Package: scratchy (via r-universe)

November 27, 2026

Title Hand-Drawn Charts, Drawings and Slides in the 'Excalidraw' Style

Version 0.1.0

Description Build, read and write 'Excalidraw' scenes from R, turn 'ggplot2' and base graphics into editable hand-drawn drawings, and render them offline with the same 'rough.js' engine and handwriting fonts that 'Excalidraw' uses. Includes the 'Excalidraw' editor as an 'htmlwidget'. Not affiliated with or endorsed by Excalidraw.

License MIT + file LICENSE

URL https://jossammil.github.io/scratchy/, https://github.com/jossammil/scratchy

BugReports https://github.com/jossammil/scratchy/issues

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Depends R (>= 4.1)

Imports cli, ggplot2 (>= 3.5.0), graphics, grDevices, grid, htmltools, htmlwidgets, jsonlite, rlang, stats, systemfonts (>= 1.3.0), textshaping (>= 1.0.5), tools, utils, svglite (>= 2.2.2), V8

Suggests bslib, chromote, gifski, httpuv, jpeg, knitr, magick, miniUI, officer, png, ragg (>= 1.5.0), rmarkdown, rsvg, rvg, shiny, testthat (>= 3.0.0), vdiffr, withr, xml2

Config/testthat/edition 3

VignetteBuilder knitr

Config/pak/sysreqs cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev make libharfbuzz-dev libicu-dev libpng-dev libuv1-dev libssl-dev libnode-dev

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

Date/Publication 2026-11-27 09:24:13 UTC

RemoteUrl https://github.com/jossammil/scratchy

RemoteRef HEAD

RemoteSha 3aea3020d94ba24af7b4e6c5af40b638cfd9a6ed

Contents

annotations . . . 3
as.data.frame.exc_scene . . . 5
as_excalidraw . . . 5
exc_add . . . 8
exc_animate . . . 8
exc_bbox . . . 10
exc_color . . . 10
exc_d2 . . . 11
exc_data_to_px . . . 12
exc_edit . . . 13
exc_font_fetch . . . 13
exc_fonts . . . 14
exc_frame . . . 15
exc_grob . . . 16
exc_group . . . 17
exc_image . . . 17
exc_install_fonts . . . 18
exc_library . . . 19
exc_line . . . 19
exc_mermaid . . . 21
exc_modify . . . 22
exc_papers . . . 22
exc_peep . . . 23
exc_pptx . . . 24
exc_preset . . . 25
exc_rect . . . 26
exc_restyle . . . 27
exc_save . . . 28
exc_scene . . . 29
exc_scene_from_json . . . 30
exc_set_style . . . 30
exc_slides_pdf . . . 31
exc_stickers . . . 31
exc_style . . . 33
exc_template_example . . . 34
exc_template_fill . . . 34
exc_text . . . 35
exc_translate . . . 37
excalidraw . . . 37
excalidraw_colors . . . 38
excalidraw_proxy . . . 39
excalidrawOutput . . . 41
read_excalidraw . . . 41
read_excalidraw_lib . . . 42
register_fonts . . . 43
run_sketch_studio . . . 43

scale_fill_excalidraw 44
sketch 45
theme_excalidraw 46
use_excalidraw_slides 47
write_excalidraw 48

Index **49**

`annotations` *Hand-drawn annotations*

Description

Add sticky notes, callouts, marks and titles to a scene. Positions use the plot's data values for scenes made from a ggplot (see `exc_data_to_px()`), or px when wrapped in `I()`.

Usage

```
exc_note(
  scene,
  text,
  x,
  y,
  to = NULL,
  width = 200,
  fill = "yellow",
  angle = -2,
  panel = 1,
  ...
)

exc_callout(scene, text, x, y, dx = 60, dy = -60, panel = 1, ...)

exc_mark(
  scene,
  x,
  y,
 type = c("circle", "box", "underline", "highlight", "strike", "bracket", "cross"),
  color = NULL,
  padding = 6,
  panel = 1,
  ...
)

exc_title(scene, title, subtitle = NULL, size = 36, underline = TRUE, ...)
```

Arguments

scene	An exc_scene.
text	The text.
x, y	Where to put it: data values, or px in I(). For exc_mark(), give one value to mark a point or category, or two values for a range.
to	Optional c(x, y) (or list(x, y)) the note's arrow points at.
width	Maximum width of the note in px before the text wraps.
fill	Note colour (an Excalidraw hue name or a colour).
angle	Tilt in degrees, for a casually stuck-on look.
panel	Panel number in faceted plots.
...	Style properties passed to the elements, such as stroke, font, font_size or roughness.
dx, dy	Offset in px from the spot to the text.
type	"circle", "box", "underline", "highlight", "strike", "bracket" or "cross".
color	Mark colour. Highlights default to a translucent yellow.
padding	Space in px between the marked area and the mark.
title, subtitle	Title and optional subtitle text.
size	Title font size in px (Excalidraw's XL is 36).
underline	Draw a hand-drawn underline under the title.

Details

- exc_note(): a sticky note, optionally with an arrow pointing to a spot.
- exc_callout(): a handwritten remark with a curved arrow to a spot.
- exc_mark(): circle, box, underline, highlight, strike through, bracket or cross out a value or a range, like rough-notation.
- exc_title(): a big title with a hand-drawn underline, above the drawing.

Value

The scene with the annotation added.

Examples

```
library(ggplot2)
p <- ggplot(mpg, aes(class)) + geom_bar(fill = "#a5d8ff")
as_excalidraw(p) |>
  exc_title("What do people drive?") |>
  exc_mark(x = "suv", y = c(0, 62), type = "box") |>
  exc_note("SUVs win!", x = "midsize", y = 55, to = c("suv", 50)) |>
  exc_callout("so few", x = "2seater", y = 5)
```

as.data.frame.exc_scene *Tidy view of a scene's elements*

Description

One row per element with its main properties. Useful for finding elements to change with `exc_modify()`.

Usage

```
## S3 method for class 'exc_scene'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`x`	An `exc_scene`.
`row.names, optional`	Ignored.
`...`	Ignored.

Value

A data frame.

as_excalidraw *Turn a plot into an editable Excalidraw scene*

Description

Draws the plot with `svglite::svglite()`, reads back every line, box, point, polygon and piece of text, and rebuilds them as Excalidraw elements: bars become hachure-filled rectangles, lines become sketchy lines, text uses a handwriting font. The result can be saved with `write_excalidraw()` and edited by hand in Excalidraw, or drawn in R with `exc_save()` / `plot()`.

Usage

```
as_excalidraw(x, ...)

## S3 method for class 'ggplot'
as_excalidraw(
  x,
  width = 800,
  height = 500,
  theme = theme_excalidraw(),
  fill_style = "hachure",
```

```
  roughness = "artist",
  outline = TRUE,
  font = "Excalifont",
  snap = TRUE,
  preset = NULL,
  seed = 1,
  max_elements = 5000,
  id = NULL,
  ...
)

## S3 method for class 'formula'
as_excalidraw(
  x,
  width = 800,
  height = 500,
  fill_style = "hachure",
  roughness = "artist",
  outline = TRUE,
  font = "Excalifont",
  snap = TRUE,
  preset = NULL,
  seed = 1,
  max_elements = 5000,
  id = NULL,
  ...
)

## S3 method for class '`function`'
as_excalidraw(
  x,
  width = 800,
  height = 500,
  fill_style = "hachure",
  roughness = "artist",
  outline = TRUE,
  font = "Excalifont",
  snap = TRUE,
  preset = NULL,
  seed = 1,
  max_elements = 5000,
  id = NULL,
  ...
)

