
PROMPT="$emoji[penguin]  > ""
PROMPT='$(random_emoji fruits)  > '
surfer=$emoji[surfer]
PROMPT="$surfer  > "
```

Technical Details

The emoji names and codes are sourced from Unicode Technical Report #51, which provides information on emoji support in Unicode. It can be found at <https://www.unicode.org/reports/tr51/index.html>.

The group definitions are added by this OMZ plugin. They are not based on external definitions.

The values in the `$emoji*` maps are the emoji characters themselves, not escape sequences or other forms that require interpretation. They can be used in any context and do not require escape sequence support from commands like `echo` or `print` .

The emoji in the main `$emoji` map are standalone character sequences which can all be output on their own, without worrying about combining characters. The values may actually be multi-code-point sequences, instead of a single code point, and may include combining characters in those sequences. But they're arranged so their effects do not extend beyond that sequence.

The exception to this is the skin tone / hair style variation selectors. These are included in the main `$emoji` map because they can be displayed on their own, as well as used as combining characters. (If they follow a character that is not one of the emoji characters they combine with, they are displayed as color swatches.)

Experimental Features

This defines some additional variables and functions, but these are experimental and subject to change at any time. You shouldn't rely on them being available. They're mostly for the use of emoji plugin developers to help decide what to include in future revisions.

Variables:

Variable	Description
<code>\$emoji_skintone</code>	Skin tone modifiers (from Unicode 8.0)

Skin Tone Variation Selection

This includes experimental support for the skin tone Variation Selectors introduced with Unicode 8.0, which let you select different skin tones for emoji involving humans.

NOTE: This really is experimental. The skin tone selectors are a relatively new feature and may not be supported by all systems. And the support in this plugin is a work in progress. It may not work in all places. In fact, I haven't gotten it to work anywhere yet. -apjanke

The "variation selectors" are combining characters which change the appearance of the preceding character. A variation selector character can be output immediately following a human emoji to change its skin tone color. You can also output a variation selector on its own to display a color swatch of that skin tone.

The `$emoji_skintone` associative array maps skin tone IDs to the variation selector characters. To use one, output it immediately following a smiley or other human emoji.

```
echo $emoji[waving_hand]$emoji_skintone[5]
```

Note that `$emoji_skintone` is an associative array, and its keys are the *names* of "Fitzpatrick Skin Type" groups, not linear indexes into a normal array. The names are `1_2`, `3`, `4`, `5`, and `6`. (Types 1 and 2 are combined into a single color.) See the [Diversity section in Unicode TR 51](#) for details.

Gemoji support

The [gemoji project](#) seems to be the de facto main source for short names and other emoji-related metadata that isn't included in the official Unicode reports. So, our list of emojis incorporates some of their aliases to make your life more convenient:

```
echo $emoji[grinning_face_with_smiling_eyes]
echo $emoji[smile]
```

These two commands yield the same emoji (😊). The first name is the official one, in the Unicode reference, and the second one is the alias that was in Gemoji's database.

TODO

These are things that could be enhanced in future revisions of the plugin.

- Incorporate CLDR data for ordering and groupings
- Short :bracket: style names (from gemoji)
- ZWJ combining function?