Package: transformerpalettes (via r-universe)

October 28, 2026

Title Color Palettes Inspired by Transformers Toys, Cartoons and Comics

Version 0.4.0.9000

Description Color palettes drawn from the Transformers franchise, centered on the 1984-1989 Generation 1 toys and extending to their Generation 2 re-releases, the cartoon animation models and the Marvel comics. Covers the leaders, every member and combined form of the combiner teams (Aerialbots, Protectobots, Technobots, Constructicons, Stunticons, Combaticons, Terrorcons, Predacons, Seacons, Pretender Monsters and Breastforce), the Dinobots, the Seeker jets, Soundwave and the cassettes, the Insecticons, the 1984 Autobot cars and Mini Vehicles and the 1986 movie cast, with separate palettes for the toy, the cartoon animation model, the Marvel comic coloring and the Generation 2 re-release where each exists. Includes 'ggplot2' color and fill scales for discrete and continuous data, team palettes that take one color from each member of a combiner team, metadata linking each palette to its character, faction, team and reference page, and helpers to shortlist palettes that suit categorical, color-blind-safe, sequential or diverging use. Hex values are approximations of the toys' plastic and paint colors; the package is a fan project and is not affiliated with or endorsed by Hasbro or Takara Tomy.

License MIT + file LICENSE

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 3.5)

Imports ggplot2, grDevices, graphics, stats, utils

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Config/Needs/website jsonlite, knitr, rmarkdown

URL https://bmaitner.github.io/transformerpalettes/,
https://github.com/bmaitner/transformerpalettes

BugReports https://github.com/bmaitner/transformerpalettes/issues

Repository https://bmaitner.r-universe.dev

Date/Publication 2026-10-28 14:39:44 UTC

RemoteUrl https://github.com/bmaitner/transformerpalettes

RemoteRef HEAD

RemoteSha f88d4e4cf4f2bfb121125f28908c2f4aa5b26b35

Contents

scale_tf 2
tf_characters 5
tf_credit 6
tf_find_palettes 7
tf_media 9
tf_palette 9
tf_palette_names 11
tf_palette_stats 11
tf_palettes 12
tf_pick 13
tf_show_palettes 15
tf_team_palette 16

Index **18**

`scale_tf` *Transformers color and fill scales for ggplot2*

Description

Color and fill scales that use a character's palette. The `_d` variants are for discrete data, the `_c` variants for continuous data, and `scale_colour_tf()` / `scale_fill_tf()` choose between the two with the `discrete` argument. `scale_color_*` spellings are aliases.

Usage

```
scale_colour_tf_d(
  palette = "optimus_prime",
  reverse = FALSE,
  medium = "toy",
  ...
)

scale_fill_tf_d(
```

```
  palette = "optimus_prime",
  reverse = FALSE,
  medium = "toy",
  ...
)

scale_colour_tf_c(
  palette = "optimus_prime",
  reverse = FALSE,
  sort = FALSE,
  medium = "toy",
  ...
)

scale_fill_tf_c(
  palette = "optimus_prime",
  reverse = FALSE,
  sort = FALSE,
  medium = "toy",
  ...
)

scale_colour_tf(
  palette = "optimus_prime",
  discrete = TRUE,
  reverse = FALSE,
  sort = FALSE,
  medium = "toy",
  ...
)

scale_fill_tf(
  palette = "optimus_prime",
  discrete = TRUE,
  reverse = FALSE,
  sort = FALSE,
  medium = "toy",
  ...
)

scale_color_tf_d(
  palette = "optimus_prime",
  reverse = FALSE,
  medium = "toy",
  ...
)

scale_color_tf_c(
```

```
  palette = "optimus_prime",
  reverse = FALSE,
  sort = FALSE,
  medium = "toy",
  ...
)

scale_color_tf(
  palette = "optimus_prime",
  discrete = TRUE,
  reverse = FALSE,
  sort = FALSE,
  medium = "toy",
  ...
)
```

Arguments

palette	Name of a character's palette. See tf_palette_names().
reverse	Logical. Reverse the order of colors in the palette?
medium	Which version of the character to use: "toy" (default), "cartoon" or "comic". See tf_media().
...	Additional arguments passed on to ggplot2::discrete_scale() (discrete) or ggplot2::scale_colour_gradientn() / ggplot2::scale_fill_gradientn() (continuous).
sort	Logical, continuous scales only. Order the palette from darkest to lightest before building the gradient, so the scale reads as low-to-high. See tf_find_palettes() for palettes that suit this.
discrete	Logical. If TRUE (default) build a discrete scale, otherwise a continuous gradient scale.