## S3 method for class 'grob'
as_excalidraw(
  x,
```

```
  width = 800,
  height = 500,
  fill_style = "hachure",
  roughness = "artist",
  outline = TRUE,
  font = "Excalifont",
  snap = TRUE,
  preset = NULL,
  seed = 1,
  max_elements = 5000,
  id = NULL,
  ...
)
```

Arguments

x	A ggplot, a formula such as ~ hist(x) or a function that draws with base graphics, or a grid grob.
...	Unused.
width, height	Size of the drawing in px (Excalidraw units, 96 per inch).
theme	For ggplots, a theme added before drawing. Defaults to theme_excalidraw(); use NULL to keep the plot's own theme.
fill_style	Fill style for filled shapes: "hachure", "cross-hatch", "solid" or "zigzag". Points and backgrounds are always solid.
roughness	Sloppiness: "architect", "artist", "cartoonist" or a number.
outline	Give filled shapes without a border a darker outline in the same hue, like Excalidraw shapes.
font	Font for all text (see exc_fonts()); NULL keeps the fonts the plot used.
snap	Round stroke widths to Excalidraw's thin / bold / extra bold.
preset	Optional exc_preset() to restyle with afterwards.
seed	Seed for the hand-drawn wobble, so the same plot always looks the same.
max_elements	Warn when the scene has more elements than this; Excalidraw slows down with very many.
id	A name for the chart. All its elements are grouped and tagged with it, so the chart can be found later, for example to swap it for a new version with exc_proxy_replace_chart(). Random if NULL.

Details

For ggplot2 plots the scene also remembers how data values map to positions, so annotations such as exc_note() and exc_mark() can be placed with data coordinates.

Value

An exc_scene.

Examples

```
library(ggplot2)
p <- ggplot(mpg, aes(class, fill = drv)) + geom_bar() + scale_fill_excalidraw()
s <- as_excalidraw(p)
s
write_excalidraw(s, tempfile(fileext = ".excalidraw"))

as_excalidraw(~ hist(faithful$eruptions, col = "#ffec99", main = "Old Faithful"))
```

---

exc_add *Add elements to a scene*

Description

Add elements to a scene

Usage

```
exc_add(scene, ...)
```

Arguments

scene	An exc_scene.
...	Elements to add on top of the existing ones.

Value

The updated scene.

Examples

```
exc_scene() |> exc_add(exc_text("Hello", 0, 0))
```

---

exc_animate *Animate a scene drawing itself*

Description

Saves a scene as an animated SVG in which the drawing sketches itself: elements appear one after another in z-order, each stroke traced from start to end (CSS `stroke-dasharray` animations), fills fade in once their outline is done, and text is written in letter by letter. The strokes are the native renderer's, so the finished drawing matches `exc_save()`. Save to a `.gif` file to record the animation in headless Chrome instead.

Usage

```
exc_animate(
  scene,
  file,
  duration = 5,
  loop = FALSE,
  hold = 2,
  text = c("write", "fade"),
  padding = 10,
  background = TRUE,
  paper = NULL,
  frames = TRUE,
  scale = 1,
  fps = 20
)
```

Arguments

scene	An exc_scene.
file	Output path ending in .svg or .gif. GIFs are recorded in Chrome or Chromium (through the chromote package) and put together with the gifski or magick package.
duration	Seconds from blank canvas to finished drawing. Each element gets time in proportion to how much it has to draw.
loop	Replay forever, showing the finished drawing for hold seconds between plays. FALSE plays once and keeps the drawing.
hold	Seconds the finished drawing stays up before a loop starts again.
text	How text appears: "write" letter by letter, "fade" each text element as a whole.
padding	Space around the drawing, in px.
background	Draw the background colour (FALSE = transparent).
paper	Optional paper background; see exc_papers(). Defaults to the paper set by exc_restyle(), if any.
frames	Draw frame outlines and names.
scale	Size: 1 = one pixel per scene px. For GIFs this is the pixel density, as in exc_save().
fps	Frames per second for GIFs (at most 50).

Details

The SVG plays in web browsers, also inside <img> tags, slides and README files. Viewers who ask their system for reduced motion, and programs that don't run CSS animations, see the finished drawing.

Value

file, invisibly.

Examples

```
s <- exc_scene(
  exc_rect(0, 0, 160, 80, label = "Idea", fill = "yellow"),
  exc_arrow(c(170, 270), c(40, 40)),
  exc_ellipse(280, 0, 150, 80, label = "Chart", fill = "blue", fill_style = "hachure")
)
f <- tempfile(fileext = ".svg")
exc_animate(s, f, duration = 3)
## Not run:
browseURL(f)
# A looping GIF (needs Chrome and the gifski or magick package)
exc_animate(s, "idea.gif", loop = TRUE)

## End(Not run)
```

---

exc_bbox *Bounding box of a scene*

Description

Bounding box of a scene

Usage

```
exc_bbox(scene, stroke = TRUE)
```

Arguments

scene	An exc_scene or exc_elements.
stroke	Include half the stroke width around each element.

Value

A named numeric vector: xmin, ymin, xmax, ymax, width, height, in px.

---

exc_color *Look up a colour by Excalidraw hue name*

Description

Look up a colour by Excalidraw hue name

Usage

```
exc_color(hue, shade = 5)
```

Arguments

hue	One of "black", "white", "transparent", or a hue such as "blue", "red" or "bronze".
shade	Shade from 1 (lightest) to 5 (strongest).

Value

A hex colour string.

Examples

```
exc_color("blue")
exc_color("yellow", 2)
```

---

exc_d2 *Diagrams from D2 in sketch mode*

Description

Renders D2 diagram text with D2's hand-drawn `--sketch` mode and returns it as a scene holding the SVG as an image (so you can annotate it, place it next to charts or save it). Needs the `d2` command-line tool (on macOS: `brew install d2`). For diagrams made of editable Excalidraw shapes, use `exc_mermaid()` instead.

Usage

```
exc_d2(
  code,
  sketch = TRUE,
  font = "Excalifont",
  theme = NULL,
  layout = NULL,
  pad = 20,
  scale = 1
)
```

Arguments

code	D2 diagram text, such as `"idea -> sketch -> chart"`.
sketch	Use D2's hand-drawn style.
font	Font for the labels (see `exc_fonts()`), or `NULL` for D2's default (Source Sans Pro).
theme	D2 theme id (see `d2 themes`).
layout	Layout engine, such as `"dagre"` or `"elk"`.
pad	Padding around the diagram, in px.
scale	Size multiplier for the image.

Details

The labels use a handwriting font (Excalifont by default), embedded in the SVG so the diagram looks the same in the Excalidraw editor, in browsers and in SVG exports. R's own PNG and PDF output draws SVG images with rsvg, which ignores embedded fonts: there the labels use the font only if it is installed on the computer (see `exc_install_fonts()`).

Value

An `exc_scene`.

Examples

```
## Not run:
exc_d2("data -> sketch: ggplot2\nsketch -> share: excalidraw") |> plot()

## End(Not run)
```

`exc_data_to_px` *Map data values to positions in a converted plot*

Description

Scenes made by `as_excalidraw()` from a ggplot remember how data values map to px. Discrete values such as `"suv"` map to the centre of their category. Wrap values in `I()` to give px directly.

Usage

```
exc_data_to_px(scene, x, y, panel = 1)
```

Arguments

`scene`	An `exc_scene` made from a ggplot.
`x, y`	Data values (or px wrapped in `I()`).
`panel`	Panel number for faceted plots, in layout order.

Value

A list with `x` and `y` in px.

exc_edit *Edit a scene by hand in the Excalidraw editor*

Description

Opens the scene in the real Excalidraw editor (in the RStudio viewer or a browser). Draw, move things around, add notes, then click **Done** to get the edited scene back in R.

Usage

```
exc_edit(
  scene = exc_scene(),
  viewer = shiny::paneViewer(minHeight = 650),
  title = "Edit drawing"
)
```

Arguments

scene	An exc_scene, or anything as_excalidraw() accepts.
viewer	Where to open the editor; see shiny::runGadget().
title	Title shown above the editor.

Value

The edited exc_scene, or the original if nothing was changed. NULL (invisibly) if you click Cancel.

Examples

