Package 'gdtools'

November 12, 2015

Version 0.0.5

License GPL-3

Title Utilities for Graphical Rendering

Description Useful tools for writing vector graphics devices.

LazyData TRUE

Imports Rcpp (>= 0.12.0)

Suggests htmltools, testthat

LinkingTo Rcpp

SystemRequirements cairo

BugReports https://github.com/davidgohel/gdtools/issues

NeedsCompilation yes

Author David Gohel [aut, cre],
Hadley Wickham [aut],
Jeroen Ooms [ctb],
Yixuan Qiu [ctb],
RStudio [cph]

Maintainer David Gohel <david.gohel@ardata.fr>

Repository CRAN

Date/Publication 2015-11-12 08:44:26

R topics documented:

font_family_exists *Check if font family exists.*

Description

Check if font family exists.

Usage

```
font_family_exists(font_family = "sans")
```

Arguments

font_family	font family name (case sensitive)

Value

A logical value

Examples

```
font_family_exists("sans")
font_family_exists("Arial")
font_family_exists("Courier")
```

raster_str *Draw/preview a raster into a string*

Description

raster_view is a helper function for testing. It uses htmltools to render a png as an image with base64 encoded data image.

Usage

```
raster_str(x, width = 480, height = 480, interpolate = FALSE)

raster_view(code)
```

Arguments

x	A raster object
width,height	Width and height in pixels.
interpolate	A logical value indicating whether to linearly interpolate the image.
code	base64 code of a raster

Examples

```
r <- as.raster(matrix(hcl(0, 80, seq(50, 80, 10)),
 nrow = 4, ncol = 5))
code <- raster_str(r, width = 50, height = 50)
if (interactive() && require("htmltools")) {
  raster_view(code = code)
}
```

`raster_write` *Draw/preview a raster to a png file*

Description

Draw/preview a raster to a png file

Usage

```
raster_write(x, path, width = 480, height = 480, interpolate = FALSE)
```

Arguments

`x`	A raster object
`path`	name of the file to create
`width,height`	Width and height in pixels.
`interpolate`	A logical value indicating whether to linearly interpolate the image.

Examples

```
r <- as.raster(matrix(hcl(0, 80, seq(50, 80, 10)),
 nrow = 4, ncol = 5))
raster_write(x = r, path = "raster.png", width = 50, height = 50)
```

`str_extents` *Compute string extents.*

Description

Determines the width and height of a bounding box that's big enough to (just) enclose the provided text.

Usage

```
str_extents(x, fontname = "sans", fontsize = 12, bold = FALSE,
  italic = FALSE)
```

Arguments

x	Character vector of of strings to measure
fontname	Font name
fontsize	Font size
bold,italic	Is text bold/italic?

Examples

```
str_extents(letters)
str_extents("Hello World!", bold = TRUE, italic = FALSE,
  fontname = "sans", fontsize = 12)
```

---

str_metrics *Get font metrics for a string.*

Description

Get font metrics for a string.

Usage

```
str_metrics(x, fontname = "sans", fontsize = 12, bold = FALSE,
  italic = FALSE)
```

Arguments

x	Character vector of of strings to measure
fontname	Font name
fontsize	Font size
bold	Is text bold/italic?
italic	Is text bold/italic?

Value

A named numeric vector

Examples

```
str_metrics("Hello World!")
```

Index