Details

Discrete scales draw the first n colors of the palette, where n is the number of levels in the data. If the data has more levels than the palette has colors, the palette is interpolated with grDevices::colorRampPalette() so that every level still receives a distinct color; expect the result to be less legible than the curated colors. For a palette with one color per member of a combiner team, pass tf_team_palette() to ggplot2::scale_colour_manual().

Value

A ggplot2 scale object.

Examples

```
library(ggplot2)

ggplot(mpg, aes(displ, hwy, colour = drv)) +
```

```
  geom_point(size = 3) +
  scale_colour_tf("optimus_prime")

ggplot(mpg, aes(displ, hwy, colour = drv)) +
  geom_point(size = 3) +
  scale_colour_tf("optimus_prime", medium = "cartoon")

ggplot(diamonds, aes(cut, fill = clarity)) +
  geom_bar() +
  scale_fill_tf_d("bruticus")

ggplot(faithfuld, aes(waiting, eruptions, fill = density)) +
  geom_tile() +
  scale_fill_tf_c("hot_spot", sort = TRUE)
```

`tf_characters` *Transformers character metadata*

Description

Background information for every palette in tf_palettes: the character, which medium the palette records (toy, cartoon, comic or Generation 2 re-release), its faction, the combiner team and combined form it belongs to, its role in that team, its alternate mode, the year of the reference, a description of the colors, and links to the reference page and image on the Transformers Wiki.

Usage

```
tf_characters
```

Format

A data frame with one row per palette and the columns:

palette Palette name, as used by `tf_palette()`. Unique.

base The character's toy palette name, shared by all of that character's palettes across media (e.g. `"optimus_prime"` for both `optimus_prime` and `optimus_prime_cartoon`).

medium `"toy"`, `"cartoon"`, `"comic"` or `"g2"`. See `tf_media()`.

character Character name as printed on the toy's packaging.

faction `"Autobot"` or `"Decepticon"`. See `tf_factions()`.

team Combiner team or subgroup, e.g. `"Aerialbots"` or `"Seekers"`, or `NA` for characters that are not part of one. See `tf_teams()`.

combiner Base palette name of the team's combined form, e.g. `"superion"`, or `NA` for leaders and for subgroups that do not combine.

role One of `"leader"`, `"standalone"` (a character outside any team), `"combiner torso"`, `"combiner limb"`, `"combined form"` or `"subgroup member"` (Dinobots, Seekers, Cassettes, Insecticons, Autobot Cars, Minibots).

alt_mode The character's alternate mode.

year Year of the reference: the original Hasbro release for toy palettes, the character's first appearance for cartoon and comic palettes.

n_colours Number of colors in the palette.

description Description of the colors in that medium.

source URL of the character's page on the Transformers Wiki, with an anchor to the relevant section for cartoon and comic palettes.

image Direct URL of the reference image the palette was checked against (a toy photo, an episode still, or a comic panel), hosted by the Transformers Wiki, or `NA` where none has been identified yet.

Source

Transformers Wiki, https://tfwiki.net/wiki/Main_Page.

Examples

```
head(tf_characters)
table(tf_characters$team, tf_characters$medium)
subset(tf_characters, base == "megatron", c(palette, medium, description))
```

`tf_credit` *Credit the rights holders behind a plot's colors*

Description

Builds a short attribution sentence naming the characters whose palettes were used and where their colors come from (the Hasbro and Takara Tomy toys, the Generation 1 animated series, or the Marvel comics), suitable for a figure caption or an acknowledgments section. Naming the source of a color scheme is ordinary factual attribution; the longer form also states that the colors are approximations and imply no endorsement, which is the wording journals tend to want.