```
if (interactive()) {
  s <- exc_edit(exc_scene(exc_text("Draw something!", 0, 0, font_size = 28)))
  write_excalidraw(s, "drawing.excalidraw")
}
```

exc_font_fetch *Download extra fonts*

Description

Some fun fonts can't be bundled because of their licence, so they are downloaded on request, stored in your user data folder and registered for this and future sessions.

Usage

```
exc_font_fetch(name = c("xkcd"))
```

Arguments

name	Which font to download.

Details

- "xkcd": the xkcd Script handwriting font by the iPython team, made from Randall Munroe's handwriting (CC BY-NC 3.0: non-commercial use).

Value

The font's family name, invisibly. Use it as font =.

Examples

```
## Not run:
exc_font_fetch("xkcd")
sketch(p, preset = "xkcd", font = "xkcd Script")

## End(Not run)
```

---

exc_fonts *Fonts available to scratchy*

Description

Lists Excalidraw's own fonts and the extra handwriting fonts bundled with the package. Fonts without an Excalidraw id still work in R; when the file is opened in Excalidraw they display as Excalifont.

Usage

```
exc_fonts()
```

Value

A data frame with the font name, its Excalidraw id, whether a font file is bundled, and the licence.

Examples

```
exc_fonts()
```

exc_frame *Frames*

Description

A frame is a named box that holds other elements. Excalidraw uses frames for slides and for exporting part of a drawing.

Usage

```
exc_frame(
  x,
  y,
  width,
  height,
  name = NULL,
  contents = NULL,
  id = NULL,
  custom = NULL
)
```

Arguments

x, y, width, height	Position and size in px.
name	Frame title shown above it.
contents	Elements to place inside the frame.
id	Element id. Generated if NULL.
custom	A list stored in the element's customData.

Value

An exc_elements object: the contents followed by the frame.

Examples

```
exc_frame(0, 0, 400, 300, "Slide 1", contents = exc_text("Hi", 20, 20))
```

exc_grob *Draw a scene with grid*

Description

Builds a grid grob that draws the scene the way Excalidraw does: rough.js strokes and fills from each element's seed, Excalidraw's arrowheads, and text in the bundled handwriting fonts. The sketchy geometry is computed once; the grob rescales to whatever space it is drawn in, keeping the aspect ratio.

Usage

```
exc_grob(
  scene,
  padding = 10,
  background = TRUE,
  text = c("auto", "outline", "font"),
  scale = NULL,
  frames = TRUE,
  paper = NULL,
  name = NULL
)
```

Arguments

scene	An exc_scene.
padding	Space around the drawing, in px (Excalidraw uses 10).
background	Draw the scene's background colour.
text	How to draw text: "outline" turns glyphs into shapes (works on every device), "font" uses the registered font (needs ragg or svglite), "auto" picks "font" on those devices and "outline" elsewhere.
scale	NULL to fit the drawing to the viewport, or a number of device pixels (at 96 dpi) per scene px for a fixed size.
frames	Draw frame outlines and names.
paper	Optional paper background: "graph", "dots", "lines", "kraft", "blueprint" or "chalkboard". Defaults to the paper set by exc_restyle(), if any.
name	Grob name.

Value

A grob of class exc_grob.

Examples

```
s <- exc_scene(exc_rect(0, 0, 120, 60, label = "hi", fill = "yellow", fill_style = "hachure"))
grid::grid.newpage()
grid::grid.draw(exc_grob(s))
```

exc_group *Group elements together*

Description

Grouped elements select and move together in Excalidraw.

Usage

```
exc_group(..., id = NULL)
```

Arguments

...	exc_elements objects.
id	Group id. Generated if NULL.

Value

An exc_elements object.

Examples

```
exc_group(exc_rect(0, 0, 50, 50), exc_text("box", 0, 60))
```

exc_image *Images*

Description

Embeds an image file (PNG, JPEG, GIF, SVG or WebP) in the scene.

Usage

```
exc_image(
  file,
  x,
  y,
  width = NULL,
  height = NULL,
  ...,
  angle = 0,
  group = NULL,
  id = NULL,
  custom = NULL
)
```

Arguments

file	Path to the image file.
x, y	Top-left corner in px.
width, height	Size in px. If only one is given the other keeps the image's aspect ratio; if neither, the image's own size is used (PNG, GIF and SVG only).
...	Style properties such as stroke, fill, fill_style, stroke_width, stroke_style, roughness, opacity, font and font_size. See exc_style().
angle	Rotation in degrees, clockwise, around the shape's centre.
group	Group id(s) to put the element in. See exc_group().
id	Element id. Generated if NULL.
custom	A list stored in the element's customData.

Value

An exc_elements object carrying the image data.

---

exc_install_fonts *Install the handwriting fonts on this computer*

Description

Copies the bundled fonts (Excalifont, Virgil, Comic Shanns, Caveat and the others in exc_fonts()) into your user font folder, so other programs such as PowerPoint, Keynote or Word can show them. Nothing is installed unless you call this.

Usage

```
exc_install_fonts(dir = NULL, overwrite = FALSE)
```

Arguments

dir	Font folder. Defaults to ~/Library/Fonts on macOS, %LOCALAPPDATA%/Microsoft/Windows/Fonts on Windows and ~/.local/share/fonts elsewhere.
overwrite	Replace fonts that are already there.

Value

The installed file paths, invisibly.

exc_library *Load an Excalidraw library from a file or the web*

Description

Load an Excalidraw library from a file or the web

Usage

```
exc_library(path)
```

Arguments

path	A .excalidrawlib file, or its URL (for example from https://libraries.excalidraw.com). Downloads are cached.

Value

An exc_library; place items with exc_library_item().

exc_line *Lines, arrows and freehand strokes*

Description

Points are absolute positions in px. Give x and y vectors, or a two-column matrix or data frame as x.

Usage

```
exc_line(
  x,
  y = NULL,
  ...,
  style = NULL,
  closed = FALSE,
  curved = FALSE,
  angle = 0,
  group = NULL,
  id = NULL,
  custom = NULL,
  seed = NULL
)

exc_arrow(
  x,
```

```
  y = NULL,
  ...,
  start = NULL,
  end = "arrow",
  curved = FALSE,
  elbowed = FALSE,
  label = NULL,
  style = NULL,
  angle = 0,
  group = NULL,
  id = NULL,
  custom = NULL,
  seed = NULL
)

exc_freedraw(
  x,
  y = NULL,
  ...,
  pressures = NULL,
  style = NULL,
  angle = 0,
  group = NULL,
  id = NULL,
  custom = NULL,
  seed = NULL
)
```

Arguments

x, y	Point coordinates.
...	Style properties such as `stroke`, `fill`, `fill_style`, `stroke_width`, `stroke_style`, `roughness`, `opacity`, `font` and `font_size`. See `exc_style()`.
style	An `exc_style()` to start from.
closed	Join the last point back to the first. Closed lines can be filled with `fill`.
curved	Draw a smooth curve through the points instead of straight segments.
angle	Rotation in degrees, clockwise, around the shape's centre.
group	Group id(s) to put the element in. See `exc_group()`.
id	Element id. Generated if `NULL`.
custom	A list stored in the element's `customData`.
seed	The rough.js seed that decides the element's wobble. Random (from R's RNG) if `NULL`; change it to "re-roll" the hand-drawn look.
start, end	Arrowheads: `NULL` for none, or one of `"arrow"`, `"bar"`, `"circle"`, `"circle_outline"`, `"triangle"`, `"triangle_outline"`, `"diamond"`, `"diamond_outline"`.
elbowed	Draw the arrow with right-angle elbows.

label	Optional text placed at the middle of the arrow.
pressures	Optional pen pressures (0-1), one per point. If NULL, Excalidraw simulates pressure.

Value

An exc_elements object.

Examples

```
exc_line(c(0, 50, 100), c(0, 40, 0))
exc_line(c(0, 80, 40), c(0, 0, 60), closed = TRUE, fill = "green")
exc_arrow(c(0, 150), c(0, 0), label = "next")
exc_freedraw(cos(seq(0, 6, 0.2)) * 30, sin(seq(0, 6, 0.2)) * 30)
```

---

exc_mermaid *Mermaid diagrams as editable Excalidraw elements*

Description

Converts Mermaid text (flowcharts, sequence and class diagrams) to Excalidraw elements with Excalidraw's own converter, run in headless Chrome (needs the chromote package and Chrome or Chromium). Other diagram types come back as an embedded image.

Usage

```
exc_mermaid(code, font_size = 20)
```

Arguments

code	Mermaid diagram text.
font_size	Font size for the diagram text.

Value

An exc_scene.

Examples

```
## Not run:
exc_mermaid("flowchart LR\n  idea[Idea] --> sketch[Sketch] --> chart{{Chart!}}") |>
  exc_save("flow.png")

## End(Not run)
```

exc_modify *Change elements in a scene*

Description

Select elements with a condition on the columns of `as.data.frame(scene)` and set new property values.

Usage

```
exc_modify(scene, where = TRUE, ...)
```

Arguments

scene	An exc_scene.
where	A condition using columns such as type, text, strokeColor or role. TRUE selects everything.
...	Properties to set: style names as in exc_style() (stroke, fill, roughness, ...) or raw Excalidraw property names (fontSize, text, x, ...).

Value

The updated scene.

Examples

```
s <- exc_scene(exc_text("a", 0, 0), exc_rect(0, 30, 50, 50))
exc_modify(s, type == "rectangle", fill = "red", fill_style = "solid")
```

exc_papers *Paper backgrounds*

Description

Names of the paper backgrounds the native renderer can draw behind a scene. Pass one as `paper =` to exc_save(), exc_grob() or plot().

Usage

```
exc_papers()
```

Value

A character vector.

Examples

```
exc_papers()
```

exc_peep *Hand-drawn people from Open Peeps*

Description

Draws a person in Pablo Stanley's Open Peeps style (CC0), generated offline with the DiceBear Open Peeps generator (MIT). The same `seed` always gives the same person. Peeps are SVG images: crisp in Excalidraw and in SVG/PDF output (the native renderer needs the rsvg package).

Usage

```
exc_peep(
  scene = NULL,
  x = 0,
  y = 0,
  size = 160,
  seed = NULL,
  face = NULL,
  head = NULL,
  accessories = NULL,
  facial_hair = NULL,
  skin = NULL,
  clothing = NULL,
  ...,
  panel = 1
)

exc_peep_options()
```

Arguments

`scene`	An `exc_scene` to add the person to, or `NULL` to just return the element.
`x`, `y`	Top-left corner, in px (or data values for converted ggplots).
`size`	Width and height in px.
`seed`	Any string or number; each gives a different person.
`face`, `head`, `accessories`, `facial_hair`	Optional choices, such as `face = "smile"` or `head = "afro"`. See `exc_peep_options()`.
`skin`, `clothing`	Colours (hex, without or with #).
`...`	Other DiceBear Open Peeps options.
`panel`	Panel number for faceted plots.

Value

The scene with the person added, or an `exc_elements` object if `scene` is `NULL`.

`exc_peep_options()`: a named list of the allowed values for each option.

Examples

```
exc_scene() |>
  exc_peep(0, 0, seed = "Ada", face = "smile") |>
  exc_peep(170, 0, seed = "Grace", clothing = "#a5d8ff")
```

---

exc_pptx *Hand-drawn PowerPoint slides*

Description

Writes one slide per scene (or per chart, or per Excalidraw frame) to a PowerPoint file. By default each drawing is placed as **editable vector shapes** (with the rvg package), so strokes can be recoloured or moved in PowerPoint; use `vector = FALSE` for a picture instead.

Usage

```
exc_pptx(
  x,
  path,
  titles = NULL,
  vector = TRUE,
  text = c("outline", "font"),
  template = NULL,
  scale = 2
)
```

Arguments

`x`	An `exc_scene`, a ggplot, a `sketch()`, or a list of these. A scene with frames becomes one slide per frame.
`path`	Output `.pptx` file.
`titles`	Slide titles (recycled). `NULL` or `NA` keeps the frame name for frame slides and leaves other slides without a title.
`vector`	Place drawings as editable vector shapes (`TRUE`) or as PNG pictures (`FALSE`).
`text`	`"outline"` or `"font"`; see above.
`template`	Optional `.pptx`/`.potx` file to use as the template.
`scale`	Pixel density for pictures when `vector = FALSE`.

Details

Text is drawn as shapes by default so it keeps the handwriting font on any computer. With `text = "font"` it stays editable text, which shows in the handwriting font only where the fonts are installed (see `exc_install_fonts()`).

Value

path, invisibly.

Examples

```
## Not run:
library(ggplot2)
p <- ggplot(mpg, aes(class)) + geom_bar(fill = "#a5d8ff")
exc_pptx(list(p, exc_mermaid("flowchart LR\n A --> B")), "talk.pptx",
         titles = c("What people drive", "How it works"))

## End(Not run)
```

---

exc_preset *Style presets*

Description

A preset bundles sloppiness, font, ink colour, fill style and paper into a named look:

- "whiteboard": Excalidraw's own look.
- "xkcd": wobblier, bowed lines and a comic font.
- "blueprint": white ink on blueprint paper.
- "chalkboard": chalky lines on a dark board.
- "pencil": thin grey lines on lined notebook paper.
- "marker": thick marker strokes and a bold font.

Usage

```
exc_preset(
  name = c("whiteboard", "xkcd", "blueprint", "chalkboard", "pencil", "marker")
)
```

Arguments

name	Preset name.

Value

An exc_preset object (a list).

Examples

```
exc_preset("blueprint")
```

exc_rect *Excalidraw shapes: rectangles, ellipses and diamonds*

Description

Excalidraw shapes: rectangles, ellipses and diamonds

Usage

```
exc_rect(
  x,
  y,
  width,
  height,
  label = NULL,
  ...,
  style = NULL,
  rounded = FALSE,
  angle = 0,
  group = NULL,
  id = NULL,
  custom = NULL,
  seed = NULL
)

exc_ellipse(
  x,
  y,
  width,
  height,
  label = NULL,
  ...,
  style = NULL,
  angle = 0,
  group = NULL,
  id = NULL,
  custom = NULL,
  seed = NULL
)

exc_diamond(
  x,
  y,
  width,
  height,
  label = NULL,
  ...,
```

```
  style = NULL,
  rounded = FALSE,
  angle = 0,
  group = NULL,
  id = NULL,
  custom = NULL,
  seed = NULL
)
```

Arguments

x, y	Top-left corner of the shape's bounding box, in px.
width, height	Size in px.
label	Optional text drawn centred inside the shape. It stays attached to the shape when you edit it in Excalidraw.
...	Style properties such as stroke, fill, fill_style, stroke_width, stroke_style, roughness, opacity, font and font_size. See exc_style().
style	An exc_style() to start from.
rounded	Round the corners (rectangles and diamonds).
angle	Rotation in degrees, clockwise, around the shape's centre.
group	Group id(s) to put the element in. See exc_group().
id	Element id. Generated if NULL.
custom	A list stored in the element's customData.
seed	The rough.js seed that decides the element's wobble. Random (from R's RNG) if NULL; change it to "re-roll" the hand-drawn look.

Value

An exc_elements object.

Examples