Usage

```
tf_credit(
  x,
  format = c("caption", "acknowledgment", "markdown"),
  package = TRUE,
  medium = "toy"
)
```

Arguments

x	What to credit: a palette name, a character vector of names, or a `tf_palette` object. Objects from `tf_pick()` and `tf_team_palette()` carry their source palettes, so the credit covers every character that contributed a color. Names may carry a medium suffix (`"megatron_cartoon"`) or rely on `medium`.
format	One of: `"caption"` One sentence for a figure caption, e.g. `Colors: Optimus Prime toy colorway by Hasbro a` `"acknowledgment"` Two sentences in the register of an acknowledgments section, naming the sources and stating that the hex values are approximations that imply no endorsement. `"markdown"` The caption with each character linked to its Transformers Wiki reference page, for R Markdown, Quarto or pkgdown captions.
package	Logical. Mention the transformerpalettes package? Set to `FALSE` when the package is cited elsewhere in the text.
medium	Medium for plain palette names; see `tf_media()`.

Value

A single character string.

See Also

tf_characters for the underlying metadata; `citation("transformerpalettes")` for the package reference.

Examples

```
tf_credit("optimus_prime")
tf_credit(c("optimus_prime", "megatron_cartoon", "megatron_comic"))
tf_credit(tf_team_palette("Stunticons"))
tf_credit(tf_pick(c("optimus_prime", "megatron"), n = 4))
tf_credit("optimus_prime", format = "acknowledgment")
cat(tf_credit(c("optimus_prime", "megatron"), format = "markdown"))

library(ggplot2)
ggplot(mpg, aes(displ, hwy, color = drv)) +
  geom_point(size = 2.5) +
  scale_color_tf("optimus_prime") +
  labs(caption = tf_credit("optimus_prime"))
```

`tf_find_palettes` *Find Transformers palettes that suit a plotting use case*

Description

Filters the palettes to those whose colors work for a given job, judged from the perceptual statistics in `tf_palette_stats()`. Many toys look great on a figure but poor on a chart; this surfaces the ones that hold up.

Usage

```
tf_find_palettes(
  use = c("categorical", "colorblind", "sequential", "diverging"),
  faction = NULL,
  team = NULL,
  role = NULL,
  medium = "toy",
  min_colours = 4
)
```

Arguments

`use` One of:

- `"categorical"` Every pair of colors is clearly distinguishable (minimum Lab distance of 20 or more). Good for unordered groups.
- `"colorblind"` Categorical, and still distinguishable under simulated deuteranopia, protanopia and tritanopia (minimum distance 15 or more in each).
- `"sequential"` One hue family (hue spread of 45 degrees or less among the chromatic colors) spanning at least 50 units of lightness, so a continuous gradient reads as low-to-high. Pair with `sort = TRUE` in `tf_palette()` or the `_c` scales.
- `"diverging"` Exactly two distinct hue families plus a light or neutral color to sit between them. The curated order is not necessarily dark-A, light, dark-B; check `tf_palette_stats()` and reorder the colors yourself for a continuous diverging scale.

The thresholds are deliberately simple heuristics computed by `tf_palette_stats()`; treat the result as a shortlist to look at with `tf_show_palettes()`, not a guarantee.

`faction`, `team`, `role`, `medium`
Optional filters passed to `tf_palette_names()`. `medium` defaults to `"toy"`; use `"all"` to search cartoon and comic palettes too.

`min_colours` Minimum number of colors the palette must have.

Value

A character vector of palette names, ordered so that the strongest candidates for the use case come first.

Examples

```
tf_find_palettes("colorblind")
tf_find_palettes("diverging", faction = "Decepticon")
tf_find_palettes("sequential")
tf_find_palettes("colorblind", medium = "cartoon")
```

tf_media — *List the media, factions and combiner teams in transformerpalettes*

Description

List the media, factions and combiner teams in transformerpalettes

Usage

```
tf_media()

tf_factions()

tf_teams(faction = NULL)
```

Arguments

`faction`	Optional faction to restrict `tf_teams()` to; see `tf_factions()`.