```
exc_rect(0, 0, 160, 80, label = "Hello", fill = "yellow", fill_style = "solid")
exc_ellipse(200, 0, 120, 80, fill = "#a5d8ff", fill_style = "hachure")
exc_diamond(360, 0, 120, 100, rounded = TRUE)
```

---

exc_restyle *Restyle a whole scene*

Description

Applies a preset (or individual settings) to every element: sloppiness, stroke width, handwriting font, fill style and ink colour. Dark strokes and text become the preset's ink colour; coloured strokes are kept.

Usage

```
exc_restyle(scene, preset = "whiteboard", ...)
```

Arguments

scene	An exc_scene.
preset	A preset name or exc_preset().
...	Settings to override: roughness, font, ink, fill_style, stroke_width, paper, background, bowing.

Value

The restyled scene. The paper is remembered for exc_save() and plot().

Examples

```
s <- exc_scene(exc_rect(0, 0, 120, 60, label = "hi", fill = "blue"))
exc_restyle(s, "chalkboard")
```

---

exc_save *Save a scene as an image*

Description

Renders the scene offline with the native renderer (rough.js in V8 plus grid) and saves it. The file type follows the extension:

- .png, .jpg: bitmap via ragg, with the handwriting fonts.
- .svg: vector via svglite, with the fonts embedded so the file looks the same everywhere.
- .pdf: vector via cairo, with text drawn as outlines.

Usage

```
exc_save(
  scene,
  file,
  scale = 1,
  padding = 10,
  background = TRUE,
  paper = NULL,
  text = c("auto", "outline", "font"),
  engine = c("native", "excalidraw"),
  embed_scene = grepl("\\.excalidraw\\.(svg|png)$", tolower(file))
)
```

Arguments

scene	An exc_scene.
file	Output path.
scale	Pixel density: 1 = one image pixel per scene px, 2 for sharper "retina" images. For vector formats it scales the page size.
padding	Space around the drawing, in px.
background	Draw the background colour (FALSE = transparent).
paper	Optional paper background; see exc_papers().
text	Text drawing mode; see exc_grob().
engine	"native" draws in R (works offline, any format). "excalidraw" exports through the real Excalidraw editor in headless Chrome for a pixel-exact match (PNG and SVG only; needs chromote).
embed_scene	For SVG and PNG, also store the scene inside the image so it can be reopened as an editable drawing (like Excalidraw's "Embed scene" export). On by default for .excalidraw.svg / .excalidraw.png file names.

Value

file, invisibly.

Examples

```
s <- exc_scene(exc_ellipse(0, 0, 120, 80, label = "sketchy!", fill = "blue",
                           fill_style = "hachure"))
f <- tempfile(fileext = ".png")
exc_save(s, f, scale = 2)
```

---

exc_scene *Create an Excalidraw scene*

Description

A scene holds elements (shapes, text, arrows, images, frames), the canvas settings (appState) and any embedded image files. It is stored exactly as Excalidraw's JSON, so reading and writing files loses nothing.

Usage

```
exc_scene(..., background = "#ffffff", app_state = list())
```

Arguments

...	Elements: results of exc_rect(), exc_text() and friends, other scenes, or lists of these.
background	Canvas background colour.
app_state	Extra appState settings to store in the file.

Value

An exc_scene object.

Examples

```
scene <- exc_scene(
  exc_rect(0, 0, 160, 80, label = "Idea", fill = "yellow"),
  exc_arrow(c(160, 260), c(40, 40)),
  exc_ellipse(260, 0, 140, 80, label = "Chart")
)
scene
```

---

exc_scene_from_json *Turn Excalidraw JSON into a scene*

Description

Useful for the input$<id>_scene value sent by the editor in Shiny.

Usage

```
exc_scene_from_json(json)
```

Arguments

json	A JSON string in Excalidraw's file format.

Value

An exc_scene.

---

exc_set_style *Set the default style for new elements*

Description

Set the default style for new elements

Usage

```
exc_set_style(...)
```

Arguments

...	Style properties, as in exc_style(), or a single exc_style. Call with no arguments to reset to Excalidraw's defaults.

Value

Invisibly, the previous default style.

Examples

```
old <- exc_set_style(roughness = "cartoonist", stroke = "blue")
exc_set_style() # back to defaults
```

---

exc_slides_pdf *Print rendered slides to PDF*

Description

Prints a rendered reveal.js presentation (HTML) to PDF with headless Chrome, one slide per page. Needs the chromote package.

Usage

```
exc_slides_pdf(html, output = sub("\\.html?$", ".pdf", html), wait = 3)
```

Arguments

`html`	The rendered slides (`.html`).
`output`	PDF path.
`wait`	Seconds to let the slides (and their hand-drawn marks) finish drawing before printing.

Value

`output`, invisibly.

---

exc_stickers *Stickers: hand-drawn icons, people and doodles*

Description

scratchy bundles a dozen community libraries from https://libraries.excalidraw.com (all MIT licensed): R and RStudio icons, stick figures, storytelling faces, speech bubbles, sticky notes, stars, hearts, emojis, data-viz icons, hand-drawn signs, general icons and data-science logos. They arrive as native Excalidraw elements, so they are hand-drawn, recolourable and editable.

Usage

```
exc_stickers(library = NULL)

exc_sticker(scene, sticker, x = 0, y = 0, width = NULL, ..., panel = 1)

exc_sticker_sheet(library, columns = 6)
```

Arguments

library	Library name, such as "emojis" or "stick-figures".
scene	An exc_scene to add the sticker to.
sticker	"library:item", where item is a number or a name, such as "emojis:12" or "stick-figures:Girl".
x, y	Where to put the sticker's top-left corner. Data values work for scenes made from a ggplot; px otherwise (or in I()).
width	Width to scale the sticker to, in px.
...	Style properties to apply to every element of the sticker, such as stroke, fill or roughness.
panel	Panel number for faceted plots.
columns	Number of columns in the preview grid.

Details

- exc_stickers() lists them; exc_sticker_sheet() draws a numbered preview of one library.
- exc_sticker() adds one to a scene.
- exc_peep() adds a hand-drawn person from Open Peeps.
- read_excalidraw_lib() and exc_library_item() work with any other library file.

Value

exc_stickers(): a data frame with one row per sticker (library, item, name, sticker). Use the sticker column with exc_sticker().

exc_sticker_sheet(): a scene with every sticker of the library and its number; draw it with plot().

Examples

```
head(exc_stickers())
exc_stickers("stick-figures")
```

exc_style *Describe an Excalidraw style*

Description

A style is a set of element properties (stroke colour, fill, sloppiness, and so on). Pass one to any element constructor with `style =`, or set a session default with `exc_set_style()`. Arguments given directly to a constructor override the style.

Usage

```
exc_style(
  stroke = NULL,
  fill = NULL,
  fill_style = NULL,
  stroke_width = NULL,
  stroke_style = NULL,
  roughness = NULL,
  opacity = NULL,
  font = NULL,
  font_size = NULL
)
```

Arguments

`stroke`	Stroke (and text) colour: a hex colour, an R colour name, or an Excalidraw hue name such as `"blue"`. Hue names win over R colour names and give Excalidraw's strong shade (see `excalidraw_colors()`).
`fill`	Background fill colour, or `"transparent"`. Hue names give Excalidraw's light fill shade, so `fill = "blue"` is `"#a5d8ff"`.
`fill_style`	`"hachure"`, `"cross-hatch"`, `"solid"` or `"zigzag"`. rough.js also understands `"dots"`, `"dashed"` and `"zigzag-line"`.
`stroke_width`	A number, or `"thin"` (1), `"bold"` (2) or `"extra_bold"` (4).
`stroke_style`	`"solid"`, `"dashed"` or `"dotted"`.
`roughness`	Sloppiness: a number, or `"architect"` (0), `"artist"` (1) or `"cartoonist"` (2).
`opacity`	Opacity from 0 to 100.
`font`	Font name (see `exc_fonts()`) or Excalidraw font id.
`font_size`	Font size in px. Excalidraw's presets are 16, 20, 28 and 36.

Value

A list of class `exc_style`.

Examples