Value

`tf_media()` returns the media a palette can record, in the order `"toy"` (the original Generation 1 release), `"cartoon"` (the animation model), `"comic"` (Marvel) and `"g2"` (the Generation 2 re-release of the toy in new colors, 1992-1994). `tf_factions()` returns a character vector of faction names. `tf_teams()` returns a character vector of team and subgroup names, in package order (Autobot teams first). All can be passed to the matching argument of `tf_palette_names()`, `tf_show_palettes()` and `tf_find_palettes()`.

Examples

```
tf_media()
tf_factions()
tf_teams()
tf_teams(faction = "Decepticon")
```

tf_palette — *Get colors from a character's palette*

Description

Returns `n` colors from the named character's palette, either the first `n` curated colors (`type = "discrete"`) or `n` colors interpolated across the whole palette (`type = "continuous"`).

Usage

```
tf_palette(
  name,
  n = NULL,
  type = c("discrete", "continuous"),
  strict = TRUE,
  sort = FALSE,
  medium = "toy"
)
```

Arguments

name	Palette name. Case-insensitive, and spaces or hyphens may stand in for underscores, so "Optimus Prime" and "Hun-Gurrr" work. See tf_palette_names(). A full name with a medium suffix ("megatron_cartoon") also works and takes precedence over medium.
n	Number of colors wanted. Defaults to every color in the palette.
type	Either "discrete" (default) to return the curated colors as they are, or "continuous" to interpolate n colors across the palette with grDevices::colorRampPalette().
strict	Logical. For discrete palettes, error when n exceeds the number of curated colors (default). If FALSE, interpolate to n colors instead.
sort	Logical. If TRUE, order the colors from darkest to lightest (by CIE lightness) before selecting or interpolating. Useful when a palette is used as a sequential gradient; see tf_find_palettes().
medium	Which version of the character to use: "toy" (default), "cartoon", "comic" or "g2". See tf_media(). Errors with a pointer to the wishlist if the character has no palette in that medium.

Value

A character vector of hex colors with class "tf_palette", which has a print method that draws the swatches.

Examples

```
tf_palette("optimus_prime")
tf_palette("optimus_prime", medium = "cartoon")
tf_palette("Megatron", 3, medium = "comic")
tf_palette("devastator", medium = "g2")
tf_palette("devastator", 12, type = "continuous")
tf_palette("hot_spot", 12, type = "continuous", sort = TRUE)

# base graphics
barplot(1:5, col = tf_palette("predaking"))
```

tf_palette_names *List the available palettes*

Description

List the available palettes

Usage

```
tf_palette_names(faction = NULL, team = NULL, role = NULL, medium = "toy")
```

Arguments

faction	Optional faction (see tf_factions()) to restrict the list to. Matched case-insensitively and by unique prefix, so "auto" works.
team	Optional team or subgroup (see tf_teams()) to restrict the list to, matched the same way.
role	Optional role to restrict the list to: one of "leader", "combiner torso", "combiner limb", "combined form" or "subgroup member", matched the same way.
medium	Which medium's palettes to list: "toy" (default), "cartoon", "comic", "g2", or "all" for every palette regardless of medium. See tf_media().

Value

A character vector of palette names accepted by tf_palette() and the scale_*_tf() functions.

Examples

```
tf_palette_names()
tf_palette_names(faction = "Decepticon")
tf_palette_names(team = "Aerialbots", medium = "cartoon")
tf_palette_names(role = "leader", medium = "all")
```

tf_palette_stats *Perceptual statistics for every Transformers palette*

Description

Computes, from the hex colors alone, how suitable each palette is for the common plotting jobs. Used by tf_find_palettes(); exposed so you can apply your own thresholds.

Usage

```
tf_palette_stats(names = tf_palette_names(medium = "all"))
```

Arguments

`names`	Palette names to compute for. Defaults to every palette in every medium.

Details

Distances are CIE76 (Euclidean in CIE Lab). Color-vision deficiency is simulated with the Machado, Oliveira and Fernandes (2009) matrices at full severity for deuteranopia, protanopia and tritanopia. Hue statistics only consider colors with chroma of at least 20, so grays and blacks do not count as a hue.