```
exc_style(stroke = "blue", fill = "blue", fill_style = "hachure")
```

exc_template_example *An example template*

Description

A KPI card with {{metric}}, {{value}} and {{change}} text and a {{chart:trend}} slot, for trying exc_template_fill(). Write it out with write_excalidraw() to open it in Excalidraw and make it your own.

Usage

```
exc_template_example()
```

Value

An exc_scene.

exc_template_fill *Data-driven templates*

Description

Design a card, poster or slide in Excalidraw, with placeholders in the text, and fill it from R:

- {{column}} in any text is replaced by that value.
- A shape whose label is {{chart:name}} gets a sketched chart drawn to fit inside it (a ggplot, or a function of the data row returning one).
- A shape whose label is {{image:name}} gets an image file fitted inside it.

Usage

```
exc_template_fill(template, data = list(), charts = list(), images = list())

exc_template_render(
  template,
  data,
  dir = ".",
  file = "card-{.row}.png",
  charts = list(),
  images = list(),
  ...
)
```

Arguments

`template`	An `exc_scene` or a path to a template file (any format `read_excalidraw()` reads).
`data`	A named list or a one-row data frame of values.
`charts`	A named list of charts: ggplots, scenes, or functions that take the data row and return one.
`images`	A named list of image file paths (or functions of the row). Values in `data` with the placeholder's name work too.
`dir`	Folder to write the files to.
`file`	File name pattern. `{.row}` is the row number and `{column}` any column's value. The extension picks the format (`.png`, `.svg`, `.pdf`, `.excalidraw`, ...).
`...`	Options passed to `exc_save()`, such as `scale`.

Details

`exc_template_fill()` fills one copy; `exc_template_render()` fills and saves one image (or `.excalidraw` file) per row of a data frame.

Value

`exc_template_fill()`: the filled `exc_scene`.

`exc_template_render()`: the file paths, invisibly.

Examples

```
card <- exc_template_fill(
  exc_template_example(),
  data = list(metric = "Sketches drawn", value = "1,204", change = "+18% this month"),
  charts = list(trend = ggplot2::ggplot(data.frame(x = 1:12, y = cumsum(rnorm(12))),
                                        ggplot2::aes(x, y)) + ggplot2::geom_line())
)
plot(card)
```

---

exc_text *Text*

Description

Text

Usage

```
exc_text(
  text,
  x,
  y,
  ...,
  hjust = 0,
  vjust = 0,
  text_align = NULL,
  style = NULL,
  angle = 0,
  group = NULL,
  id = NULL,
  custom = NULL
)
```

Arguments

`text`	The text. Use `"\n"` for line breaks.
`x, y`	Position of the anchor point in px.
`...`	Style properties: `stroke` (text colour), `font`, `font_size`, `opacity`. See `exc_style()`.
`hjust, vjust`	Where the anchor sits on the text box: 0 = left/top, 0.5 = centre, 1 = right/bottom.
`text_align`	`"left"`, `"center"` or `"right"`. Defaults from `hjust`.
`style`	An `exc_style()` to start from.
`angle`	Rotation in degrees, clockwise, around the shape's centre.
`group`	Group id(s) to put the element in. See `exc_group()`.
`id`	Element id. Generated if `NULL`.
`custom`	A list stored in the element's `customData`.

Value

An `exc_elements` object.

Examples

```
exc_text("Hand-drawn!", 100, 50, font_size = 28, hjust = 0.5)
```

exc_translate *Move or scale a whole scene*

Description

Move or scale a whole scene

Usage

```
exc_translate(scene, dx = 0, dy = 0, scale = 1)
```

Arguments

scene	An exc_scene or exc_elements.
dx, dy	Offset in px.
scale	Scale factor applied around the scene's top-left corner (after moving). Sizes, points and font sizes scale; stroke widths stay the same.

Value

The transformed object.

excalidraw *The Excalidraw editor as an htmlwidget*

Description

Shows a scene in the real Excalidraw editor, in the RStudio viewer, a browser, R Markdown/Quarto HTML or a Shiny app. In "edit" mode you can draw, annotate and restyle; in Shiny the edited scene comes back as input$<id>_scene (see excalidrawOutput()).

Usage

```
excalidraw(
  scene = exc_scene(),
  mode = c("edit", "view"),
  theme = c("light", "dark"),
  grid = FALSE,
  zen = FALSE,
  width = NULL,
  height = NULL,
  elementId = NULL,
  debounce = 400,
  extras = TRUE
)
```

Arguments

scene	An exc_scene, or anything as_excalidraw() accepts (such as a ggplot).
mode	"edit" or "view" (read-only, pan and zoom).
theme	"light" or "dark".
grid	Show the grid.
zen	Zen mode: hide most of the toolbar.
width, height	Widget size (CSS units or pixels).
elementId	Optional element id.
debounce	Milliseconds to wait after the last edit before sending the scene to Shiny.
extras	Add the scratchy sidebar (the button next to Library): the bundled sticker libraries, Open Peeps people, the style presets of exc_preset(), Mermaid diagrams, and exports to PNG, SVG, animated SVG, .excalidraw and Obsidian .excalidraw.md.

Details

The editor and its fonts are bundled with the package, so it works offline. Because the bundle is a JavaScript module made of several files, save it with htmlwidgets::saveWidget(w, file, selfcontained = FALSE).

Value

An htmlwidget.

Examples

```
if (interactive()) {
  excalidraw(exc_scene(exc_rect(0, 0, 160, 80, label = "Edit me!", fill = "yellow")))
}
```

---

excalidraw_colors *Excalidraw's colour palette*

Description

Excalidraw's palette is built on Open Color. Each hue has five shades, from very light (1) to strong (5). Excalidraw uses shade 5 for strokes and shade 2 for fills.

Usage

```
excalidraw_colors(type = c("stroke", "background", "all"), n = NULL)
```

Arguments

type	"stroke" (strong colours for lines and text), "background" (light fills), or "all" for the full table.
n	Number of colours to return. Colours repeat if n is larger than the number of hues.

Value

A named character vector of hex colours, or a data frame for type = "all".

Examples

```
excalidraw_colors("stroke", 4)
excalidraw_colors("background", 4)
head(excalidraw_colors("all"))
```

---

excalidraw_proxy *Control an Excalidraw editor from the Shiny server*

Description

Create a proxy for an excalidrawOutput() and send it commands without re-rendering the editor (so hand-drawn changes are kept).

Usage

```
excalidraw_proxy(outputId, session = shiny::getDefaultReactiveDomain())

exc_proxy_update(proxy, scene, keep_app_state = FALSE)

exc_proxy_add(proxy, scene, placement = c("as-is", "right"))

exc_proxy_replace_chart(proxy, scene, chart = "chart")

exc_proxy_scroll_to(proxy)

exc_proxy_set(
  proxy,
  theme = NULL,
  view_mode = NULL,
  zen = NULL,
  grid = NULL,
  background = NULL
)

exc_proxy_reset(proxy)
```

```
exc_proxy_export(proxy, type = c("png", "svg"), embed = FALSE, scale = 2)

exc_proxy_mermaid(proxy, code, font_size = 20)
```

Arguments

outputId	The editor's output id.
session	The Shiny session.
proxy	A proxy from excalidraw_proxy().
scene	A scene, elements, or anything as_excalidraw() accepts.
keep_app_state	Keep the editor's current canvas settings.
placement	"as-is" keeps the elements' coordinates; "right" places them to the right of what's already there.
chart	A name for the chart. Elements from an earlier call with the same name are replaced (in the same place); everything else stays.
theme, view_mode, zen, grid, background	Editor settings to change.
type	Export format, "png" or "svg". The result arrives as input$<outputId>_export: a list with type and data (base64 PNG or SVG text).
embed	Embed the scene in the image so it stays editable.
scale	PNG pixel density.
code	Mermaid diagram text, such as "flowchart LR\n A --> B".
font_size	Font size for the diagram text.

Value

excalidraw_proxy() returns a proxy object; the other functions return the proxy invisibly, so they can be piped.

Examples