Value

A data frame with one row per palette and the columns:

palette Palette name.

n_colours Number of colors.

min_de Smallest pairwise Lab distance between any two colors. Larger is easier to tell apart; 20 is a comfortable floor for categorical use.

min_de_deutan, min_de_protan, min_de_tritan The same after simulating each color-vision deficiency.

lum_range Range of CIE lightness (0 to 100) across the palette.

hue_spread Angular spread of the chromatic colors' hues, in degrees. Near 0 means one hue family.

hue_clusters Number of distinct hue families (hues separated by more than 60 degrees).

has_light Whether the palette contains a light or neutral color that can serve as a midpoint.

categorical, colorblind, sequential, diverging Logical flags as used by `tf_find_palettes()`.

Examples

```
s <- tf_palette_stats()
head(s[order(-s$min_de_deutan), c("palette", "min_de", "min_de_deutan")])
```

`tf_palettes` *Transformers palettes*

Description

A named list of color palettes covering the Transformers cast from the 1984 Generation 1 toys onward: the leaders, every member and combined form of the combiner teams, the Dinobots, Seekers, cassettes, Insecticons, the 1984 Autobot cars and Mini Vehicles, and the 1986 movie cast. Each character has a palette for the original toy, most have one for the cartoon, many have one for the Marvel comic, and the three combiner teams Hasbro re-released in Generation 2 have one for that; see the `medium` column of tf_characters. Each element is a character vector of four to six hex colors ordered so that the first few are the most distinct from one another (body color first, then secondary color, then accents). Use `tf_palette()` to pull colors from a palette, `tf_team_palette()` to take one color from each member of a combiner team, and tf_characters for background on each character.

Usage

```
tf_palettes
```

Format

A named list of character vectors of hex colors.

Details

Toy palettes are named after the character (optimus_prime); cartoon and comic palettes carry a suffix (optimus_prime_cartoon, optimus_prime_comic). Hex values are approximations: toy colors were curated by eye from product photography on the Transformers Wiki (https://tfwiki.net/wiki/Main_Page) and the toys' published color descriptions, cartoon colors from the animation models as they appear in episode stills, and comic colors from the character's early Marvel appearances. Chrome and die-cast parts do not reduce to a single flat color; those entries reflect how the part photographs.

See Also

tf_palette(), tf_palette_names(), tf_media(), tf_teams(), tf_characters, tf_show_palettes()

Examples

```
length(tf_palettes)
tf_palettes$optimus_prime
tf_palettes$optimus_prime_cartoon
tf_palettes$devastator
```

---

tf_pick — *Pick the most distinguishable colors from one or more palettes*

Description

Pools the colors of the named palettes and selects the n that are hardest to confuse with one another on a given kind of plot: light background, dark background, or output that will be read in grayscale. Optionally judges distinctness under simulated color-vision deficiency. Use it when a single character's palette does not have enough usable colors, or when you want a team's or a faction's colors but only the ones that hold up on a chart.

Usage

```
tf_pick(
  palettes,
  n = 5,
  style = c("light", "dark", "grayscale"),
  colorblind = FALSE,
  medium = "toy",
  background = NULL,
```

```
  min_contrast = 3
)
```

Arguments

palettes	Character vector of one or more palette names; see `tf_palette_names()`. Team names are not accepted; use `tf_palette_names(team = ...)` to expand a team.
n	Number of colors wanted.
style	One of `"light"` (white background), `"dark"` (near-black background) or `"grayscale"` (white background, judged by lightness alone).
colorblind	Logical. Require the colors to stay distinct under simulated deuteranopia, protanopia and tritanopia?
medium	Medium for plain palette names; see `tf_media()`.
background	Optional background color overriding the one implied by `style`, as a hex string.
min_contrast	Minimum WCAG 2 contrast ratio a color must have against the background to be considered. Set to 1 to disable the filter.

Details

The selection runs in three steps.

1. **Pool.** The colors of every palette in `palettes` are combined, with duplicates removed. Names may be plain (`"optimus_prime"`) or carry a medium suffix (`"optimus_prime_cartoon"`); plain names use `medium`.
2. **Filter for the background.** Colors whose WCAG 2 contrast ratio against the background is below `min_contrast` are dropped, so a light-mode selection never contains a near-white and a dark-mode selection never contains a near-black. The background is white for `"light"` and `"grayscale"` and near-black (`"#121212"`) for `"dark"` unless `background` is given. The default of 3 is the WCAG minimum for graphical objects; lower it to keep more colors, raise it for text.
3. **Choose.** Every pair of remaining colors is scored by its distance in CIE Lab (CIE76). With `colorblind = TRUE` the distance is the smallest of the normal-vision distance and the distances after simulating deuteranopia, protanopia and tritanopia (Machado, Oliveira and Fernandes 2009), so two colors only count as distinct if every viewer can tell them apart. With `style = "grayscale"` only the lightness difference counts, so the colors survive conversion to gray. The `n` colors whose closest pair is farthest apart are chosen (a maximum-dispersion subset, found exhaustively when the pool is small and by greedy seeding plus swap search otherwise), then ordered so that each color is as different as possible from those before it.

The result is a categorical palette: it does not try to build a gradient. The smallest pairwise distance in the selection is returned as the `min_distance` attribute; values above about 20 are comfortably distinct, and below about 12 two colors will be hard to tell apart. If the pool cannot supply `n` colors after filtering, the function errors and says how many it had, so add palettes or lower `min_contrast`.

Value

A character vector of `n` hex colors with class `"tf_palette"`, named by the palette each color came from, and with attributes `min_distance` (smallest pairwise distance under the chosen criterion), `style`, `colorblind` and `background`.

See Also

tf_find_palettes() to shortlist whole palettes instead; tf_palette_stats() for the underlying measures.

Examples

```
# five colors for a white-background plot, from the two leaders
tf_pick(c("optimus_prime", "megatron"), n = 5)

# the same for a dark theme, and color-blind safe
tf_pick(c("optimus_prime", "megatron"), n = 4, style = "dark", colorblind = TRUE)

# a whole team, cartoon colors, printable in grayscale
tf_pick(tf_palette_names(team = "Combaticons"), n = 4,
        style = "grayscale", medium = "cartoon")

library(ggplot2)
cols <- tf_pick(tf_palette_names(team = "Predacons"), n = 3, colorblind = TRUE)
ggplot(mpg, aes(displ, hwy, colour = drv)) +
  geom_point(size = 2.5) +
  scale_colour_manual(values = unname(cols)) +
  theme_minimal()
```

---

tf_show_palettes *Display Transformers palettes*

Description

Draws a swatch strip for each palette on the current graphics device, labeled with the palette name and, by default, the hex code under each swatch. Useful for picking a character or for reporting a color that looks wrong. With no arguments every toy palette is drawn, which is a lot; filter by faction, team, role or use to narrow it down, or pass a character's palettes across media to compare them.

Usage

```
tf_show_palettes(
  names = NULL,
  faction = NULL,
  team = NULL,
  role = NULL,
  use = NULL,
  medium = "toy",
  ncol = 3,
  hex = TRUE
)
```

Arguments

`names`	Optional character vector of palette names to show. Defaults to every palette matching the other filters.
`faction`	Optional faction to restrict to; see `tf_factions()`.
`team`	Optional combiner team to restrict to; see `tf_teams()`.
`role`	Optional role to restrict to; see `tf_palette_names()`.
`use`	Optional use case to restrict to; see `tf_find_palettes()`.
`medium`	Which medium's palettes to show: `"toy"` (default), `"cartoon"`, `"comic"` or `"all"`. Ignored when `names` is given.
`ncol`	Number of columns to arrange the swatch strips in.
`hex`	Logical. Print the hex code under each swatch?

Value

Invisibly, the character vector of palette names that were drawn.