```
## Not run:
server <- function(input, output, session) {
  output$canvas <- renderExcalidraw(excalidraw())
  observeEvent(input$add_note, {
    excalidraw_proxy("canvas") |>
      exc_proxy_add(exc_rect(0, 0, 160, 80, label = "New note", fill = "yellow"))
  })
}

## End(Not run)
```

excalidrawOutput *Excalidraw in Shiny*

Description

Output and render functions for the `excalidraw()` widget. The widget reports back to the server as:

- `input$<outputId>_scene`: the scene as Excalidraw JSON, shortly after each edit. Turn it into a scene with `exc_scene_from_json()`.
- `input$<outputId>_selection`: ids of the selected elements.
- `input$<outputId>_export`: the result of `exc_proxy_export()`.
- `input$<outputId>_mermaid`: whether a `exc_proxy_mermaid()` call worked.

Usage

```
excalidrawOutput(outputId, width = "100%", height = "600px")

renderExcalidraw(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

`outputId`	Output id.
`width, height`	Size of the editor.
`expr`	An expression that returns an `excalidraw()` widget.
`env, quoted`	Standard htmlwidgets arguments.

Value

A Shiny output or render function.

read_excalidraw *Read a scene from a file*

Description

Reads Excalidraw JSON files (`.excalidraw`, `.json`), Obsidian drawings (`.excalidraw.md`), and SVG or PNG images with an embedded scene (such as Excalidraw exports made with "Embed scene", or files from `write_excalidraw()`).

Usage

```
read_excalidraw(path)
```

Arguments

path	File path.

Value

An exc_scene.

---

read_excalidraw_lib *Excalidraw libraries*

Description

Libraries (.excalidrawlib files) are collections of reusable drawings, such as the ones at https://libraries.excalidraw.com. read_excalidraw_lib() returns the items; exc_library_item() places one in a scene; write_excalidraw_lib() saves elements as a library.

Usage

```
read_excalidraw_lib(path)

write_excalidraw_lib(items, path)

exc_library_item(library, item, x = 0, y = 0, width = NULL)
```

Arguments

path	File path.
items	A named list of exc_elements (one per library item), or an exc_library.
library	An exc_library from read_excalidraw_lib().
item	Item number or name.
x, y	Where to put the item's top-left corner, in px.
width	Optional width to scale the item to.

Value

read_excalidraw_lib(): a list of items, each with a name and elements.

register_fonts *Register the bundled handwriting fonts*

Description

Called automatically when the package loads. Registered fonts work on devices that use 'system-fonts', such as `ragg::agg_png()` and `svglite::svglite()`. Other devices (for example `grDevices::pdf()`) get text drawn as glyph outlines by the native renderer instead.

Usage

```
register_fonts()
```

Value

Invisibly, the names of the registered font families.

run_sketch_studio *Sketch Studio: a point-and-click app for hand-drawn charts*

Description

A Shiny app built on scratchy:

1. **Data**: pick a built-in dataset, upload a CSV or paste from a spreadsheet.
2. **Chart**: choose a chart type and what goes on each axis.
3. **Style**: preset, sloppiness, fill, font, grid, and a new wobble.
4. **Add**: sticky notes, stickers, hand-drawn people and Mermaid diagrams, straight onto the canvas.
5. **Templates**: fill an Excalidraw template from a CSV, one image per row.
6. **Export**: PNG, SVG, PDF, `.excalidraw`, Obsidian `.excalidraw.md`, an editable `.excalidraw.svg`, and the R code that recreates the chart.

Usage

```
run_sketch_studio(..., launch.browser = interactive())
```

Arguments

`...`	Options passed to `shiny::shinyApp()`, such as `port`.
`launch.browser`	Open the app in a browser.

Details

The canvas is the real Excalidraw editor: draw, move and annotate freely. Changing chart settings swaps only the chart; your drawing stays.

Value

A Shiny app object; printing it (or calling the function at the console) runs the app.

Examples

```
if (interactive()) run_sketch_studio()
```

---

scale_fill_excalidraw *Excalidraw colour scales for ggplot2*

Description

Discrete scales use Excalidraw's palette: light fills for `fill`, strong colours for `colour`. The continuous versions run through the five shades of one hue.

Usage

```
scale_fill_excalidraw(..., type = c("background", "stroke"))

scale_colour_excalidraw(..., type = c("stroke", "background"))

scale_color_excalidraw(..., type = c("stroke", "background"))

scale_fill_excalidraw_c(..., hue = "blue")

scale_colour_excalidraw_c(..., hue = "blue")
```

Arguments

`...`	Passed to the underlying ggplot2 scale.
`type`	For the discrete scales, `"background"` (light) or `"stroke"` (strong) colours.
`hue`	For the continuous scales, an Excalidraw hue such as `"blue"`.

Value

A ggplot2 scale.

Examples

```
library(ggplot2)
p <- ggplot(mpg, aes(class, fill = drv)) + geom_bar() + scale_fill_excalidraw()
sketch(p)
```

sketch *Draw any plot in Excalidraw style*

Description

`sketch()` wraps a plot so that printing it converts it with `as_excalidraw()` and draws the result with the native renderer. The plot is laid out at the size of the device it is drawn on, so it works in the RStudio plot pane, in knitr/Quarto chunks, and with `ggplot2::ggsave("plot.png", sketch(p), width = 7, height = 4)`.

Usage

```
sketch(
  plot,
  ...,
  preset = NULL,
  paper = NULL,
  background = TRUE,
  width = NULL,
  height = NULL
)
```

Arguments

`plot`	A ggplot, a formula such as `~ hist(x)`, a function that draws with base graphics, or a grob.
`...`	Options passed to `as_excalidraw()`, such as `fill_style`, `roughness`, `font`, `theme` or `seed`.
`preset`	Optional `exc_preset()` name.
`paper`	Optional paper background; see `exc_papers()`.
`background`	Draw the background colour; `FALSE` leaves it transparent (nice on slides with a paper background).
`width, height`	Fixed layout size in px. By default the device size is used.

Value

An `exc_sketch` object. Print it to draw it, or turn it into a scene with `as_excalidraw()`.

Examples

```
library(ggplot2)
p <- ggplot(mpg, aes(displ, hwy, colour = drv)) + geom_point(size = 3) +
  scale_colour_excalidraw()
sketch(p)
sketch(p, preset = "chalkboard")
sketch(~ barplot(VADeaths, beside = TRUE))
```

`theme_excalidraw` *An Excalidraw-style ggplot2 theme*

Description

A clean, whiteboard-like theme: handwriting font, ink-coloured axis lines and ticks, no grey panel, and text sizes that land on Excalidraw's own font sizes (16, 20 and 28 px at the default `base_size`). It works with plain ggplot2 output and is the default theme used by `as_excalidraw()` and `sketch()`. It also pairs well with sketchy geoms from other packages such as ggsketch.

Usage

```
theme_excalidraw(
  base_size = 15,
  base_family = "Excalifont",
  ink = "#1e1e1e",
  grid = c("none", "y", "x", "xy"),
  background = "white",
  preset = NULL
)
```

Arguments

`base_size`	Base font size in points. 15 pt = 20 px, Excalidraw's default.
`base_family`	Font family; see `exc_fonts()`.
`ink`	Colour for text, axes and ticks.
`grid`	Which grid lines to draw: `"none"`, `"x"`, `"y"` or `"xy"`. Grid lines are dashed and light.
`background`	Plot background colour, or `"transparent"`.
`preset`	Optional `exc_preset()` whose font and ink to use.