Examples

```
tf_show_palettes(team = "Aerialbots")
tf_show_palettes(team = "Aerialbots", medium = "cartoon")
tf_show_palettes(faction = "Decepticon", role = "combined form", ncol = 2)
tf_show_palettes(use = "colorblind", ncol = 4)
tf_show_palettes(c("megatron", "megatron_cartoon", "megatron_comic"), ncol = 1)
```

`tf_team_palette` *One color per member of a team*

Description

Builds a categorical palette for a combiner team or subgroup by taking one color from each member, in the team's order (torso first, then limbs, for a combiner team). Each color is named after the member it came from, so the palette doubles as a legend.

Usage

```
tf_team_palette(
  team,
  which = 1L,
  distinct = FALSE,
  combined = FALSE,
  medium = "toy"
)
```

Arguments

team	Combiner team name; see tf_teams(). Matched case-insensitively and by unique prefix.
which	Which color to take from each member's palette when distinct = FALSE. 1 (the default) is the body color; 2 the secondary color, and so on. Members with fewer colors than which fall back to their last color.
distinct	Logical. Choose each member's color to be as different as possible from those already chosen? See Details.
combined	Logical. Include the team's combined form as the last color?
medium	Which version of the team to use: "toy" (default), "cartoon" or "comic". Errors if any member lacks a palette in that medium.

Details

By default the body color of each member is used, which is faithful but collapses for teams cast from shared plastics: every Constructicon is the same green, three Protectobots are white, and so on. With distinct = TRUE each member instead contributes the color from its palette that is furthest (in CIE Lab) from the colors already chosen for the members before it, so the palette stays legible while every color is still one the member actually wears. The Stunticons, Combaticons and Terrorcons are distinct either way; for the others distinct = TRUE is the useful setting. Check the result with tf_palette_stats() on the returned colors if legibility matters.

Value

A named character vector of hex colors with class "tf_palette".

Examples

```
tf_team_palette("Stunticons")
tf_team_palette("Seekers")
tf_team_palette("Aerialbots")
tf_team_palette("Aerialbots", distinct = TRUE)
tf_team_palette("Predacons", which = 2)
tf_team_palette("Combaticons", medium = "cartoon")

library(ggplot2)
ggplot(mpg, aes(displ, hwy, colour = drv)) +
  geom_point() +
  scale_colour_manual(values = unname(tf_team_palette("Stunticons")))
```

Index

∗ **datasets**
    tf_characters, 5
    tf_palettes, 12

ggplot2::discrete_scale(), *4*
ggplot2::scale_colour_gradientn(), *4*
ggplot2::scale_colour_manual(), *4*
ggplot2::scale_fill_gradientn(), *4*
grDevices::colorRampPalette(), *4*, *10*

scale_color_tf *(*scale_tf*)*, 2
scale_color_tf_c *(*scale_tf*)*, 2
scale_color_tf_d *(*scale_tf*)*, 2
scale_colour_tf *(*scale_tf*)*, 2
scale_colour_tf_c *(*scale_tf*)*, 2
scale_colour_tf_d *(*scale_tf*)*, 2
scale_fill_tf *(*scale_tf*)*, 2
scale_fill_tf_c *(*scale_tf*)*, 2
scale_fill_tf_d *(*scale_tf*)*, 2
scale_tf, 2

tf_characters, 5, *7*, *12, 13*
tf_credit, 6
tf_factions *(*tf_media*)*, 9
tf_factions(), *5*, *9*, *11*, *16*
tf_find_palettes, 7
tf_find_palettes(), *4*, *9–12*, *15, 16*
tf_media, 9
tf_media(), *4, 5*, *7*, *10, 11*, *13, 14*
tf_palette, 9
tf_palette(), *5*, *8*, *11–13*
tf_palette_names, 11
tf_palette_names(), *4*, *8–10*, *13, 14*, *16*
tf_palette_stats, 11
tf_palette_stats(), *7, 8*, *15*, *17*
tf_palettes, *5*, 12
tf_pick, 13
tf_pick(), *7*
tf_show_palettes, 15
tf_show_palettes(), *8, 9*, *13*
tf_team_palette, 16
tf_team_palette(), *4*, *7*, *12*
tf_teams *(*tf_media*)*, 9
tf_teams(), *5*, *11*, *13*, *16, 17*