Details

The handwriting fonts are registered with 'systemfonts', so a plot printed directly with this theme needs a device that uses it: ragg (in RStudio, set Tools > Global Options > Graphics > Backend to "AGG") or svglite. `sketch()` works on every device because it draws text as outlines.

Value

A ggplot2 theme.

Examples

```
library(ggplot2)
p <- ggplot(mpg, aes(class)) + geom_bar(fill = "#a5d8ff") + theme_excalidraw(grid = "y")
sketch(p, theme = NULL)
```

`use_excalidraw_slides` *Hand-drawn Quarto slides*

Description

Sets up a Quarto presentation that uses the `excalidraw-revealjs` format: Excalifont headings, Excalidraw colours, sketchy code blocks and callouts, and figures sized for slides. In the slides you can use:

- `[text]{.rn-underline}`, `.rn-circle`, `.rn-box`, `.rn-highlight`, `.rn-strike`, `.rn-cross` and `.rn-bracket` for hand-drawn marks (add `.fragment` to reveal them on click);
- `[text]{.sticky-note}` (also `.blue`, `.pink`, `.green`);
- `::: {.sketch-box}` for a box with a rough border (`data-fill` adds a hachure fill);
- `excalidraw: paper:` in the YAML (`dots`, `graph`, `lines`, `kraft`, `blueprint`, `chalkboard`);
- `sketch()` and `plot()` of scenes in R chunks.

Usage

```
use_excalidraw_slides(path = "slides", file = "slides.qmd", overwrite = FALSE)
```

Arguments

`path`	Folder for the presentation. Created if needed.
`file`	Name of the slides file.
`overwrite`	Replace existing files.

Value

The path of the slides file, invisibly.

Examples

```
## Not run:
use_excalidraw_slides("talk")
# then: quarto render talk/slides.qmd

## End(Not run)
```

`write_excalidraw` *Write a scene to a file*

Description

The format follows the file extension:

- `.excalidraw` or `.json`: Excalidraw's JSON format, for excalidraw.com and the VS Code extension.
- `.excalidraw.md`: a drawing for the Obsidian Excalidraw plugin. Images are saved in a folder next to it.
- `.excalidraw.svg`, `.excalidraw.png` (or any `.svg` / `.png`): a rendered image with the scene embedded, so it can be opened again as an editable drawing in excalidraw.com, VS Code or with `read_excalidraw()`.

Usage

```
write_excalidraw(scene, path, pretty = TRUE, compress = TRUE, ...)
```

Arguments

`scene`	An `exc_scene`.
`path`	File path.
`pretty`	Indent the JSON (JSON and Obsidian files).
`compress`	For Obsidian files, store the drawing compressed, as the plugin does by default.
`...`	For images, options passed to `exc_save()` such as `scale` or `paper`.

Value

`path`, invisibly.

Examples

```
s <- exc_scene(exc_rect(0, 0, 100, 60, label = "hi"))
f <- tempfile(fileext = ".excalidraw")
write_excalidraw(s, f)
read_excalidraw(f)

# an image that stays editable
f2 <- tempfile(fileext = ".excalidraw.png")
write_excalidraw(s, f2, scale = 2)
read_excalidraw(f2)
```

Index

annotations, 3
as.data.frame.exc_scene, 5
as_excalidraw, 5
as_excalidraw(), *12, 13*, 38, *40*, *45, 46*

exc_add, 8
exc_animate, 8
exc_arrow *(*exc_line*)*, 19
exc_bbox, 10
exc_callout *(*annotations*)*, 3
exc_color, 10
exc_d2, 11
exc_data_to_px, 12
exc_data_to_px(), *3*
exc_diamond *(*exc_rect*)*, 26
exc_edit, 13
exc_ellipse *(*exc_rect*)*, 26
exc_font_fetch, 13
exc_fonts, 14
exc_fonts(), *7*, *11*, *18*, *33*, *46*
exc_frame, 15
exc_freedraw *(*exc_line*)*, 19
exc_grob, 16
exc_grob(), *22*, *29*
exc_group, 17
exc_group(), *18*, *20*, *27*, *36*
exc_image, 17
exc_install_fonts, 18
exc_install_fonts(), *12*, *24*
exc_library, 19
exc_library_item *(*read_excalidraw_lib*)*, 42
exc_library_item(), *19*, *32*, *42*
exc_line, 19
exc_mark *(*annotations*)*, 3
exc_mark(), *7*
exc_mermaid, 21
exc_mermaid(), *11*
exc_modify, 22
exc_modify(), *5*
exc_note *(*annotations*)*, 3
exc_note(), *7*
exc_papers, 22
exc_papers(), *9*, *29*, *45*
exc_peep, 23
exc_peep(), *32*
exc_peep_options *(*exc_peep*)*, 23
exc_pptx, 24
exc_preset, 25
exc_preset(), *7*, *28*, *38*, *45, 46*
exc_proxy_add *(*excalidraw_proxy*)*, 39
exc_proxy_export *(*excalidraw_proxy*)*, 39
exc_proxy_export(), *41*
exc_proxy_mermaid *(*excalidraw_proxy*)*, 39
exc_proxy_mermaid(), *41*
exc_proxy_replace_chart *(*excalidraw_proxy*)*, 39
exc_proxy_replace_chart(), *7*
exc_proxy_reset *(*excalidraw_proxy*)*, 39
exc_proxy_scroll_to *(*excalidraw_proxy*)*, 39
exc_proxy_set *(*excalidraw_proxy*)*, 39
exc_proxy_update *(*excalidraw_proxy*)*, 39
exc_rect, 26
exc_rect(), *29*
exc_restyle, 27
exc_restyle(), *9*, *16*
exc_save, 28
exc_save(), *5*, *8, 9*, *22*, *28*, *35*, *48*
exc_scene, 29
exc_scene_from_json, 30
exc_scene_from_json(), *41*
exc_set_style, 30
exc_set_style(), *33*
exc_slides_pdf, 31
exc_sticker *(*exc_stickers*)*, 31
exc_sticker_sheet *(*exc_stickers*)*, 31
exc_stickers, 31
exc_style, 33

exc_style(), *18*, *20*, *22*, *27*, *30*, *36*
exc_template_example, 34
exc_template_fill, 34
exc_template_fill(), *34*
exc_template_render
(exc_template_fill), 34
exc_text, 35
exc_text(), *29*
exc_title (annotations), 3
exc_translate, 37
excalidraw, 37
excalidraw(), *41*
excalidraw_colors, 38
excalidraw_colors(), *33*
excalidraw_proxy, 39
excalidrawOutput, 41
excalidrawOutput(), *37*, *39*

grDevices::pdf(), *43*

I(), *3*, *12*, *32*

ragg::agg_png(), *43*
read_excalidraw, 41
read_excalidraw(), *35*, *48*
read_excalidraw_lib, 42
read_excalidraw_lib(), *32*
register_fonts, 43
renderExcalidraw (excalidrawOutput), 41
run_sketch_studio, 43

scale_color_excalidraw
(scale_fill_excalidraw), 44
scale_colour_excalidraw
(scale_fill_excalidraw), 44
scale_colour_excalidraw_c
(scale_fill_excalidraw), 44
scale_fill_excalidraw, 44
scale_fill_excalidraw_c
(scale_fill_excalidraw), 44
shiny::runGadget(), *13*
shiny::shinyApp(), *43*
sketch, 45
sketch(), *24*, *46*, *47*
svglite::svglite(), *5*, *43*

theme_excalidraw, 46
theme_excalidraw(), *7*

use_excalidraw_slides, 47

write_excalidraw, 48
write_excalidraw(), *5*, *34*, *41*
write_excalidraw_lib
(read_excalidraw_lib), 42